

KING FAHD UNIVERSITY OF
PETROLEUM & MINERALS



Coded Orthogonal Frequency Division Multiplexing
(COFDM)

Term # 012

Course # EE 573

Term Project Report

Date Of Submission : 22-May-2002

Submitted To
DR. ASRAR UL HAQ SHEIKH

Submitted By
KAMRAN ARSHAD
STUDENT ID # 210261

Table Of Contents

TABLE OF CONTENTS	2
ABSTRACT	5
INTRODUCTION	6
1.1 MOTIVATION	7
1.2 OUTLINE OF THE REPORT	8
PRELIMINARIES	9
2.1 MOBILE RADIO CHANNELS	9
2.1.1 N-RAY MULTIPATH CHANNEL MODEL	10
2.1.2 GAUSSIAN MOBILE RADIO CHANNEL	11
2.2 INTERFERENCE	12
2.2.1 INTER-SYMBOL INTERFERENCE	12
2.2.2 CO-CHANNEL INTERFERENCE	15
2.3 DIVERSITY	15
2.3.1 TIME DIVERSITY	16
2.3.1 FREQUENCY DIVERSITY	16
2.3.2 SPACE DIVERSITY	17
2.4 MULTIPATH	17
ORTHOGONAL FREQUENCY DIVISION MULTIPLEXING	19
3.1 OFDM HISTORY	19
3.2 PRINCIPLES OF OFDM	20
3.2 SINGLE CARRIER VS. MULTI-CARRIER SYSTEMS	23
3.4 WHY USE MULTIPLE CARRIERS	24
3.5 ORTHOGONALITY AND THE USE OF FFT	25
3.5.1 PRESERVING THE ORTHOGONALITY	26
3.5.2 USE OF FFT	27
3.6 CYCLIC OFDM SYMBOL EXTENSION	27
3.7 INHERENT DIFFICULTIES REGARDING OFDM	28
3.7.1 PEAK TO MEAN POWER RATIO	29
3.7.2 SYNCHRONIZATION	30

CODING **31**

4.1 WAVEFORM CODING	32
4.2 STRUCTURED SEQUENCE CODING	33
4.3 CODE RATE AND REDUNDANCY	33
4.4 CODING GAIN	34
4.5 LINEAR BLOCK CODES	34
4.5.1 VECTOR SPACES	35
4.5.2 VECTOR SUBSPACES	35
4.5.3 A (6,3) LINEAR BLOCK CODE EXAMPLE	36
4.6 CONVOLUTIONAL CODES	37
4.6.1 ENCODER STRUCTURE	37
4.6.2 ENCODER REPRESENTATIONS	39
4.6.3 CATASTROPHIC CONVOLUTIONAL CODE	43
4.6.4 HARD-DECISION AND SOFT-DECISION DECODING	43
4.6.5 PERFORMANCE ANALYSIS OF CONVOLUTIONAL CODE	51
4.6.6 TRANSFER FUNCTION OF CONVOLUTIONAL CODE	51
4.6.7 DISTANCE PROPERTIES	52
4.6.8 ERROR PROBABILITIES	53
4.6.9 DECODING DEPTH	54
4.6.10 DEGREE OF QUANTIZATION	54
4.6.11 DECODING COMPLEXITY FOR CONVOLUTIONAL CODES	54
4.7 TURBO CODES	55
4.7.1 ENCODING	55
4.7.3 DECODING	59
4.7.3 PERFORMANCE ANALYSIS OF TURBO CODES	65

COFDM **68**

5.1 WHY WE NEED ERROR CODING	69
5.2 SOFT DECISION AND CHANNEL STATE INFORMATION	70
5.3 INTERLEAVING	73
5.4 MORE CODING	73
5.5 CONCLUSION	74

LITERATURE SURVEY **74**

6.1 SURVEY	74
-------------------	-----------

COMPARISON OF CONVOLUTIONAL AND TURBO CODES FOR OFDM IN WIRELESS APPLICATIONS **77**

7.1 INTRODUCTION	77
-------------------------	-----------

7.2 PERFORMANCE OF TURBO CODES AND SERIALY CONNECTED CONVOLUTIONAL CODES IN WIDEBAND CDMA	79
7.2.1 BRIEF OVERVIEW OF TURBO AND SERIALY CONNECTED CONVOLUTIONAL CODES	79
7.2.2 SYSTEM MODEL	80
7.2.3 SIMULATION RESULTS	82
7.2.4 CONCLUSION	87
7.3 COMPARISON BETWEEN CONVOLUTIONALLY AND TURBO CODED OFDM	88
REFERENCES	92

ABSTRACT

With the rapid growth of digital communication in recent years, the need for high speed data transmission is increased. Moreover, future wireless communication systems are expected to support a wide range of services which includes video, data and voice. To achieve the necessary capacity for these systems, several schemes have been proposed and demonstrated at the base station of wireless communication systems.

OFDM is a promising candidate for achieving high data rates in mobile environment, due to its resistance to inter symbol interference (ISI). A common problem found in high speed communication is ISI. ISI occurs when a transmission interferes with itself and the receiver cannot decode the transmission correctly. For example, in wireless communication, same transmission is sent in all directions. Because the signal reflects from large objects such as mountains or buildings, the receiver sees than one copy of the signal. In communication terminology, this is called MULTIPATH. Since the indirect paths take more time to travel to the receiver, the delayed copies of the signal interfere with the direct signal causing ISI. So as communication system increase their information transfer speed, the time for each transmission necessarily becomes shorter. Since the delay time caused by the multipath remain constant, ISI becomes a limitation in high data rate communication. OFDM avoids this problem by sending many low speed transmissions simultaneously.

In order to minimize probability of error in detection at the receiver, COFDM is the modulation scheme of choice for all forms of digital broadcasting both audio, digital including satellite, terrestrial and cable. In all existing standards, coding refers to consist of inner convolutional codes with an outer Reed-Solomon code, but in case of OFDM, we can increase capacity as well as performance by replacing at least the inner code with a Turbo Code. So main objective of my term paper is the thorough literature survey of COFDM, which coding scheme is more feasible. So I will try to survey the literature regarding coding with OFDM or COFDM.

Chapter 1

Introduction

Radio transmission has allowed people to communicate without any physical connection for more than hundred years. When Marconi managed to demonstrate a technique for wireless telegraphy more than a century ago it was a major breakthrough and the start of a completely new industry. May be one could not call it a mobile wireless system, but there was no wire! Today, the progress in the semiconductor technology has made it possible, not to forgot affordable, for millions of people to communicate on the move all around the world.

The Mobile Communication Systems are often categorized as different generations depending on the services offered. The *first generation* comprises the analog frequency division multiple access (FDMA) systems such as the NMT and AMPS (Advanced Mobile Phone Services). The *second generation* consists of the first digital mobile communication systems such as the time division multiple access (TDMA) based GSM (Global System for Mobile Communication), D-AMPS (Digital AMPS), PDC and code division multiple access (CDMA) based systems such as IS-95. These systems mainly offer speech communication, but also data communication limited to rather low transmission rates. The *third generation* is currently under development and the first prototype systems have been just delivered.

During the past few years, there has been an explosion in wireless technology. This growth has opened a new dimension to future wireless communications whose ultimate goal is to provide universal personal and multimedia communication without regard to mobility or location [1,2,3] with high data rates. To achieve such an objective, the next generation personal communication networks will need to be support a wide range of services which will include high quality voice, data, facsimile, still pictures and streaming video. These future services are likely to include applications which require high transmission rates of several Mbps.

In the current and future mobile communications systems, data transmission at high bit rates is essential for many services such as video, high quality audio and mobile

integrated service digital network. When the data is transmitted at high bit rates, over mobile radio channels, the channel impulse response can extend over many symbol periods, which leads to inter symbol interference (ISI). Orthogonal Frequency Division Multiplexing (OFDM) is one of the promising candidate to mitigate the ISI. In OFDM signal bandwidth is divided into many narrow subchannels which are transmitted in parallel. Each subchannel is typically chosen narrow enough to eliminate the effect of delay spread. By combining OFDM with Turbo Coding and antenna diversity, the link budget and dispersive-fading limitations of the cellular mobile radio environment can be overcome and the effects of co-channel interference can be reduced.

1.1 Motivation

The public's desire for mobile communications and computing, as evidenced by the popularity of cellular phones, pagers, and laptop computers, combined with the rapid growth in demand for internet access, suggest a very promising future for wireless data services. The key to realizing this potential is the development and deployment of high performance radio systems, supporting high data rates with wide area coverage. Communicating at these high transmission rates over the harsh and hostile wireless channels with a limited spectrum, however, creates many difficulties and challenging problems. This implies that counter measures should be employed to mitigate the delay spread impairments and increase in the capacity of wireless systems will need to be achieved. One possible approach to communicate at high data rates, and increasing the system capacity is to use a very effective technique so called Orthogonal Frequency Division Multiplexing (OFDM). OFDM with diversity and coding has been proposed as an effective means for achieving high rates in wireless environment. Turbo codes have been shown to give near-capacity performance in additive white Gaussian noise channels and is being considered to enhance mobile wireless channel performance.

1.2 Outline of the Report

The overall plan of this report is as follows. After some introductory remarks in this chapter, we begin in earnest in Chapter 2. Chapter 2 gives fundamentals of mobile communications. Our major emphasis in the chapter is on the concepts of small scale propagation such as fading, time delay spread and Doppler spread. We also cover some fundamentals of wireless communication such as multipath and different models of mobile communication channels. In addition, some diversity concepts are also presented. Chapter 3 introduces Orthogonal Frequency Division Multiplexing (OFDM) systems and some associated difficulties and requirements are also discussed. Here we demonstrate the basic principles of OFDM technique, its advantages, and a comparison of OFDM systems with the existing systems. The use of FFT reduces the complexity at the receiver, so here I also discuss the use of FFT with OFDM, which replaces the bank of oscillators.

Chapter 4 is related with coding, here I will try to provide the basic material, regarding coding theory and some fundamental coding schemes. I also provide a brief overview of coding classification, and also discuss some basic terms like code rate, redundancy, coding gain etc. I also presents a complete description of block codes, some fundamental issues of convolutional codes, and than finally I conclude the discussion on turbo codes. As I stated earlier, turbo codes provides performance near Shannon limit in AWGN channels and is now being considered to enhance mobile wireless channel performance. Extensive research has been done on the use of turbo codes with OFDM.

Chapter 5 provides an overview on COFDM, which is a form of modulation, most suitable for the needs of terrestrial broadcasting. So in this chapter I try to cover some fundamental aspects of COFDM, put an emphasis on why we need error coding, what is soft decision and hard decision, what is the difference in performance of an uncoded and coded OFDM system and finally concluded why COFDM is an ideal choice for DAB and DVB system, and what are the aspects of using COFDM in radio mobile channels.

Chapter 6 provides a brief literature survey regarding OFDM in general, and COFDM in particular. Chapter 7 provides a nice comparison between different coding schemes with OFDM.

Chapter 2

Preliminaries

In contrast to wireline communications, mobile communications offers users mobility and allows users to communicate conveniently worldwide. However, transmission over radio media is extremely random and unpredictable, and does not offer easy analysis. The transmission may be reflected, diffracted and blocked by some obstacles inside the radio channel. In addition, the signal quality at the receiver may even be affected by the movement of the mobile subscriber. In terms of the stationary and stability of the media, mobile communications definitely is not as good as wireline communications where the media is highly predictable. As a consequence, in mobile radio system design, it is critical to know the mechanism behind electro mechanic wave propagation, so modeling of the radio channel is required. On the other hand, due to the limited spectrum and channel characteristics, there are various interferences which greatly affects the signal quality. These impairments have a strong negative impact on mobile communications, and countermeasures like diversity must be used to improve the link performance in hostile mobile environments. These characteristics are discussed in this chapter.

2.1 Mobile Radio Channels

In this article we consider the communication channel which exists between the transmitter and receiver. Accurate characterization of this channel is essential if we are remove the impairments imposed by the channel using signal processing at the receiver. The mobile radio channel has been characterized in a number of excellent treatises by Jakes, Lee and Steele. Mobile radio links are established between a fixed base station (BS) and a number of roaming mobile stations (MS). In order to cover a large area, it is necessary to have a number of transmitters. The coverage area of each transmitter is defined as area in which satisfactory communications between the mobile and the

transmitter can be achieved, and this is known as a cell. Nearby cells are assigned different frequencies, but as the distance between cells increases, the interference between them reduces, and it becomes possible to reuse frequencies, thus increasing system capacity.

As radio wave propagates, multiple reflections, diffractions and scattering occur from various objects. The interaction between these waves causes multipath fading in which two or more versions of the transmitted signal arrive at different times and with different attenuation. These echoes cause various kinds of interferences, to the received signal depending on the speed of transmission and the channel bandwidth. With multiple fading, there will be rapid changes in signal strength over a small travel distance or time interval. Moreover, due to varying Doppler shifts on different multipath signals, this will lead to spreading of the frequency spectrum or random frequency modulation. But our major emphasis is on the modeling of multipath fading channel.

2.1.1 N-Ray Multipath Channel Model

The small-scale variations due to the multipath fading of a mobile radio signal can be modeled as the impulse response of a mobile radio channel. This impulse response is in general a time-varying continuous time function and can reflect all the small scale properties of the channel and simulate or analyze any type of radio transmission through the channel. The time varying nature of impulse response is due to the motion of the objects in the radio environment and the time variability mainly depends on the transmission rate the speed of the receiver motion in space provided that the speeds of motion of surrounding objects at the transmitter and receiver are negligible. But when we consider high speed transmission than the channel can be thought as stationary for several packets, and usually a N-ray model is used to express the channel impulse response as [15,16]

$$c(t) = \sum_{n=0}^{N-1} \beta_n \delta(t - \tau_n)$$

where β_n and τ_n are the complex path gain, and time delay for the nth path of the mobile radio channel, respectively.

A common accepted model suggested that $|\beta_n|$ be a random variable with Rayleigh Distribution and $\angle\beta_n$ be a random variable uniformly distributed from 0 to 2π . Therefore, $c(t)$ is a zero mean complex Gaussian random variable. In addition, it is assumed that paths with different delays within the excess delay are equally probable. Consequently, the path delay is assumed as constant increment delay, so

$$\tau_n = \frac{\tau_{exc}}{N-1}$$

where τ_{exc} is excess delay of the multipath channel.

Besides most of the time, we will consider the composite channel impulse response, $h(t)$, rather than the pure channel impulse response, $c(t)$, as defined . The composite channel response, is equal to

$$h(t) = g(t) \otimes c(t)$$

in which $g(t)$ is the resultant pulse shaping filter at the transmitter and receiver, and \otimes denotes the convolution operator between two continuous time functions.

2.1.2 Gaussian Mobile Radio Channel

The simplest type of channel is the Gaussian Channel, which is basically a linear time invariant transmission system impaired by the thermal noise generated in the receiver. The noise is assumed to have a constant power spectral density (PSD) over the channel bandwidth, and the Gaussian amplitude probability density function (PDF). This type of channel is occasionally realized in digital mobile radio communications mainly in microcells where it is possible to have a line of sight with essentially no multipath propagation. Even when there is a multipath , but the mobile is stationary and there are no other moving objects such in its vicinity, as vehicles, the mobile channel may be thought of as Gaussian. This is because, due to the zero mobile speed, the fading is represented by a local path loss. The Gaussian channel is also important for providing an upper bound on system performance (for example BER).

2.2 Interference

In mobile radio transmission, there are many types of interference, jamming the desired radio signals, and all or at least most of these impairments should be investigated and analyzed in the design of mobile transmission. Then, countermeasures should be designed aiming to mitigate the effects. In this section, we particularly concerned with the inter symbol interference (ISI) and co-channel interference (CCI) which are usually the most dominant interferences in high speed transmissions.

2.2.1 Inter-Symbol Interference

The bandwidth of the channel is generally bandlimited and when a pulse with unlimited bandwidth (e.g. rectangular pulse) is transmitted through this channel, it will cause a truncation or distortion of that signal in the frequency domain. Equivalently, there is time dispersion (because when we truncate the pulse in frequency domain, in time domain it tends to increase) of the pulse and the pulse of each symbol will smear into the time intervals of succeeding symbols. This type of interference is regarded as inter-symbol interference and this leads to an increased probability of error at the receiver in detecting a symbol.

Obviously, a bandlimited pulse can be chosen for transmission in order to avoid the distortion in frequency domain due to the bandlimited channel. However, cutting the bandwidth of the transmitting pulse will in turns stretch or widen the width of the pulse in time. This causes overlapping of the symbols directly. The relationships among bandwidth of the channel, bandwidth of the transmitting pulse, and ISI are illustrated in figure shown:

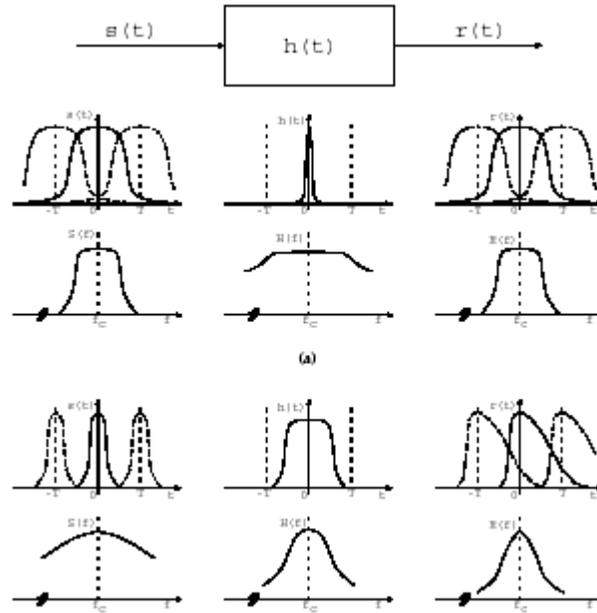


Figure 2.1: relationships among bandwidth of the channel, bandwidth of the transmitting pulse, and ISI

The transfer function of the filter is:

$$G(f) = \begin{cases} T & 0 \leq |f| \leq \frac{1-\alpha}{2T} \\ \frac{T}{2} \left[1 - \sin \left(\frac{\pi T}{\alpha} \left(|f| - \frac{1}{2T} \right) \right) \right] & \frac{1-\alpha}{2T} \leq |f| \leq \frac{1+\alpha}{2T} \\ 0 & |f| \geq \frac{1+\alpha}{2T} \end{cases}$$

and the corresponding impulse response is

$$g(t) = \frac{\cos \left(\frac{\alpha \pi t}{T} \right) \operatorname{sinc} \left(\frac{\pi t}{T} \right)}{1 - \left(\frac{2\alpha t}{T} \right)^2}$$

where α is the raised cosine rolloff factor which ranges from 0 and 1. The impulse response and transfer function of a raised cosine filter are plotted in figure 2.2 and 2.3 for various values of α . It can be seen for larger values of α , the impulse response of the filter decays faster, but with narrower signal bandwidth and vice versa. As a result, the rolloff factor, α , is in fact a design parameter for engineers and a spectral efficient transmission without ISI can be possibly achieved if α is properly selected.

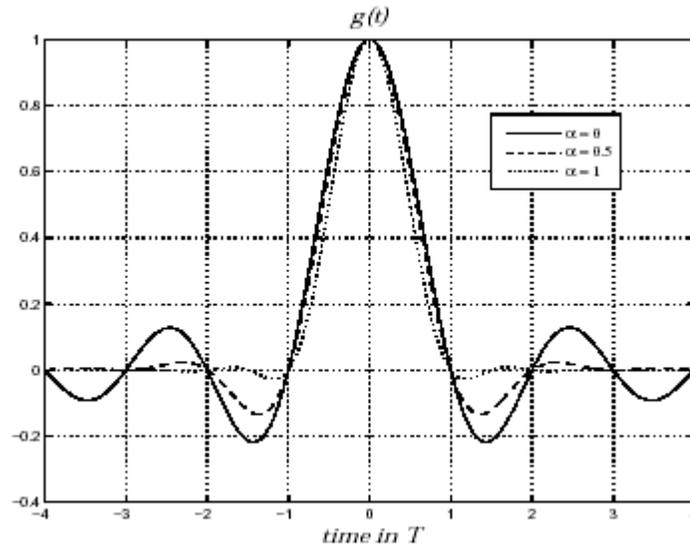


Figure 2.2: Pulse shape having a raised cosine filter

Note from figure 2.2, that the raised cosine filter is theoretically time unlimited. In practice, a truncated version of the filter is used since it is impossible to transmit a signal forever. Normally, the filter is only generated for time period between $-6T$ and $6T$. However, this will introduce very few performance degradation because of the fast impulse response decay at the zero-crossings (approximately as $\frac{1}{t^3}$ for $t \gg T$).

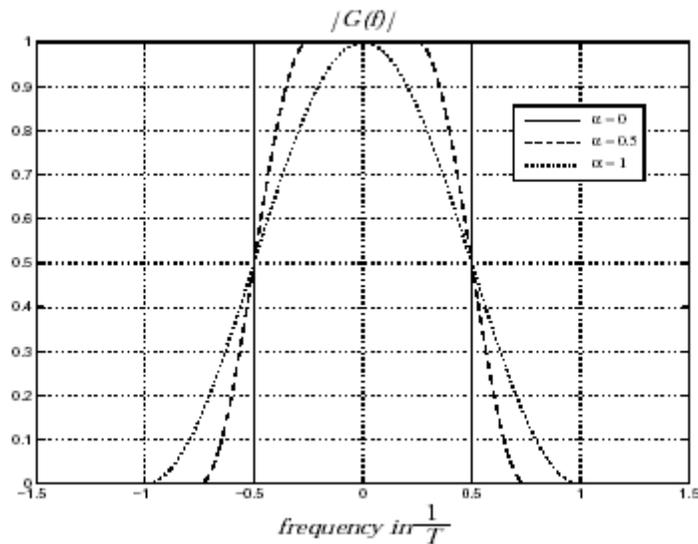


Figure 2.3: Magnitude transfer function of a raised cosine filter

In fact, pulse shaping techniques can only avoid creation of ISI when a bandlimited transmission is considered in AWGN channel. As discussed in section 2.1.1 and 2.1.2, radio channel induces multipath fading in mobile transmissions. As a consequence, even pulse shaping technique is employed, paths with delays greater the symbol period cause ISI by a multipath channel.

2.2.2 Co-Channel Interference

Apart from the interference caused by the channel, another interference which is also one of the most critical factor that limits the system performance and the capacity of the system is the co-channel interference (CCI). CCI is existed in any wireless multiple access system. In frequency division multiple access (FDMA) or time division multiple access (TDMA) or space division multiple access (SDMA) system,, frequency reuse is utilized to provide unlimited coverage of wireless system, so that there are users who share the sane frequency band at the same time and hence, co-channel users surely create CCI one another. So there is always a trade off between spectral efficiency and system performance.

Compared with TDMA and FDMA, SDMA reuses every frequency band more frequently to enhance the spectral efficiency and/or support more users by using spatial information of the co-channel users. Furthermore, another famous and highly potential wireless multiple access system, spread spectrum multiple access (SSMA) system also suffers the problem of CCI. Although, the users, in theory, are using the same frequency band and can be separated by some orthogonal strategies such as direct sequence (DS), the near far problem caused by imperfect correlation of the sequences and the impact of relative distances of the co-channel users from the base is a major limitation of the performance and capacity of SSMA system.

2.3 Diversity

So as already discussed, there are many impairments imposed by the channel. Some countermeasures are available to overcome or reduce the effects of these problems and one effective way which is often used is diversity. Diversity makes use of the signal from

multiple independent channels to detect the received signal. The independent channels are sometimes referred as diversity channels. Diversity is normally grouped into three main categories:

- Time Diversity
- Frequency Diversity
- Space Diversity

2.3.1 Time Diversity

Time diversity is usually carried out by channel coding in combination with limited interleaving. The mechanism of channel coding is that after some special structuring of the transmitted digital symbols, the receiver is able to detect and/or correct some of the errors caused by the channel. To this end, the underlying assumption is that the digital symbols transmitted at different time fade differently and it is unusual to have many consecutive symbols in a deep fade at a time. However, in practice, interleaving prior to transmission should be done to randomize the errors and simulate a fast fading channel. In addition, to be effective, the interleaving delay should be greater than the coherent time of the channel, T_c .

In fact, time diversity is extremely useful in fast fading channels or high mobility environment, but it offers very little protection under slow fading environment unless significant interleaving delays can be tolerated.

2.3.1 Frequency Diversity

Frequency diversity can be exploited simply by transmitting the same information through different frequency channels with frequency separation greater than the coherence bandwidth, B_c of the channel. As a matter of fact, the signals at different frequencies, see the channel differently and fade independently to provide diversity. In addition, frequency diversity is usually referred as path diversity. When the energy of the transmitting signal spread over a spectrum with bandwidth larger than that of the channel, multipath fading is induced and path diversity exists.

In TDMA, path diversity is obtained through the use of equalizers. Equalization is usually implemented by an array of delay elements or a tapped delay line which weights the delayed paths according to their SNRs and sums them up after co-phasing to form an array output. The array output signal has taken into account the paths with different delays and can have any degree of diversity gain depending on how much complexity can be tolerated at the receiver. As a result, equalization is used as a countermeasure to equalize the radio channel which is supposed not to be frequency selective instead of providing diversity gain.

2.3.2 Space Diversity

Diversity can also be obtained by making use of spatial information of received signals and this is known as space diversity. Space diversity is obtained by an array of antennas which can either form directional pattern to resolve signals from different direction of arrivals. To receive independent array signals to provide diversity (i.e. antenna diversity). In contrast to time and frequency diversity, space diversity is effective in any fading environment, and the performance is not restricted by the amount of frequency spreading of the transmitted signal. However, the only thing sacrificed is the space and/or complexities of the design together with the signal processing at the transmitter and/or the receiver.

2.4 Multipath

Multipath is the composition of a primary signal plus duplicate or echoed images caused by reflections of signals off objects between the transmitter and receiver. In figure 2.4, the receiver hears the primary signal sent directly from the transmission facility, but it also sees secondary signals that are bounced off nearby objects. These bounced signals will arrive at the receiver later than the incident signal. Because of this misalignment, the out of phase signals will cause intersymbol interference or distortion of the received signal. Although most of the multipath caused by bounces of tall objects, multipath can also occur from bounces on low objects such as lakes and pavements.

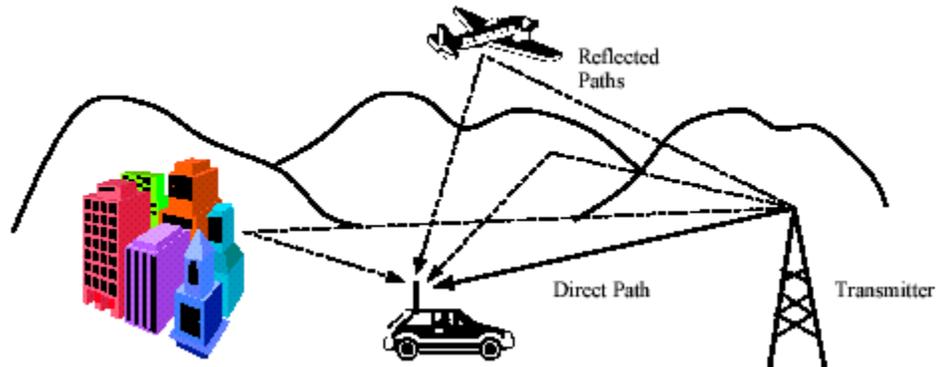


Figure 2.4: Multipath Reception

The actual received signal is a combination of a primary and several echoed versions. Because the distance traveled by the original signal is shorter than the bounced signal, the time differential causes two signals to be received. These signals are overlapped and combined into a single one. In real time, the time between the first received signal and the last echoed signal is called the *Delay Spread* which can be as high as 4 seconds.

In analog systems, such as television, this multipath situation can actually be seen by the human eye. Sometimes there is a “ghost” image on your television and no matter how much you adjust the set, the image does not go away. In these analog systems, this is an annoyance. In digital systems, it usually corrupts the data stream and causes loss of data and lower performance.

Chapter 3

Orthogonal Frequency Division Multiplexing

OFDM is an efficient data transmission system, which is particularly suited for transmission over frequency selective channels. On such channels, the insertion of a guard interval before each transmitted block, longer than the largest delay spread, removes the necessity of further inter channel interference equalization. A loss of 10-20% in spectral efficiency due to the insertion of guard interval is normally accepted. For some multipath channels, having some large delay spread, this require longer OFDM symbol duration with a corresponding increase in number of sub-carriers. This also entails higher FFT complexity and greater phase noise sensitivity. The use of fixed guard interval does not protect the system if the channel has longer echoes.

3.1 OFDM History

The concept of using parallel data transmission by means of frequency division multiplexing (FDM) was published in mid 60s by Chang [4,17]. Some early developers can be traced back in the 50s. A U.S. patent was filled and issued in January, 1970. The idea was to use parallel data streams and FDM with overlapping subchannels to avoid the use of high speed equalization, and to combat impulsive noise, and multipath distortion as well as to fully use the available bandwidth. The initial applications were in the military communications. In the telecommunications field, the term of discrete Multitone, multi-channel modulation and multi-carrier modulation (MCM) are widely used and sometimes they are interchangeable with OFDM. In OFDM, each carrier is orthogonal to all other carriers. However, this condition is not always maintained in MCM. OFDM is an optimal version of multi carrier transmission schemes.

For a large number of subchannels, the arrays of sinusoidal generators and coherent demodulator require in parallel system become unreasonably expensive and complex. The receiver needs precise phasing of the demodulator carriers and sampling times in order to keep crosstalk between subchannels acceptable. Weinstein and Ebert [9] applied the discrete Fourier transform to parallel data as part of modulation and demodulation process. But the problem with FFT is that here we can use only limited frequencies, which are the integral multiples of $1/T$, where T is the symbol time period.

In the 1980s, OFDM has been studied for high-speed modems, digital mobile communications [11] and high density recording. One of the systems used a pilot tone [18] for stabilizing carrier and clock frequency control and trellis coding was implemented.

In 1990s, OFDM has been exploited for wideband data communications over mobile radio FM channels, high bit-rate digital subscriber line, asymmetric digital subscriber line, very high speed digital subscriber lines, digital audio broadcasting (DAB) and HDTV terrestrial broadcasting.

3.2 Principles of OFDM

The first OFDM scheme with overlapping spectra was introduced by Chang in 1966 [4]. As opposed to today's systems, a bank of sinusoidal generators was used to produce the orthogonal channels. In 1971, Weinstein and Ebert [9] proposed to use the Discrete Fourier Transform (DFT) for this purpose and the first modern OFDM system was born. There exist many overviews and detailed presentation of OFDM in the literature. But here I will present most fundamental properties and concepts needed for the rest of the report.

In OFDM system many sub-channels are used in parallel. The channel are overlapping in

$$x_k = \sqrt{\frac{E_s}{N}} \sum_{n=0}^{N-1} c_n e^{j2\pi \frac{nk}{N}}$$

frequency, but the distance between them is chosen so that the different channels are somehow orthogonal. In the most simple form the sample at time k of the complex envelope is given by [19] The modern configuration of an OFDM system is shown in figure:

Where N is the size of DFT and E_s is the symbol energy. The size of the DFT is equal to the number of sub-channels available for transmission, but all of the channels need not to be active. The sub-channel bandwidth is given by:

$$f_{sc} = \frac{1}{T_s} = \frac{f_{samp}}{N}$$

where f_{samp} is the sampling rate and T_s is the symbol time. Normally, a cyclic extension [11] of the symbol is used to mitigate the effects of ISI. This, so called, cyclic prefix is a copy of the last G samples and these are transmitted prior to the actual symbol. Alternatively, a prefix of length G_1 and a postfix of length G_2 can be used. The main idea behind OFDM is to use long symbols such that the part affected by ISI can be disregarded with only a minor energy loss. The signal is often also windowed in order to improve the performance. This is often important for reducing out of band emissions and to decrease the inter-channel interference, ICI due to frequency offsets. Different approaches for the windowing functions are proposed in the literature, normally multiplication with a raised cosine [19] is proposed. The complex envelope including both pulse shaping/windowing and the pre/postfix can be written as:

$$x(t) = \sqrt{\frac{E_s}{N+G}} \sum_{l=-\infty}^{\infty} \sum_{k=-G_1}^{N+G_2} \sum_{n=n}^{N-1} c_{k,l} e^{j2\pi n \frac{k}{N}} w(t - \frac{k}{f_s} - lT_{tot})$$

where $c_{k,l}$ denotes the data symbol on the k -th sub-channel during the l -th OFDM symbol, $w(t)$ denotes the windowing function, and $T_{tot} = (N+G)/f_s$ denotes the total OFDM symbol time including the cyclic pre/postfix.

An often used model of wireless system is that the transmitted signal is convolved with the channel impulse response. In OFDM receivers, a DFT is performed in order to detect the data symbols on each of the sub-channels. If we assume ideal synchronization, a cyclic extension exceeding the maximum excess delay (delay between the first and last tap of the channel) and no frequency offset then the received symbol after the DFT is given by:

$$r_k = H_k c_k + n_k$$

for a slowly varying time dispersive channel with additive Gaussian noise. Here n_k denotes the white Gaussian noise and H_k denotes the complex value of the transfer function at sub-channel k . In the ideal case, the symbol is only disturbed by noise, phase shift and attenuated by the frequency response, after demodulation and transmission over a multipath channel. This is in contrast with single carrier system where the received signal is generally corrupted by ISI.

A block description of a conventional OFDM modulator and demodulator is presented in figure 3.2:

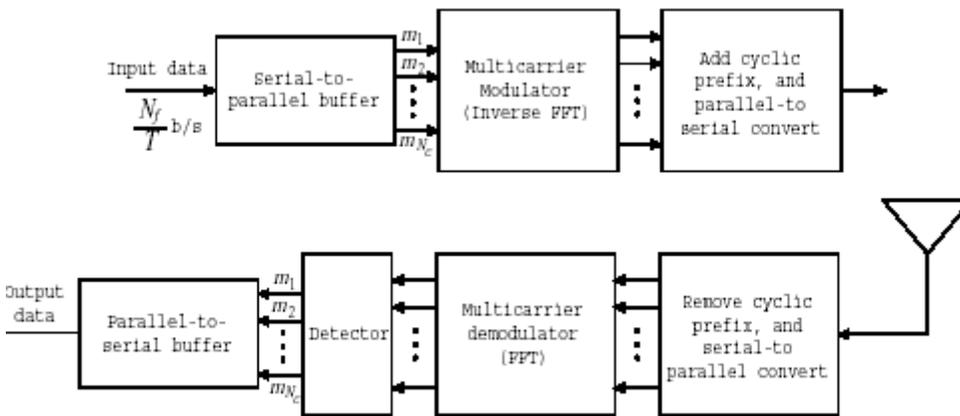


Figure 3.1: Block description of a conventional OFDM modulator and demodulator chain

First the code bits are fed into modulator where groups of $\log_2 M$ bits are mapped to M -ary symbols, where M is the size of symbol alphabet on each carrier. Then N of these symbols are grouped into one OFDM symbol (note, however, that some of the outer channels may be zero-valued). Each data symbol is assigned to a specific sub-channel by the IDFT, the inverse discrete Fourier transform. Normally this latter operation is performed as an inverse fast Fourier transforms due to its lower complexity. Then the cyclic extension is added and finally pulse shaping is performed before the signal is up-converted and transmitted over the channel.

At the receiver, the received signal is down converted and sampled. Then the samples are fed in parallel to the channel estimator, the FFT and the automatic control gain, respectively. The automatic control gain (AGC) tracks the incoming power and adjust the

level so that the fully dynamic range of the analog-to-digital converter is used. The task of the synchronizer, or more precise the coarse synchronizer, is to find the start of each data packet and each OFDM symbol, and to find an estimate of the frequency offset so that compensation can be made before the samples are fed to the FFT block. The OFDM is very sensitive to frequency offsets, and in order to avoid offset compensation has to be made before the DFT.

The channel estimator is used to enable coherent modulation schemes. The time dispersive channel results in a phase shift and amplitude attenuation according to equation defined above, and therefore the values of the transfer function at the different sub-channel frequencies have to be estimated in order to use coherent detection. In addition, if there is a small timing offset, this will result in linearly increasing phase between the sub-channels. The channel estimator can estimate and compensate for these phase shifts since they can be regarded as a part of the channel transfer function.

The FFT block performs a DFT on the received OFDM symbol in order to retrieve the data symbols. These symbols are then passed to the modulation stage where the influence the channel is removed so that hard or soft decision can be made.

3.2 Single Carrier vs. Multi-Carrier Systems

The main difference a single carrier and a multi carrier system is how the influence of the channel is mitigated. In an OFDM system the receiver has to perform an FFT in order to demodulate the data, but the equalizer can on the other hand be really simple since a time dispersive channel only results in a multiplication by the value of the channel transfer function after the FFT. In a single carrier system, there is no FFT block in the receiver, but the equalizer is designed to mitigate the effects of ISI instead. The complexity of the equalizer can be high if the ISI spans over several symbols, but if it spans over only a few symbols the solution is straightforward. An estimate of the complexity of the equalizer is found by comparing the maximum excess delay of the channel to the symbol time, if the ISI spans over several symbols, then OFDM is to prefer to the use of long (and therefore complex) equalizer.

Another advantage of multi carrier modulation is that it is possible to use adaptive modulation on a sub-channel basis. This means that different modulation levels can be used on different sub-channel. On a “bad” sub-channel where the channel attenuation is high, a low level modulation scheme, such as binary PSK, can be used on a good sub-channel a higher order modulation, such as 16-QAM can be used. This means that the channel is more efficiently used and that the bit error rate can be decreased significantly compared to fixed modulation schemes. Note, however, that adaptive schemes require both extra hardware and control signaling.

The main disadvantage of using multi-carrier systems, OFDM systems in particular, are the sensitivity to frequency offsets and the large dynamic range of the desired signal. The sensitivity to frequency offsets and the large dynamic range of the signal to be transmitted. Frequency offsets cause both ICI and attenuation of the desired signal. The sensitivity to frequency offsets can be explained by the large frequency side lobes of the signal and the sensitivity can be decreased by using a window function with a more attractive frequency characteristics compared to the one resulting from the rectangular time domain window. The large dynamic range can be understood by looking at the OFDM signal as a sum of N sinusoids. Sometimes the waveforms add constructively, which means that a peak is produced. For most of the time however, the waveforms have different signs such that the sum achieves low amplitude.

3.4 Why Use Multiple Carriers

The use of multipath carriers follows from the presence of significant levels of multipath. Suppose we modulate a carrier with digital information. During each symbol, we transmit the carrier with a particular phase and amplitude which is chosen from the constellation in use. Each symbol conveys a number of bits of information, equal to the logarithm of the number of different states in the constellation.

Now imagine that the signal is received via two paths, with a relative delay between them. Taking transmitted symbol n as an example, the receiver will attempt to demodulate the data that was sent in this symbol by examining all the received information relating to symbol n , both the directly received information and the delayed information.

When the relative delay is more than one symbol period, the signal received via the second path acts purely as an interference, since it only carries information belonging to a previous symbol or symbols. Such ISI implies that only very small levels of the delayed signal can be tolerated.

When the relative delay is less than one symbol period, part of the signal received via the second path acts purely as interference, since it only carries information belonging to the previous symbol, but may act constructively or destructively to the main path information.

This tells us that, if we are to cope with any appreciable level of delayed signals, the symbol rate must be reduced sufficiently so that the total delay spread (between the first and last received path) is only a modest fraction of the symbol period. The information that can be carried by a single carrier is thus limited in the presence of multipath. If one carrier then cannot carry the information rate we require, this leads naturally to the idea of dividing the high-rate data into many low-rate parallel streams, each conveyed by its own carrier, of which there are a large number. This is a form of FDM, the first step towards COFDM.

Even when the delay spread is less than one symbol period, a degree of ISI from the previous symbol remains. This could be eliminated if the period for which each symbol is transmitted were made longer than the period over which the receiver integrates the signal, a first indication that adding a guard interval may be a good thing.

3.5 Orthogonality and the Use of FFT

The use of very large number of carriers is a prospect which is particularly daunting, surely we would need many modulators/demodulators and filters to accompany them? it would also appear that an increase of bandwidth would be required to accommodate them. Both these worries can fortunately be dispelled if we do one simple thing: we specify that the carriers are evenly spaced by precisely $f_u = 1/T_u$ where T_u is the period over which the receiver integrates the demodulated signal. When we do this, the carriers form what mathematicians call an orthogonal set.

The k^{th} carrier can be written as:

$$\Psi_k(t) = e^{jk\omega_u t}$$

Where $\omega_u = 2\pi/T_u$, and the orthogonality condition that the carrier satisfy is:

$$\int_{\tau}^{\tau+T_u} \Psi_k(t) \Psi_l^*(t) dt = 0, \quad k \neq l$$

$$\int_{\tau}^{\tau+T_u} \Psi_k(t) \Psi_k^*(t) dt = T_u, \quad k = l$$

more intuitively, what this represent is the common procedure of demodulating a carrier by means of multiplying it by a carrier of the same frequency (“beating it down to zero frequency”) and than integrating the result. Any other carriers will give rise to “beat tones” which are at integral multiples of ω_u . All of these unwanted tones therefore have an integer numbers of cycles during the integration period T_u , and thus integrate to zero. Hence without any explicit filtering, we can separately demodulate all the carriers without any mutual crosstalk, just by our particular choice of carrier spacing. Furthermore, we have not wasted any spectrum either. The carriers are closely packed so that they occupy the same spectrum in total as would a single carrier.

3.5.1 Preserving the Orthogonality

In practice, our carriers are modulated by complex numbers which change from symbol to symbol. If the integration period spans two symbols (as for the delayed paths), not only will there be same carrier ISI, but in addition there will be ICI as well. This happens because the beat tones from other carriers may no longer integrate to zero if they change in phase and/or amplitude during the period. We avoid this by adding a guard interval, which ensures that all the information integrated comes from the same symbol and appears constant during it.

Now if we add a guard interval, than the symbol period is extended so it exceeds the receiver integration period T_u . Since all the carriers are cyclic within T_u , so too is the whole modulation signal. Thus the segment added at the beginning, of the symbol to form the guard interval is identical to the segment of the same length at the end of the symbol. As long as the delay of any path with respect to the main path is less than the guard

interval, all the signal component within the integration period come from the same symbol and the orthogonality criterion is satisfied. ICI and ISI will only occur when the relative delay exceeds the guard interval.

The length of guard interval is so chosen to match the level of multipath expected. It should not too large a fraction of T_u , otherwise too much data capacity (and spectral efficiency) will be sacrificed. To avoid very long delays, T_u must therefore be made large, implying a large number of carriers, from hundred to thousands.

3.5.2 Use of FFT

We have avoided thousands of filters, thanks to orthogonality, what about implementing all the demodulating carriers, multipliers and integrators.

In practice, we work with the received signal in sampled form, (samples, above the Nyquist limit, off course). The process of integration then becomes one of summation, and the whole demodulation process takes on a form which is identical to the discrete fourier transform (DFT). Fortunately, we already have efficient algorithms of FFT, so that we are able to built laboratory COFDM equipments quite easily. Common versions of FFT operates on a group of 2^M time samples and deliver the same number of frequency coefficients. The inverse FFT is similarly used in the transmitter to generate the OFDM signal from the input data.

3.6 *Cyclic OFDM Symbol Extension*

Using frequency division or orthogonal multiplexing (OFDM) two different sources of interference can be identified. ISI is defined as the crosstalk between signals within the sub-channel of consecutive FFT frames, which are separated in time by symbol time interval of duration T. Inter channel interference is the cross talk between adjacent sub-channels or frequency slots of the same FFT frame. Since the effects of these interference sources and there mitigation effects are similar, it is convenient to introduce the term multidimensional interference (MDI).

A very nice approach to combat the defined MDI is to transmit quasi periodically extended time domain blocks, i.e. OFDM symbols, after modulation by IFFT. The length

of the added quasi periodic extension depends on the memory length of the channel, in other words, on the length of the transient response of the channel to the quasi periodic excitation constituted by the modulated signal.

Every block of length T modulated signal segment is quasi periodically extended by a length of T_T transient duration simply repeating N_T samples, corresponding to the duration of $(T + T_T)$. Trailing and leading samples of this extended block are corrupted by the channel's transient response, hence the receiver is instructed to ignore the first j number of samples of the received block and also disregard $(M + N_T - j)$ trailing samples. Only the central M number of samples are demodulated by FFT at the receiver, which are essentially unaffected by the channel's transient response.

The number of extension samples N_T required depends on the length of the channel's transient response and the number of modulation levels. If the number of modulation levels is high, the maximum acceptable MDI due to channel transients must be kept low in order to maintain sufficient noise margins before the data is corrupted. This then requires a large quasi-periodic extension, i.e. N_T must also be higher. It must also be appreciated that the MDI-corrupted extension actually wastes channel capacity as well as transmitted power. However, if the useful information blocks are long, i.e. $M \geq 128$, the extension length can be kept as low as 10% of the useful information block length.

3.7 Inherent Difficulties Regarding

OFDM

OFDM transmission over mobile communication channels can alleviate the problem of multipath propagation, recent research efforts have focused on solving a set of inherent difficulties regarding OFDM, namely the peak to mean power ratio, time and frequency synchronization, and on mitigating the effects of frequency selective fading channel.

3.7.1 Peak to mean power ratio

OFDM has several properties which make it an attractive modulation scheme for high speed transmission links. However, one major difficulty is its large peak to average power ratio (PAPR). These large peaks cause saturation in power amplifiers, leading to intermodulation products among the sub carriers and distributing out of band energy. Therefore, it is desirable to reduce the PAPR. To reduce the PAPR, several techniques have been proposed such as clipping, coding, peak windowing, tone reservation and tone injection. But, most of these methods are unable to achieve simultaneously a large reduction in PAPR, with low complexity, with low coding overhead, without performance degradation and without transmitter receiver symbol handshake. The complex envelope of the OFDM signal, consisting of N carriers is given by:

$$S_{total} = \sum_{k=-\infty}^{\infty} \sum_{n=0}^{N-1} a_{n,k} \cdot g(t - kT) \cdot e^{jn\frac{2\pi}{T}t}$$

where $g(t)$ is rectangular pulse of duration T and T is OFDM symbol duration.

Peak to Average power ratio is defined by:

$$PAPR = \frac{\max |s(t)|^2}{\mathcal{E}\{|s(t)|^2\}}$$

where $\mathcal{E}\{\cdot\}$ denotes the expectation. From the central limit theorem, for large values of N , the real and imaginary values of $s(t)$ become Gaussian distributed. The amplitude of the OFDM signal therefore has a Rayleigh distribution with zero mean and a variance of N times the variance of one complex sinusoid.

Assuming the samples to be mutually uncorrelated, the cumulative distribution function for the peak power per OFDM symbol, can be given by (Muller and Huber, 1997),

$$P_r\{PAPR > \gamma\} = \left(1 - (1 - e^{-\gamma})^N\right)$$

From this, it is seen that large PAPR occurs only infrequently.

In the literature, two kinds of approaches are investigated which assure that the transmitted OFDM signal does not exceed the amplitude A_o if a given back off is used.

- The first method make use of redundancy in such a way that any data sequence leads to the magnitude of OFDM signal greater than A_o or that at least the probability of higher amplitude peaks is greatly reduced. This approach does not result in interference of the OFDM signal.
- In the second approach, the OFDM signal is manipulated by a correcting function, which eliminates the amplitude peaks. The out of band interference caused by the correcting function is zero or negligible. However, interference of the OFDM signal itself is tolerated to a certain extent.

Peak to power ratio effect is described by the Peak Envelop and Crest factor, which is defined as the square root of PAPR. Some significant methods to reduce PAPR are described as:

- a) Block Coding
- b) Use of M-Sequences
- c) Selective Scrambling
- d) Clipping and Filtering
- e) Peak Windowing
- f) Tone reservation

3.7.2 Synchronization

Time and frequency synchronization between the transmitter and receiver are of crucial importance as regards to the performance of an OFDM link [18]. A wide variety of techniques have been proposed for estimating and correcting both timing and carrier frequency offsets at the OFDM receiver. Fine frequency and timing tracking algorithms exploiting the OFDM signal's cyclic extension were published by Moose [20], Daffara [21] and Sandell [21].

Normally the most common scheme for synchronization is the use of superimposed pilot sequences. The advantage of using superimposed pilots for this purpose is that a continuous sequence is present that can be used for synchronization purposes. This means that the frequency offset range can be increased compared to methods where cyclic prefix can be used for synchronization. Further, the time synchronization signal shows a more

attractive behavior since it has a sharper peak compared to the methods incorporating a cyclic prefix. This technique is suitable for both tracking and acquisition and it is very flexible in that the power of the pilot sequence can be adjusted to the specific needs. In acquisition mode all the transmitted power can be devoted pilot sequence and then the transmitted signal is equivalent to the preamble described by Fredrik Tufvesson ("Preamble based Frequency synchronization in OFDM based systems). Now after reviewing the literature regarding synchronization and channel estimation, I think it is well worth spending a few percent of the transmitted power and bandwidth on known pilot symbols. The loss caused by the pilot symbols is often small compared to the gain that can be achieved, so pilot symbols are generally a good investment.

Chapter 4

Coding

Channel coding refers to the class of signal transformations designed to improve communications performance, by enabling the transmitted signals to better withstand the effects of various channel impairments, such as noise, fading and jamming. Usually, the goal of channel coding is to reduce the probability of bit error, or to reduce the required E_b/N_o , at the cost of expanding more bandwidth than would otherwise be necessary. Since the early days of information and coding theory the goal has always been to come close to the Shannon limit performance with a tolerable complexity. The results achieved so far shows that it is relatively easy to operate at signal to noise ratios above the values determined by the channel cut off. Basically the use of Large Scale Integration (LSI) circuits has made it possible to achieve as much as 8-dB performance improvement through coding, at much less cost than through the use of other methods such as higher power transmitters or larger antennas.

Channel coding can be partitioned into two study areas:

- *Waveform Coding*: deals with transforming waveforms into “better” waveforms, to make the detection process less subjective to errors.
- *Structured Coding*: deals with transforming data sequences into “better” sequences, having structured redundancy. The redundant bits can then be used for the detection and correction of errors.

Since we are dealing with the coding techniques used in OFDM transmission, so our main focus is on structured coding. However, here I provide a very brief overview of waveform coding.

4.1 Waveform Coding

Waveform coding procedures transform a waveform set into an improved waveform set. The improved waveform set can then be used to provide improved P_B compared to the original set. The most popular of such waveform codes are referred as *orthogonal* and *biorthogonal* codes. The coding procedure endeavors to make each of the waveforms in the coded signal set as unlike as possible, or in other words the cross correlation coefficient ρ_{xy} should be as small as possible. The smallest possible value of ρ_{xy} (i.e. $\rho_{xy} = -1$) occurs when the signals are antipodal. In general, it is possible to make all the

cross-correlation coefficients equal to zero. The set is then said to be orthogonal. Antipodal signal sets are optimum in the sense, that each signal in the set is most distant from the other signal in the set. Compared to antipodal signals, the distance properties of orthogonal sets can be thought of as “second best”.

4.2 Structured Sequence Coding

Now as we know in M-ary digital signaling, each waveform contains k bits of information and we have M waveforms such that $M = 2^k$. Now as we know, in case of orthogonal M-ary signaling, we can decrease probability of error P_B by increasing M (expanding the bandwidth). Similarly, it is possible to decrease P_B by encoding k binary digits into one of M orthogonal codewords. The major disadvantage with such orthogonal coding technique is the associated inefficiency use of bandwidth. The required transmission bandwidth grows exponentially with k for an orthogonal set of waveforms, means as we will increase the value of k , the probability of error decreases, but with the expense of increase in the bandwidth. Structured sequence coding is partitioned into two important categories:

- ✓ Convolutional Codes
- ✓ Block Codes

However, recently, new coding scheme, named Turbo Codes, are seen to be more effective in number of situations. Turbo codes, first presented to the coding community in 1993 [23], represents the most important breakthrough in coding since Ungerboeck introduced Trellis codes in 1982 [24].

These techniques allow us to attain a P_B performance comparable to waveform coding techniques but with a lower bandwidth requirements. The codewords of these codes are usually nonorthogonal.

4.3 Code Rate and Redundancy

In the case of block codes, the source data are segmented into blocks of k data bits, also called information bits or message bits, each block can represent any one of 2^k distinct messages. The encoder transforms each k -bit data blocks into a larger block of n bits,

called code bits or channel symbols. The $(n - k)$ bits which the encoder adds to each data block, are redundant bits, parity bits, or check bits, they carry no new information. The code is referred to as an (n, k) code. The ratio of redundant bits to data bits, $(n - k)/k$, within a clock is referred the redundancy of code, and the ratio of data bits to total bits, k/n , is called the code rate. The code rate can be thought of as the portion of a code bit that constitutes information. For example, in a rate $\frac{1}{2}$ code, each code bit carries $\frac{1}{2}$ bit of information.

4.4 Coding Gain

Figure 4.1 shows the probability of bit error, P_B versus E_b/N_o for coherent binary PSK modulation in combination with examples of various (n, k) codes over a Gaussian Channel. The (1,1) curve illustrates the uncoded PSK performance, while the other curves illustrates the coded PSK performance using block codes. From the curves we can see clearly, that they move in the wrong direction when compared with the uncoded curve.

The curves in figure 4.1 indicates that the strength of a code is seen only after an E_b/N_o threshold has been exceeded. For values of E_b/N_o less than this threshold, the coding manifests itself only as overhead bits resulting in reduced energy per bit, compared to the uncoded case, before the threshold is exceeded, the redundant bits are simply “excess baggage” without any ability to improve performance. *Code Gain* is defined as the reduction, expressed in decibels, in the required E_b/N_o to achieve a specified error performance of an error-correcting coded system over an uncoded one with the same modulation.

4.5 Linear Block Codes

Linear block codes are generally characterized by the (n, k) notation as described in section 4.3.2. The encoder transforms a block of k messages into a longer block of n code word digits, constructed from a given alphabet of elements. The k bit message from 2^k distinct message sequence referred as k -tuples. The n -bit block can form as many as

2^n distinct sequences, referred to as n-tuples. The encoding procedure assigns to each of the 2^k message k-tuples, one of the 2^n tuples. A block code represents a one-to-one assignment, whereby the 2^k codeword n-tuples; the mapping can be accomplished via a look-up table.

4.5.1 Vector Spaces

The set of all binary n-tuples, V_n , is called a vector space over the binary field of two elements (0 and 1). The binary field has two operations, addition and multiplication such that the results of all operations are in the same set of two elements. The arithmetic operations of addition and multiplication are defined by the conventions of algebraic field.

4.5.2 Vector Subspaces

A subset S of the vector space V_n is called a subspace if the following two conditions are met:

- (a) The all-zeros vector is in S.
- (b) The sum of any two vectors in S is also in S.

These properties are fundamental for the algebraic characterization of the linear block codes. Suppose that V_i and V_j are two code vectors, in an (n, k) binary block code. The code is said to be linear if and only if, $(V_i \oplus V_j)$ is also a code vector. A linear block code, then, is one in which vectors outside the subspace cannot be created by the addition of legitimate code vectors (members of the subspace).

For example, consider a vector space of 16 4-tuples:

```

0000  0001  0010  0011  0100  0101  0110  0111
1000  1001  1010  1011  1100  1101  1110  1111

```

An example of a subset, that forms a subspace is:

```

0000  0101  1010  1111

```

It is easy to verify that the addition of any two vectors in the subspace can only yield one of the other members of the subspace. So in short, a set of 2^k n-tuples, is called a linear block code, if and only if, it is a subspace of the vector space V_n of all n-tuples.

4.5.3 A (6,3) Linear Block Code Example

Consider a (6,3) code which have $2^k = 2^3 = 8$, message vectors, and therefore eight code

<u>MESSAGE VECTOR</u>	<u>CODE VECTOR</u>
000	000000
100	110100
010	011010
110	101110
001	101001
101	011101
011	110011
111	000111

vectors. There are $2^n = 2^6 =$ sixty-four 6-tuples in the V_6 vector space. It is quite easy to check that eight code vectors shown above form a subspace of V_6 (that is the sum of any two code vectors yields another code vector member of the subspace). Therefore, these code represents a linear block code.

But for larger values of k, the use of lookup table becomes prohibitive. Because for larger values of k, the size of lookup table is very high, and hence a large amount of memory is required to store such a matrix. Fortunately, it is possible to reduce complexity by generating the required code vectors as needed, instead of storing them. In general, we can define a generator matrix of order $k \times n$ as,

$$G = \begin{bmatrix} V_1 \\ V_2 \\ V_3 \end{bmatrix} = \begin{bmatrix} v_{11} & v_{12} & v_{13} \\ v_{21} & v_{22} & v_{23} \\ v_{31} & v_{32} & v_{33} \end{bmatrix}$$

Code vectors, as convention, are designated as row vectors. Thus the message m, a sequence of k message bits, is shown below as:

$$m = m_1, m_2, \dots, m_k$$

So if U is the code vector, than in matrix notation, we can write it as:

$$U = mG$$

So for the above example, the matrix G is as:

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

Now the code vector for the message 110 is given simply as:

$$U = [1 \quad 1 \quad 0] \begin{bmatrix} V_1 \\ V_2 \\ V_3 \end{bmatrix} = 1.V_1 + 1.V_2 + 0.V_3$$

which yields 101110.

The most common block codes are:

- ✓ Hamming Codes
- ✓ Golay Code
- ✓ BCH Codes
- ✓ Reed-Solomon Codes

4.6 Convolutional Codes

This chapter describes the encoder and decoder structures for convolutional codes. The encoder will be represented in many different but equivalent ways. Also, the main encoding strategy for convolutional codes, based on the Viterbi Algorithm, will be described. A firm understanding of convolutional codes is an important prerequisite to the understanding of turbo codes.

4.6.1 Encoder Structure

A convolutional code introduces redundant bits into the data stream through the use of linear shift registers as shown in Figure 4.2.

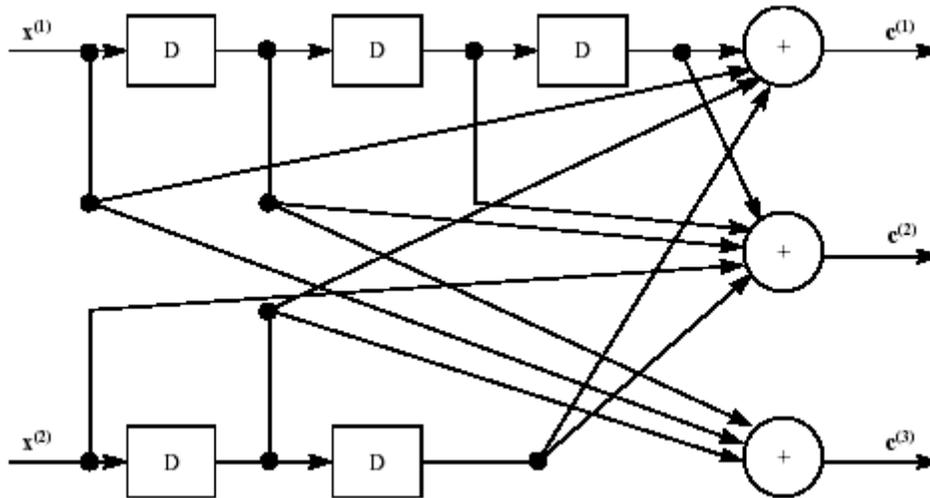


Figure 4.2: Example convolutional encoder where $x(i)$ is an input information bit stream and $c(i)$ is an output encoded bit stream

The information bits are input into shift registers and the output encoded bits are obtained by modulo-2 addition of the input information bits and the contents of the shift registers. The connections to the modulo-2 adders were developed heuristically with no algebraic or combinatorial foundation.

The code rate r for a convolutional code is defined as

$$r = \frac{k}{n}$$

where k is the number of parallel input information bits and n is the number of parallel output encoded bits at one time interval. The constraint length K for a convolutional code is defined as

$$K = m + 1$$

where m is the maximum number of stages (memory size) in any shift register. The shift registers store the state information of the convolutional encoder and the constraint length relates the number of bits upon which the output depends. For the convolutional encoder shown in Figure 2.1, the code rate $r=2/3$, the maximum memory size $m=3$, and the constraint length $K=4$.

A convolutional code can become very complicated with various code rates and constraint lengths. As a result, a simple convolutional code will be used to describe the code properties as shown in Figure 4.3.

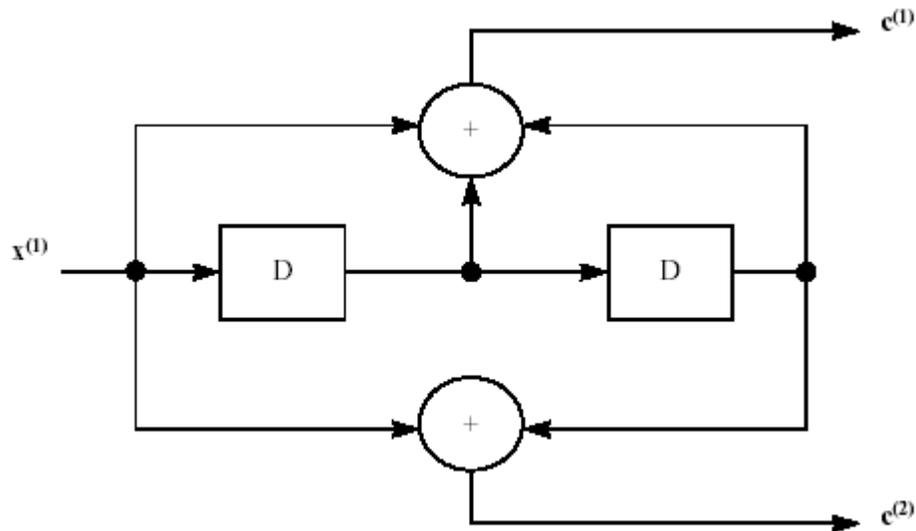


Figure 4.3: Convolutional encoder with $k=1$, $n=2$, $r=1/2$, $m=2$, and $K=3$.

4.6.2 Encoder Representations

The encoder can be represented in several different but equivalent ways. They are

1. Generator Representation
2. Tree Diagram Representation
3. State Diagram Representation
4. Trellis Diagram Representation

4.6.2.1 Generator Representation

Generator representation shows the hardware connection of the shift register taps to the modulo-2 adders. A generator vector represents the position of the taps for an output. A “1” represents a connection and a “0” represents no connection. For example, the two generator vectors for the encoder in Figure 2.2 are $g_1 = [111]$ and $g_2 = [101]$ where the subscripts 1 and 2 denote the corresponding output terminals.

4.6.2.2 Tree Diagram Representation

The tree diagram representation shows all possible information and encoded sequences for the convolutional encoder. Figure 4.4 shows the tree diagram for the encoder in Figure 2.2 for four input bit intervals.

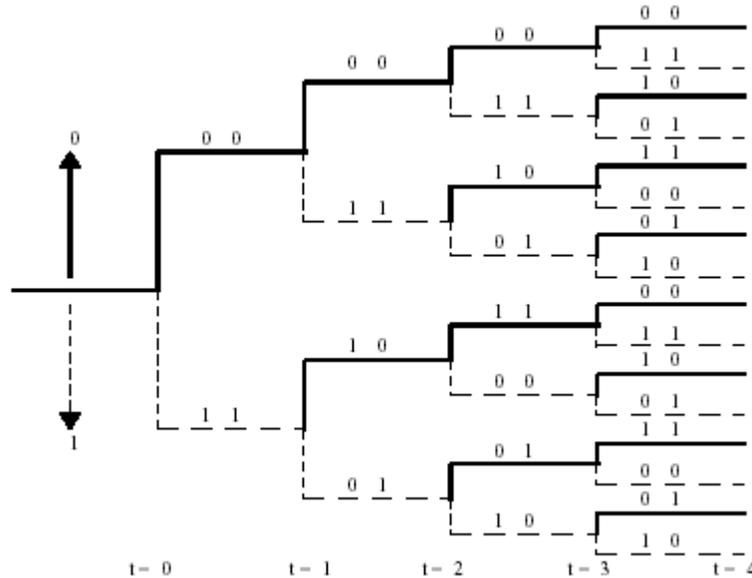


Figure 4.4: Tree diagram representation of the encoder in Figure 4.4 for four input bit intervals.

In the tree diagram, a solid line represents input information bit 0 and a dashed line represents input information bit 1. The corresponding output encoded bits are shown on the branches of the tree. An input information sequence defines a specific path through the tree diagram from left to right. For example, the input information sequence $\mathbf{x}=\{1011\}$ produces the output encoded sequence $\mathbf{c}=\{11, 10, 00, 01\}$. Each input information bit corresponds to branching either upward (for input information bit 0) or downward (for input information bit 1) at a tree node.

4.6.2.3 State Diagram Representation

The state diagram shows the state information of a convolutional encoder. The state information of a convolutional encoder is stored in the shift registers. Figure 4.5 shows the state diagram of the encoder in Figure 4.2.

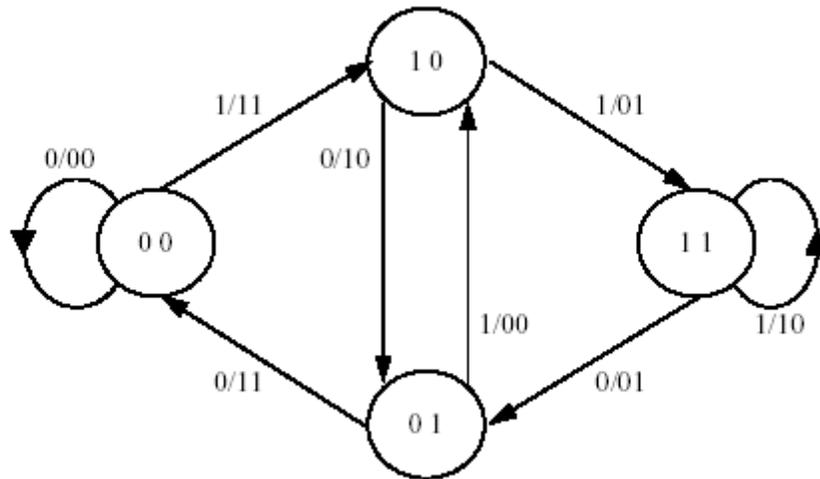


Figure 4.5: State diagram representation of the encoder in Figure 4.3.

In the state diagram, the state information of the encoder is shown in the circles. Each new input information bit causes a transition from one state to another. The path information between the states, denoted as x/c , represents input information bit x and output encoded bits c . It is customary to begin convolutional encoding from the all zero state. For example, the input information sequence $\mathbf{x}=\{1011\}$ (begin from the all zero state) leads to the state transition sequence $\mathbf{s}=\{10, 01, 10, 11\}$ and produces the output encoded sequence $\mathbf{c}=\{11, 10, 00, 01\}$. Figure 4.6 shows the path taken through the state diagram for the given example.

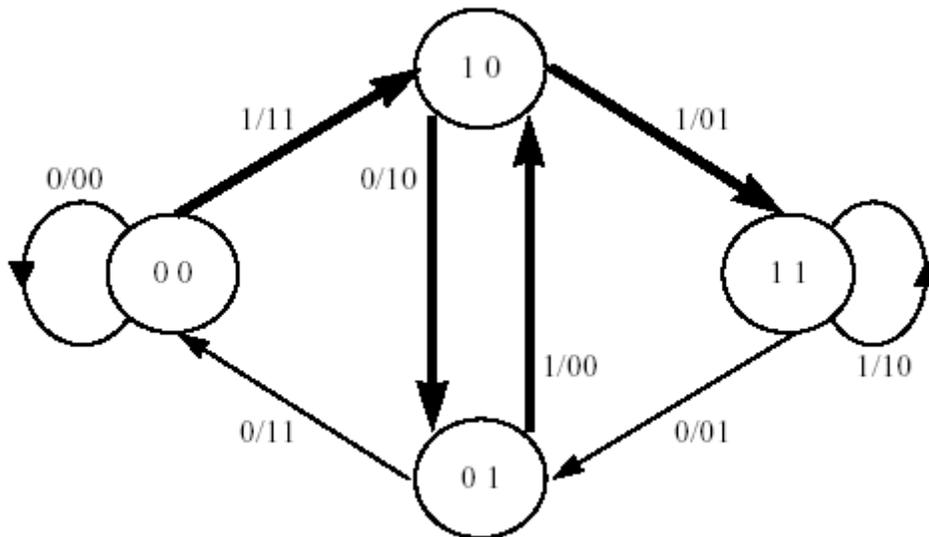


Figure 4.6: The state transitions (path) for input information sequence {1011}.

4.6.2.4 Trellis Diagram Representation

The trellis diagram is basically a redrawing of the state diagram. It shows all possible state transitions at each time step. Frequently, a legend accompanies the trellis diagram to show the state transitions and the corresponding input and output bit mappings (x/c). This compact representation is very helpful for decoding convolutional codes as discussed later. Figure 4.7 shows the trellis diagram for the encoder in Figure 2.2.

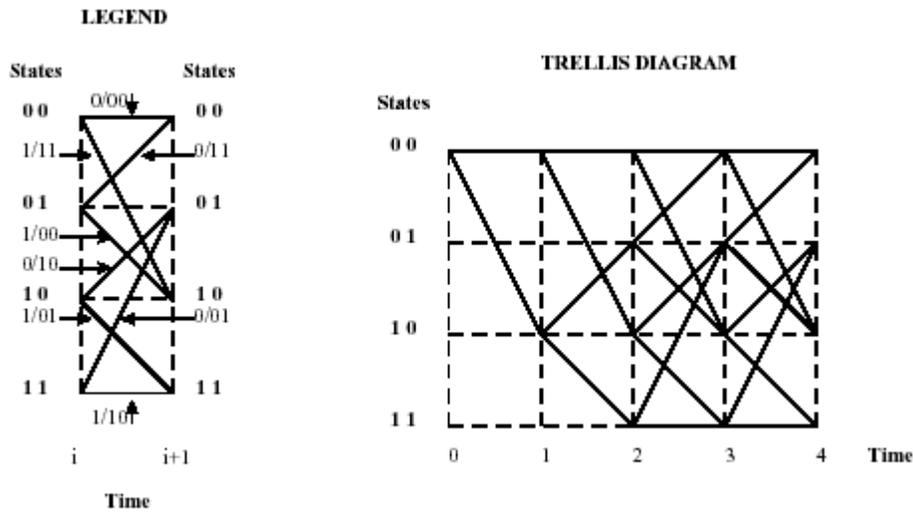


Figure 4.7: Trellis diagram representation of the encoder in Figure 4.2 for four input bit intervals.

Figure 4.8 shows the trellis path for the state transitions in Figure 4.6.

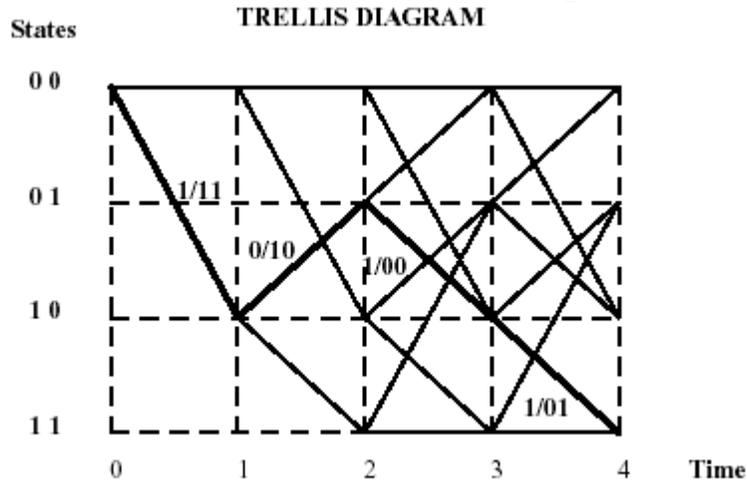


Figure 4.8: Trellis path for the state transitions in Figure 4.5.

4.6.3 Catastrophic Convolutional Code

Catastrophic convolutional code causes a large number of bit errors when only a small number of channel bit errors is received. This type of code needs to be avoided and can be identified by the state diagram. A state diagram having a loop in which a nonzero information sequence corresponds to an all-zero output sequence identifies a catastrophic convolutional code. Figure 4.9 shows two examples of such code.

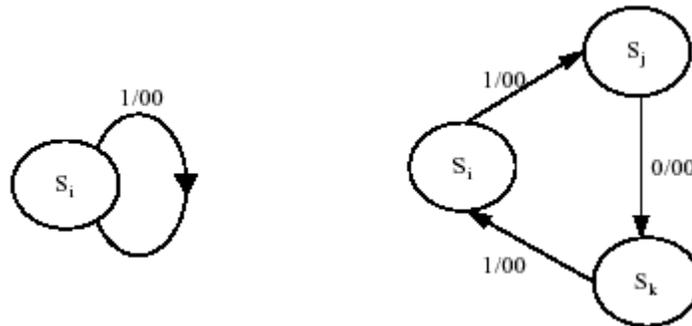


Figure 4.9: Examples of catastrophic convolutional code.

4.6.4 Hard-Decision and Soft-Decision

Decoding

Hard-decision and soft-decision decoding refer to the type of quantization used on the received bits. Hard-decision decoding uses 1-bit quantization on the received channel values. Soft-decision decoding uses multi-bit quantization on the received channel values. For the ideal soft-decision decoding (infinite-bit quantization), the received channel values are directly used in the channel decoder. Figure 4.10 shows hard- and softdecision decoding.

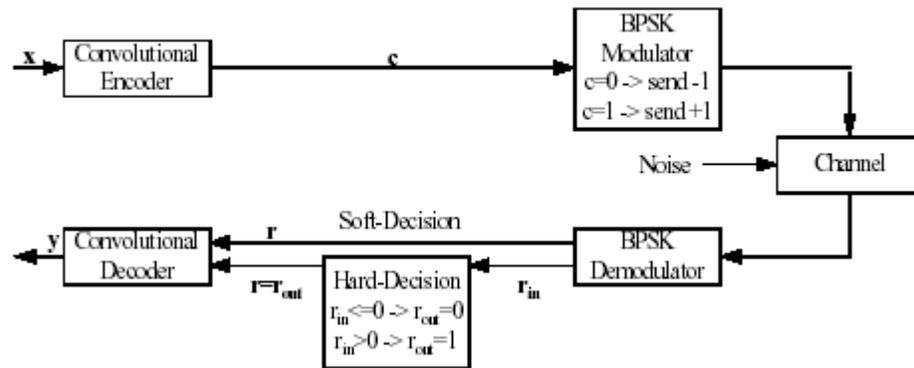


Figure 4.10: Hard- and Soft-decision decoding [Woe94].

4.6.4.1 Hard-Decision Viterbi Algorithm

For a convolutional code, the input sequence x is “convoluted” to the encoded sequence c . Sequence c is transmitted across a noisy channel and the received sequence r is obtained. The Viterbi algorithm computes a maximum likelihood (ML) estimate on the estimated code sequence y from the received sequence r such that it maximizes the probability $p(r|y)$ that sequence r is received conditioned on the estimated code sequence y . Sequence y must be one of the allowable code sequences and cannot be any arbitrary sequence. Figure 4.11 shows the described system structure.

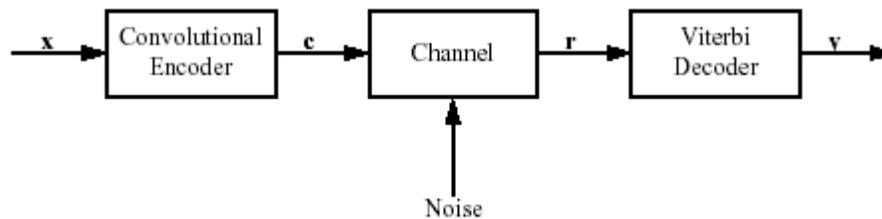


Figure 4.11: Convolutional code system.

For a rate r convolutional code, the encoder inputs k bits in parallel and outputs n bits in parallel at each time step. The input sequence is denoted as

$$\mathbf{x}=(x_0(1), x_0(2), \dots, x_0(k), x_1(1), \dots, x_1(k), x_{L+m-1}(1), \dots, x_{L+m-1}(k))$$

And the coded sequence is denoted as,

$$\mathbf{c}=(c_0(1), c_0(2), \dots, c_0(n), c_1(1), \dots, c_1(n), c_{L+m-1}(1), \dots, c_{L+m-1}(n))$$

Where L denotes the length of input information sequence and m denotes the maximum Length of the shift registers. Additional m zero bits are required at the tail of the

information sequence to take the convolutional encoder back to the all-zero state. It is required that the encoder start and end at the all-zero state. The subscript denotes the time index while the superscript denotes the bit within a particular input k-bit or output n bit block. The received and estimated sequences \mathbf{r} and \mathbf{y} can be described similarly as

$$\mathbf{r}=(r_0(1), r_0(2), \dots, r_0(n), r_1(1), \dots, r_1(n), r_{L+m-1}(1), \dots, r_{L+m-1}(n))$$

and,

$$\mathbf{y}=(y_0(1), y_0(2), \dots, y_0(n), y_1(1), \dots, y_1(n), y_{L+m-1}(1), \dots, y_{L+m-1}(n))$$

For ML decoding, the Viterbi algorithm selects \mathbf{y} to maximize $p(\mathbf{r}|\mathbf{y})$. The channel is assumed to be memoryless, and thus the noise process affecting a received bit is independent from the noise process affecting all of the other received bits.

From probability theory, the probability of joint, independent events is equivalent to the product of the probabilities of the individual events. Thus,

$$\begin{aligned} p(\mathbf{r}|\mathbf{y}) &= \prod_{i=0}^{L+m-1} [P(r_i^{(1)}|y_i^{(1)})P(r_i^{(2)}|y_i^{(2)})\dots P(r_i^{(n)}|y_i^{(n)})] \quad [\text{Wic95}] \\ &= \prod_{i=0}^{L+m-1} \left(\prod_{j=1}^n P(r_i^{(j)}|y_i^{(j)}) \right) \quad [\text{Wic95}] \end{aligned}$$

This equation is called the likelihood function of \mathbf{y} given that \mathbf{r} is received [Vit71]. The estimate that maximizes $p(\mathbf{r}|\mathbf{y})$ also maximizes $\log p(\mathbf{r}|\mathbf{y})$ because logarithms are monotonically increasing functions.

$$\log p(\mathbf{r}|\mathbf{y}) = \sum_{i=0}^{L+m-1} \left(\sum_{j=1}^n \log P(r_i^{(j)}|y_i^{(j)}) \right) \quad [\text{Wic95}]$$

For an easier manipulation of the summations over the log function, a bit metric is defined. The bit metric is defined as

$$\log p(\mathbf{r}|\mathbf{y}) = \sum_{i=0}^{L+m-1} \left(\sum_{j=1}^n \log P(r_i^{(j)}|y_i^{(j)}) \right) \quad [\text{Wic95}]$$

where a and b are chosen such that the bit metric is a small positive integer [Wic95]. The values a and b are defined for binary symmetric channel (BSC) or hard-decision decoding. Figure 4.12 shows a BSC.

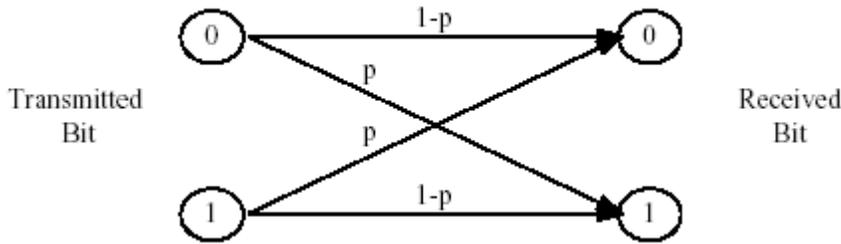


Figure 4.12: The binary symmetric channel model, where p is the crossover probability.

For BSC, a and b can be chosen in two distinct ways. For the conventional way, they can be chosen as

$$a = \frac{1}{\log p - \log(1-p)} \quad [\text{Wic95}]$$

and,

$$b = -\log(1-p) \quad [\text{Wic95}]$$

The resulting bit metric is then

$$M(r_i^{(j)}|y_i^{(j)}) = \frac{1}{[\log p - \log(1-p)]} [\log p(r_i^{(j)}|y_i^{(j)}) - \log(1-p)]$$

From the BSC model, it is clear that $p(r_i^{(j)}|y_i^{(j)})$ can only take on values p and $1-p$. Table 2.1 shows the resulting bit metric.

$M(r_i^{(j)} y_i^{(j)})$	Received Bit $r_i^{(j)} = 0$	Received Bit $r_i^{(j)} = 1$
Decoded Bit $y_i^{(j)} = 0$	0	1
Decoded Bit $y_i^{(j)} = 1$	1	0

Table 4.1: Conventional Bit Metric Values

This bit metric shows the cost of receiving and decoding bits. For example, if the decoded bit $y_i^{(j)} = 0$, and the received bit $r_i^{(j)} = 0$, then the cost $M(r_i^{(j)}|y_i^{(j)}) = 0$. However, if the decoded bit $y_i^{(j)} = 0$ and the received bit $r_i^{(j)} = 1$, then the cost $M(r_i^{(j)}|y_i^{(j)}) = 1$. As it can be seen, this is related to the Hamming distance and is known as the Hamming

distance metric. Thus, the Viterbi algorithm chooses the code sequence \mathbf{y} through the trellis that has the smallest cost/Hamming distance relative to the received sequence \mathbf{r} .

Alternatively, a and b can be chosen as

$$a = \frac{1}{\log(1-p) - \log p} \quad [\text{Wic95}]$$

and,

$$b = -\log p \quad [\text{Wic95}]$$

The resulting alternative bit metric is then,

$$M(r_i^{(j)}|y_i^{(j)}) = \frac{1}{[\log(1-p) - \log p]} [\log p(r_i^{(j)}|y_i^{(j)}) - \log p]$$

Table 4.2 shows the resulting alternative bit metric.

$M(r_i^{(j)} y_i^{(j)})$	Received Bit $r_i^{(j)} = 0$	Received Bit $r_i^{(j)} = 1$
Decoded Bit $y_i^{(j)} = 0$	1	0
Decoded Bit $y_i^{(j)} = 1$	0	1

Table 4.2: Alternative Bit Metric Values

For this case, the Viterbi algorithm chooses the code sequence \mathbf{y} through the trellis that has the largest cost/Hamming distance relative to the received sequence \mathbf{r} . Furthermore, for an arbitrary channel (not necessarily BSC), the values a and b are found on a trial-and-error basis to obtain an acceptable bit metric.

From the bit metric, a path metric is defined. The path metric is defined as

$$M(\mathbf{r}|\mathbf{y}) = \sum_{i=0}^{L+m-1} \left(\sum_{j=1}^n M(r_i^{(j)}|y_i^{(j)}) \right) \quad [\text{Wic95}]$$

and indicates the total cost of estimating the received bit sequence \mathbf{r} with the decoded bit sequence \mathbf{y} in the trellis diagram. Furthermore, the k th branch metric is defined as

$$M(\mathbf{r}_k|\mathbf{y}_k) = \sum_{j=1}^n M(r_k^{(j)}|y_k^{(j)}) \quad [\text{Wic95}]$$

and the k th partial path metric is defined as

$$\begin{aligned}
 M^k(\mathbf{r}|\mathbf{y}) &= \sum_{i=0}^k M(\mathbf{r}_i|\mathbf{y}_i) \quad [\text{Wic95}] \\
 &= \sum_{i=0}^k \left(\sum_{j=1}^n M(r_i^{(j)}|y_i^{(j)}) \right) \quad [\text{Wic95}]
 \end{aligned}$$

The k th branch metric indicates the cost of choosing a branch from the trellis diagram. The k th partial path metric indicates the cost of choosing a partially decoded bit sequence \mathbf{y} up to time index k . The Viterbi algorithm utilizes the trellis diagram to compute the path metrics. Each state (node) in the trellis diagram is assigned a value, the partial path metric. The partial path metric is determined from state $s = 0$ at time $t = 0$ to a particular state $s = k$ at time $t \geq 0$. At each state, the “best” partial path metric is chosen from the paths terminated at that state [Wic95]. The “best” partial path metric may be either the larger or smaller metric, depending whether a and b are chosen conventionally or alternatively.

The selected metric represents the survivor path and the remaining metrics represent the nonsurvivor paths. The survivor paths are stored while the nonsurvivor paths are discarded in the trellis diagram. The Viterbi algorithm selects the single survivor path left at the end of the process as the ML path. Trace-back of the ML path on the trellis diagram would then provide the ML decoded sequence.

The hard-decision Viterbi algorithm (HDVA) can be implemented as follows [Rap96], [Wic95]:

$S_{k,t}$ is the state in the trellis diagram that corresponds to state S_k at time t . Every state in the trellis is assigned a value denoted $V(S_{k,t})$.

1. (a) Initialize time $t = 0$.
(b) Initialize $V(S_{0,0}) = 0$ and all other $V(S_{k,t}) = +\infty$.
2. (a) Set time $t = t+1$.
(b) Compute the partial path metrics for all paths going to state S_k at time t .

First, find the t^{th} branch metric,

$$M(\mathbf{r}_t|\mathbf{y}_t) = \sum_{j=1}^n M(r_t^{(j)}|y_t^{(j)}).$$

This is calculated, from the Hamming distance,

$$\sum_{j=1}^n |r_t^{(j)} - y_t^{(j)}|$$

Second, compute the t^{th} partial path metric,

$$M^*(\mathbf{r}|\mathbf{y}) = \sum_{i=0}^L M(\mathbf{r}_i|\mathbf{y}_i)$$

3. (a) Set $V(S_k, t)$ to the “best” partial path metric going to state S_k at time t .
Conventionally, the “best” partial path metric is the partial path metric with the smallest value.
- (b) If there is a tie for the “best” partial path metric, then any one of the tied partial path metric may be chosen.
4. Store the “best” partial path metric and its associated survivor bit and state paths.
5. If $t < L+m-1$, return to Step 2.

The result of the Viterbi algorithm is a unique trellis path that corresponds to the ML codeword.

A simple HDVA decoding example is shown below. The convolutional encoder used is shown in Figure 2.2. The input sequence is $\mathbf{x}=\{1010100\}$, where the last two bits are used to return the encoder to the all-zero state. The coded sequence is $\mathbf{c}=\{11, 10, 00, 10, 00, 10, 11\}$. However, the received sequence $\mathbf{r}=\{10, 10, 00, 10, 00, 10, 11\}$ has a bit error (underlined). Figure 4.13 shows the state transition diagram (trellis legend) of the example convolutional encoder.

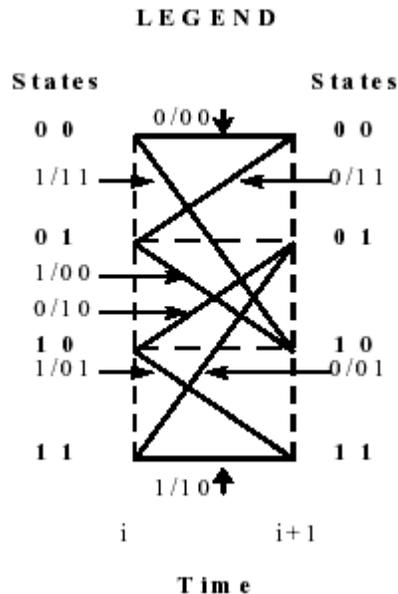


Figure 4.13: The state transition diagram (trellis legend) of the example convolutional encoder.

The state transition diagram shows the estimated information and coded bits along the branches (needed for the decoding process). HDVA decoding chooses the ML path through the trellis as shown in Figure 2.13. The chosen partial path (accumulated) metric for this example is the smallest Hamming distance and are shown in the figure for every node. The bold partial path metrics correspond to the ML path. Survivor paths are represented by bold solid lines and competing paths are represented by simple solid lines. For metric “ties”, the first branch is always chosen.

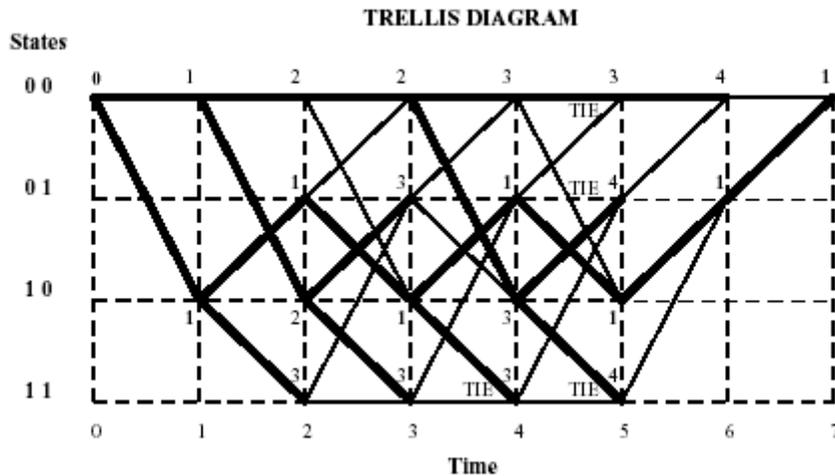


Figure 4.14: HDVA decoding of the example.

From the trellis diagram in Figure 4.14, the estimated code sequence is $\mathbf{y}=\{11, 10, 00, 10, 00, 10, 11\}$ which is the code sequence \mathbf{c} . Utilizing the state transition diagram in Figure 2.12, the estimated information sequence is $\mathbf{x}'=\{1010100\}$.

4.6.4.2 Soft-Decision Viterbi Algorithm

There are two general methods of implementing a soft-decision Viterbi algorithm. The first method (Method 1) uses Euclidean distance metric instead of Hamming distance metric. The received bits used in the Euclidean distance metric are processed by multi-bit quantization. The second method (Method 2) uses a correlation metric where its received bits used in this metric are also processed by multi-bit quantization.

4.6.5 Performance Analysis of Convolutional Code

The performance of convolutional codes can be quantified through analytical means or by computer simulation. The analytical approach is based on the transfer function of the convolutional code which is obtained from the state diagram. The process of obtaining the transfer function and other related performance measures are described below.

4.6.6 Transfer Function of Convolutional Code

The analysis of convolutional codes is generally difficult to perform because traditional algebraic and combinatorial techniques cannot be applied. These heuristically constructed codes can be analyzed through their transfer functions. By utilizing the state diagram, the transfer function can be obtained. With the transfer function, code properties such as distance properties and the error rate performance can be easily calculated. To obtain the transfer function, the following rules are applied:

1. Break the all-zero (initial) state of the state diagram into a start state and an end state. This will be called the modified state diagram.
2. For every branch of the modified state diagram, assign the symbol D with its exponent equal to the Hamming weight of the output bits.
3. For every branch of the modified state diagram, assign the symbol J .
4. Assign the symbol N to the branch of the modified state diagram, if the branch transition is caused by an input bit 1.

For the state diagram in Figure 4.5, the modified state diagram is shown in Figure 4.15.

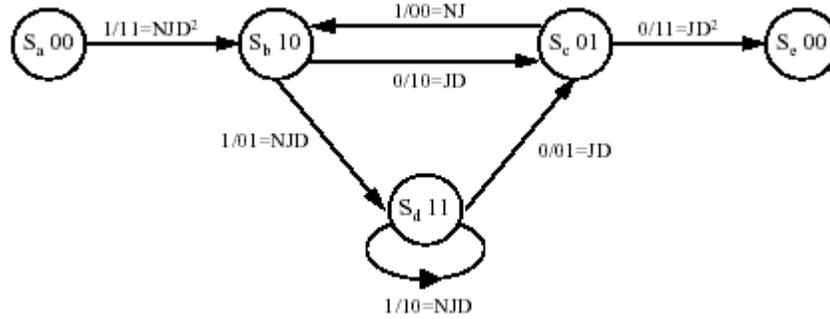


Figure 4.15: The modified state diagram of Figure 4.5 where S_a is the start state and S_e is the end state

Nodal equations are obtained for all the states except for the start state in Figure 4.15.

These results are

$$\begin{aligned} S_b &= NJD^2 S_a + NJS_c \\ S_c &= JDS_b + JDS_d \\ S_d &= NJDS_b + NJDS_d \\ S_e &= JD^2 S_c \end{aligned}$$

The transfer function is defined to be

$$T(D, N, J) = \frac{S_{end}(D, N, J)}{S_{start}(D, N, J)}$$

and for Figure 4.14,

$$T(D, N, J) = \frac{S_e}{S_a}$$

By substituting and rearranging,

$$\begin{aligned} T(D, N, J) &= \frac{NJ^3 D^5}{1 - (NJ + NJ^2)D} && \text{(closed form)} \\ &= NJ^3 D^5 + (N^2 J^4 + N^2 J^5)D^6 + (N^3 J^5 + 2N^3 J^6 + N^3 J^7)D^7 + \dots && \text{(expanded polynomial form)} \end{aligned}$$

4.6.7 Distance Properties

The free distance between a pair of convolutional codewords is the Hamming distance between the pair of codewords. The minimum free distance, d_{free} , is the minimum Hamming distance between all pairs of complete convolutional codewords and is defined as,

$$\begin{aligned} d_{free} &= \min\{d(\mathbf{y}_1, \mathbf{y}_2) | \mathbf{y}_1 \neq \mathbf{y}_2\} \quad [\text{Wic95}] \\ &= \min\{w(\mathbf{y}) | \mathbf{y} \neq \mathbf{0}\} \quad [\text{Wic95}] \end{aligned}$$

where $d(\bullet, \bullet)$ is the Hamming distance between a pair of convolutional codewords and (\bullet) is the Hamming distance between a convolutional codeword and the all-zero codeword (the weight of the codeword). The minimum free distance corresponds to the ability of the convolutional code to estimate the best decoded bit sequence. As d_{free} increases, the performance of the convolutional code also increases. This characteristic is similar to the minimum distance for block codes. From the transfer function, the minimum free distance is identified as the lowest exponent of D . From the above transfer function for Figure 2.14, $d_{\text{free}} = 5$. Also, if N and J are set to 1, the coefficients of D_i 's represent the number of paths through the trellis with weight D_i . More information about the codeword is obtained from observing the exponents of N and J . For a codeword, the exponent of N indicates the number of 1s in the input sequence, and the exponent of J indicates the length of the path that merges with the all-zero path for the first time [Pro95].

4.6.8 Error Probabilities

There are two error probabilities associated with convolutional codes, namely first event and bit error probabilities. The first event error probability, P_e , is the probability that an error begins at a particular time. The bit error probability, P_b , is the average number of bit errors in the decoded sequence. Usually, these error probabilities are defined using the Chernoff Bounds and are derived in [Pro95], [Rhe89], [Wic95].

For hard-decision decoding, the first event error and bit error probabilities are defined as,

$$P_e < T(D, N, J) \Big|_{D=\sqrt{p(1-p)}, N=1, J=1}$$

and

$$P_b < \frac{dT(D, N, J)}{dN} \Big|_{D=\sqrt{p(1-p)}, N=1, J=1}$$

where

$$p = Q\left(\sqrt{\frac{2rE_b}{N_o}}\right)$$

and

$$Q(x) = \int_x^{\infty} \frac{1}{\sqrt{2\pi}} e^{-u^2/2} du$$

For soft-decision decoding, the first event error and bit error probabilities are defined as,

$$P_e < T(D, N, J) \Big|_{D=2^{-25/N}, N=1, J=1}$$

and

$$P_b < \frac{dT(D, N, J)}{dN} \Big|_{D=2^{-25/N}, N=1, J=1}$$

Two other factors also determine the performance of the Viterbi decoder. They are commonly referred to as the decoding depth and the degree of quantization of the received signal.

4.6.9 Decoding Depth

The decoding depth is a window in time that makes a decision on the bits at the beginning of the window and accepts bits at the end of the window for metric computations. This scheme gives up the optimum ML decoding at the expense of using less memory and smaller decoding delay. It has been experimentally found that if the decoding depth is 5 times greater than the constraint length K then the error introduced by the decoding depth is negligible [Pro95].

4.6.10 Degree of Quantization

For soft-decision Viterbi decoding, the degree of the quantization on the received signal can affect the decoder performance. The performance of the Viterbi decoder improves with higher bit quantization. It has been found that an eight-level quantizer degrades the performance only slightly with respect to the infinite bit quantized case [Wic95].

4.6.11 Decoding Complexity for Convolutional Codes

For a general convolutional code, the input information sequence contains $k \cdot L$ bits where k is the number of parallel information bits at one time interval and L is the number of time intervals. This results in $L+m$ stages in the trellis diagram. There are exactly $2^{k \cdot L}$ distinct paths in the trellis diagram, and as a result, an exhaustive search for the ML sequence would have a computational complexity on the order of $O[2^{k \cdot L}]$. The Viterbi algorithm reduces this complexity by performing the ML search one stage at a time in the trellis. At each node (state) of the trellis, there are 2^k calculations. The number of nodes

per stage in the trellis is 2^m . Therefore, the complexity of the Viterbi algorithm is on the order of $O[(2^k)(2^m)(L+m)]$. This significantly reduces the number of calculations required to implement the ML decoding because the number of time intervals L is now a linear factor and not an exponent factor in the complexity. However, there will be an exponential increase in complexity if either k or m increases.

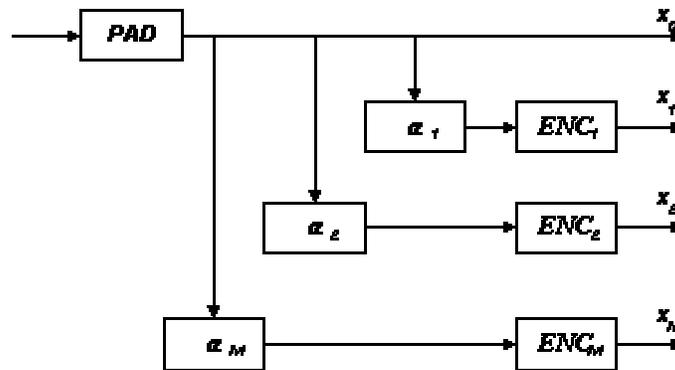
4.7 Turbo Codes

Turbo codes, first presented to the coding community in 1993 [23], represents the most important breakthrough in coding since Ungerboeck introduced the trellis codes in 1982. The invention of turbo codes involved reviving some of the basic concepts and algorithms and combine them with some new clever ideas. Because the principles surrounding turbo codes are both uncommon and novel, it has been difficult for the initiate to enter into the study of these codes.

Now here I provide a brief introduction of turbo codes and a brief description of encoder. Here my basic objective is to provide a simple introduction of turbo codes and turbo encoder/decoder for those who already have some knowledge in the field of algebraic and trellis codes.

4.7.1 Encoding

Turbo Code is the parallel concatenation of two or more systematic codes. A generalized turbo encoder is known in the figure below:



In the figure, a data block u , which is k bits long enters the coders. The PAD block appends $n - k$ tail bits to the data block, which yields the sequence x_0 . This n bit sequence is then fed in parallel into M sets of interleavers α_i and encoders ENC_i . Each interleaver scrambles the x_0 sequence in a pseudo-random fashion and feeds its output into a constituent encoder. Each of the M constituent encoders presents a parity sequence x_i at its output. The information sequence x_0 together with the M parity sequences are concatenated to form the code word.

4.7.1.1 Encoder

The common practice of encoder is to use Recursive Systematic Convolutional (RSC) encoders. By using a convolution encoder, it is possible for the decoder to utilize a modified version of the Viterbi algorithm. Recursive encoder is used as nonrecursive encoder will result in output codes with poor distance properties. For the diagram of the turbo encoder, please refer to the diagram shown in our example below.

4.7.1.2 Puncture of the Output

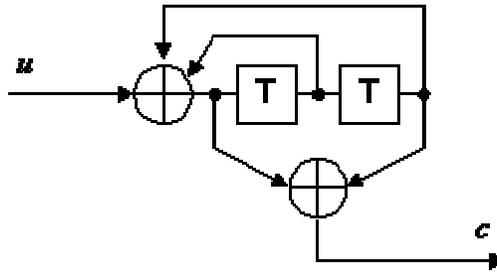
It is a common practice to puncture the output of the encoder in order to increase the code rate to 1/2. For a rate 1/2 punctured turbo code, the first output stream is the input stream itself (plus the necessary padding), while the second output stream is generated by multiplexing the M non-systematic output of the RSC encoders.

4.7.1.3 Example of Encoding of Turbo Code

Because turbo codes are linear block codes, the encoding operation can be viewed as the modulo-2 matrix multiplication of an information vector with a generator matrix. Here we demonstrate the encoding a sequence to Turbo code with the help of matrix representation:

ENCODING

Let the code generator matrix $g = [1 \ 1 \ 1 \ ; \ 1 \ 0 \ 1]$, the encoder will become:



The encoder has constraint length of 3 with memory of 2 only. The encoding algorithm is similar to that of convolutional encoding. In order to generate a block code using a parallel concatenation of convolutional encoders, it is desirable for encoders to start and end in the all-zero state. To ensure this will happen, extra bits are needed to "clear" the memory of encoder. Thus, the number of extra bits needed equals to the memory of encoder. In this example, 2 extra bits are needed.

For an input $u = [1\ 0\ 1]$,

Input u_k	State T_1	State T_2	Feedback X	Output c_k
1	0	0	1	1
0	1	0	1	1
1	1	1	1	0
0	1	1	0	1
1	0	1	0	1
	0	0		

Therefore, output code $c = [1\ 1\ 0\ 1\ 1]$.

Interleaving

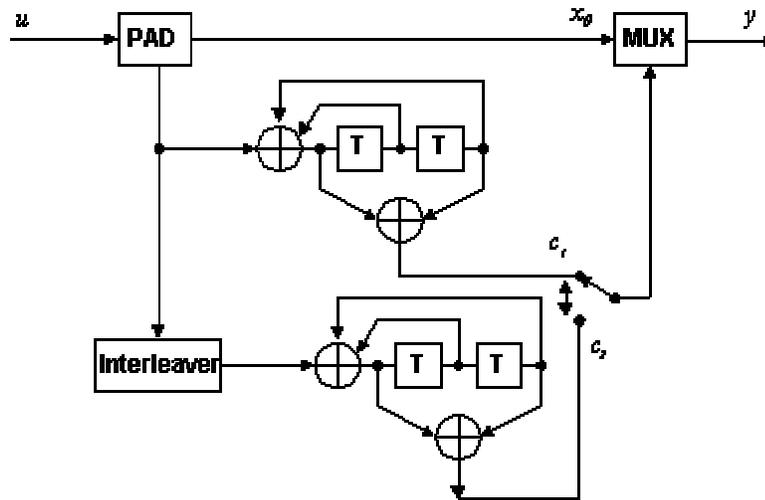
Consider a pseudo-random interleaver α with the mapping table as follow:

l	$\alpha(l)$
1	2
2	5
3	4
4	1
5	3

The interleave function $\alpha(l)$ means the l^{th} bit will take the $\alpha(l)^{\text{th}}$ bit of the original code. The above table can also be represent by a interleaver matrix $\alpha = [2 \ 5 \ 4 \ 1 \ 3]$. With input $u = [1 \ 0 \ 1 \ (0 \ 1)]$, interleaver output = $[0 \ 1 \ 0 \ 1 \ 1]$.

Multiplexing

Given the structure of constituent encoder and the interleaver, the turbo encoder become:



With input $u = [1 \ 0 \ 1 \ 0 \ 1]$,

- The first output stream x_0 is the input stream u itself with padding and equals to $[1 \ 0 \ 1 \ 0 \ 1]$.
- The encoder 1 output c_1 equals to $[1 \ 1 \ 0 \ 1 \ 1]$.
- The encoder 2 input equals to $[0 \ 1 \ 0 \ 1 \ 1]$.
- The encoder 2 output c_2 equals to $[0 \ 1 \ 1 \ 0 \ 0]$.

The turbo coder output is therefore the multiplexing of the above three codes.

Output Puncturing

There are two different way of multiplexing: output puncture or not.

If output puncturing is not implemented, the output code are simply multiplexed together by taking a bit from each stream alternatively. The resulting code becomes

x_0 : $[1 \ 0 \ 1 \ 0 \ 1]$

c_1 : $[1 \ 1 \ 0 \ 1 \ 1] \Rightarrow y$: $[1 \ 1 \ 0 \ 0 \ 1 \ 1 \ 1 \ 0 \ 1 \ 0 \ 1 \ 0 \ 1 \ 1 \ 0]$

c_2 : [0 1 1 0 0]

If output puncturing is implemented, the all the encoder output will multiplex into the second channel only, achieving code rate = 1/2. The code for the second channel will become the multiplexing of output of both encoders c_1 and c_2 in the example:

c_1 : [1 1 0 1 1]

c_2 : [0 1 1 0 0] $\Rightarrow x_1$: [1 1 0 0 1]

and the final turbo encoder output is the multiplexing of the systematic data x_0 and the multiplexed encoder stream x_1 :

x_0 : [1 0 1 0 1]

x_1 : [1 1 0 0 1] $\Rightarrow y$: [1 1 0 1 1 0 0 0 1 1]

4.7.3 Decoding

It is proposed that an iterative decoding scheme should be used. The decoding algorithm is similar to Viterbi algorithm in the sense that it produces soft outputs. While the Viterbi algorithm outputs either 0 or 1 for each estimated bit, the turbo code decoding algorithm outputs a continuous value of each bit estimate. While the goal of the Viterbi decoder is to minimize the code word error by finding a maximum likelihood estimate of transmitted code word, the soft output decoding attempts to minimize bit error by estimating the posterior probabilities of individual bits of the code word. We called the decoding algorithm Software Decision Viterbi Decoding.

The turbo decoder consists of M elementary decoders - one for each encoder in turbo encoding part. Each elementary decoder uses the Software Decision Viterbi Decoding to produce a software decision for each received bit. After an iteration of the decoding process, every elementary decoder shares its soft decision output with the other $M - 1$ elementary decoders.

In theory, as the number of these iterations approaches infinity, the estimate at the output of decoder will approach the maximum a posteriori (MAP) solution.

An Example of Decoding of Turbo Code

We make use of the previous example of encoding of Turbo code to illustrate the decoding of the Turbo Code. The encoded output is

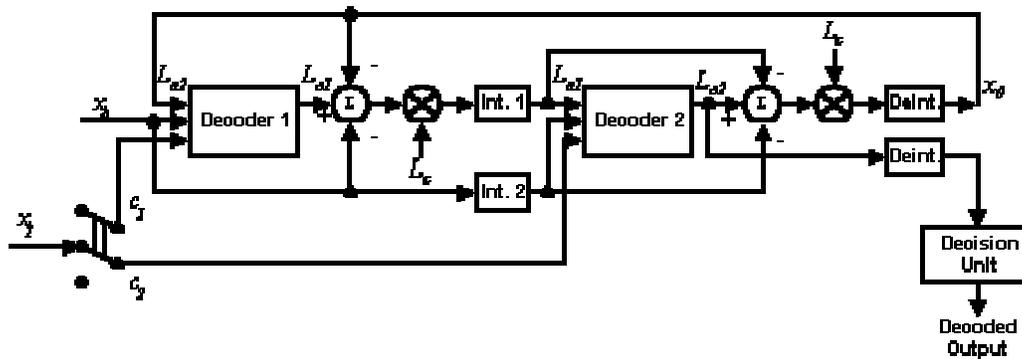
$$[1\ 1\ 0\ 0\ 1\ 1\ 1\ 0\ 1\ 0\ 1\ 0\ 1\ 1\ 0]$$

Suppose the channel is noisy, some bits are corrupted after transmission, the received bits becomes

$$[1\ 1\ 0\ \underline{1}\ 1\ 1\ 1\ 0\ 1\ 0\ \underline{0}\ 0\ 1\ 1\ \underline{1}] \text{ (bits underlined are corrupted)}$$

The Overall Schematic

The turbo-code decoder can be described in the following diagram:



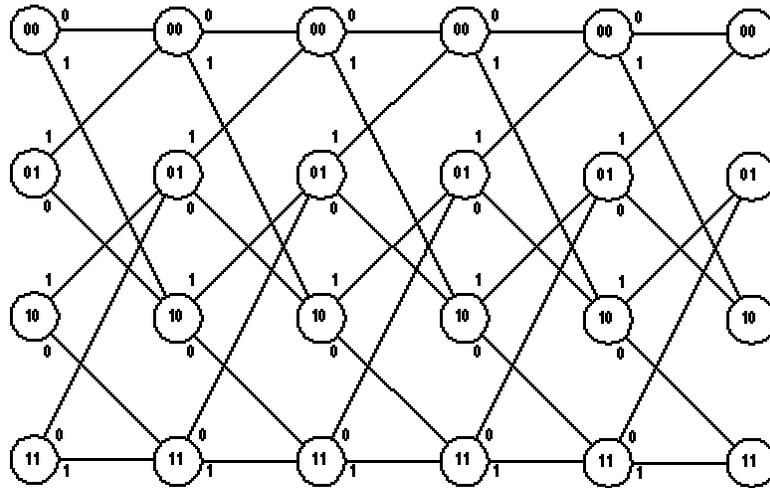
Assuming zero decoder delay in the turbo-decoder, the decoder 1 computes a soft-output from the systematic data (x_0), code information of encoder 1 (y_1) and a-priori information (La_2). From this output, the systematic data (x_0) and a-priori information (La_2) are subtracted. The result is multiplied by the scaling factor called channel reliability L_c to compensate the distortion. The result is uncorrelated with x_k and is denoted as Le_1 , for extrinsic data from decoder 1.

Decoder 2 takes as input the interleaved version of Le_1 (the a-priori information La_1), the code information of second encoder (y_2) and the interleaved version of systematic data ($a(x_k)$). Decoder 2 generates a soft output, from which the systematic data ($L_c a(x_0)$) and a-priori information (La_1) was subtracted. The result is multiplied by the scaling factor called channel reliability L_c to compensate the distortion. The extrinsic data from decoder 2 (Le_2) is interleaved to produce La_2 , which is fed back to decoder 1. And the iterative process continues.

Soft Output Decoding Algorithm (SOVA)

Step 1: Form the Trellis

Inside the decoder, Soft-Output Viterbi Algorithm (SOVA) is used to determine the result with maximum likelihood. The process SOVA is similar to that Viterbi algorithm. A trellis is formed first.



The trellis is to show how the codes are outputted by encoder. The bits in each node represents the states of the encoder. Each line represents a transition on receipt of codes from encoder. The encoder output (decoder input) for all combination of states and input of the encoders are summarized as follows.

State 1	State 2	Input	Output	Next State 1	Next State 2	Decoder Input
0	0	0	0	0	0	00
0	0	1	1	1	0	11
0	1	0	0	1	0	00
0	1	1	1	0	0	11
1	0	0	1	1	1	01
1	0	1	0	0	1	10
1	1	0	1	0	1	01
1	1	1	0	1	1	10

Step 2: Determine the Accumulated Maximum Likelihood for Each State

From the above table, each state has its own input bit pattern. When bit streams are inputted to decoder, they can be compared to the input $(x_0 x_1)$ to see whether they are matched. The result is represent by likelihood of input:

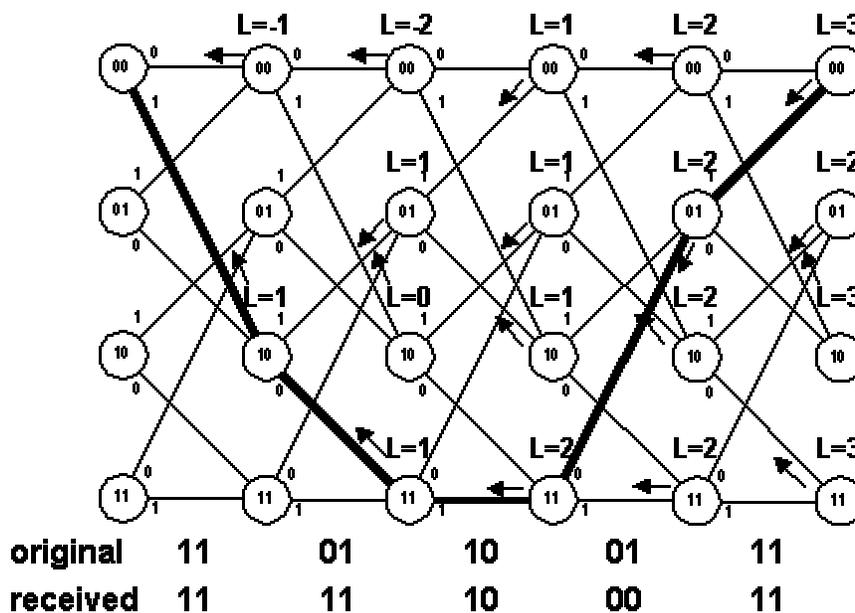
- $L_i = -1$ if no bits are matched
- 0 if 1 bit are matched
- 1 if all 2 bits are matched

The overall likelihood of a transition is sum of likelihood of input and a-prior likelihood information (L_p).

$$L = L_i + L_p$$

- $L_p = 0.5 \times$ a-prior information L_{an} if x_0 are 1
- $-0.5 \times$ a-prior information L_{an} if x_0 are 0

The algorithm is as follows: Start from state 00, the overall likelihood of each transition are evaluated. The overall likelihood of each node is obtained by the maximum accumulated likelihood. With this algorithm, the following will be obtained.



Step 3: Find the Surviving Path

The surviving path is thus derived by tracing back from last stage (stage 5) to the first one (stage 0) and is the results of hard-output Viterbi Algorithm [1 0 1 0 1]

Step 4: Determine the Soft Output

To find the soft-output, non-surviving paths are considered. We consider the non-surviving path the one that is product by making different decision in one of the stage in tracing back. For example, when tracing back from (last stage) stage 5, one of the non-surviving path is found by tracing back to state 00 instead of state 01 from stage 5 to stage 4. Following the arrows, bit 1 is also 1. We found that bit 1 will have the following results when decision is changed in different stages:

Stage 1 2 3 4 5

Bit 1 X X 0 1 1 Where X means cannot trace back to state 00 in stage 0

Delta - - 3 1 2

Here we define a function delta to describe the tendency to have non-surviving path. It is the difference in overall likelihood when a different decision in a particular the stage. The soft-output of bit 1 is evaluated by the following formula:

$bit_value \times \min(\text{delta which make bit 1 change to value other than } bit_value)$

$bit_value = 1$ for bit output = 1

-1 for bit output = 0

In bit one, the decision only changes when a state changes in stage 3. The minimum delta is 3. Thus the soft-output of the bit one is 3.

Repeat it for bit two (originally bit two = 0),

Stage 2 3 4 5

Bit 1 X 1 1 1 where X means cannot trace back to state 00 in stage 0

Delta - 3 1 2

From the above results, in bit two, the decision changes when a state changes in stage 3, 4 and 5. The minimum delta is 1. Thus the soft-output of the bit two is -1. Using this algorithm, the soft-output will become [3 -1 1 -1 2]

Feeding Data to Another Decoder

After the soft-output is evaluated by SOVA decoder, the data will be passed to the second decoder for further decoding. Before passing to data to the second decoder, two processes are performed to the decoder output:

the a-prior information (La_2) and the systematic data (x_0) are subtracted

$$\begin{array}{rcccccc}
 d_1 & 3 & -1 & 1 & -1 & 2 \\
 La_2 & 0 & 0 & 0 & 0 & 0 \\
 x_0 & 1 & 1 & 1 & -1 & 1 \\
 Le_1 & 2 & -2 & 0 & 0 & 1
 \end{array}$$

the result is multiplied by the scaling factor called channel reliability L_c . The reason of the factor is because the SOVA algorithm suffers a major distortion which is caused by over-optimistic soft outputs. The factor is used to compensate this distortion.

$$L_c = \text{mean}(Le_1) \times 2 / \text{var}(Le_1)$$

Iterative Decoding

The data from first decoder $L_c Le_1$, together with the systematic data $\alpha(x_k)$ and code information from of second encoder (y_2), are then fed into the second decoder for decoding, the decoding algorithm is the same as the first one.

After decoding, the output of second decoder is processed in the same way and fed back to the first decoder. The process continues. The number of iterations depends on the designer. Usually, the larger the iteration, the more accurate the data but the longer the time its takes for decoding.

Decision of Output

After iterations of decoding, the decoding results is the sign of the soft-output of the last decoder. Take the example, for the results of first decoder, the output become:

<i>decoder output</i>	3	-1	1	-1	2
<i>result</i>	1	0	1	0	1

which is the same as the input bit stream u . i.e. the error can be recovered.

4.7.3 Performance Analysis of Turbo Codes

Since its existence, Turbo Code becomes a hot topic of many researchers. Many researches have been done to study how to study the characteristics of this powerful code. One of the objectives of the project is to study how the Turbo Code can be applied to daily applications. This cannot be done without the knowledge of the characteristics of this coding scheme. Therefore we take time to have a look at the performance issue of turbo code.

In general, the following is the major factors of turbo code:

- Frame Size
- Encoder Memory Size
- Encoder Output Puncturing
- Number of decoder iterations
- Noise level

The above factors will affect the performance of the Turbo code in different ways.

4.7.3.1 Frame Size

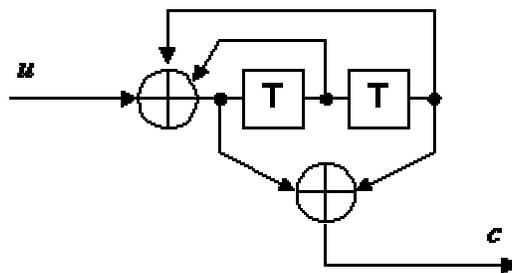
The larger the frame size, the bits can be interleaved with larger distance. Thus the correlation between adjacent bits will become smaller. This will give better performance on Turbo Code in terms of accuracy.

In the decoding of Turbo Code, the size of trellis formed is linearly proportional to the frame size. The complexity of the decoding algorithm is independent of the frame size. Thus decoding time increases linearly with the frame size. That means the decoding time for each bit is unchanged regardless of the change in frame size.

However, as Turbo Code is block code, one has to wait for the completion of decoding the whole block before getting the decoded output. Thus, increasing the frame size will make the whole decoding process longer, thus increasing the latency.

4.7.3.2 Encoder Memory Size

In Turbo Code encoding scheme, there are M Recursive Systematic Convolutional (RSC) encoders. Each encoder has memory to store the previous bits/state information. The memory size of an encoder is the number of bit/state can be stored in the encoder. In our example the encoder has a memory size of 2, which means two bits/states can be stored in the encoder at any time.



Like Convolution Code, for larger memory size, Turbo Code has better performance as the coding algorithm becomes more sophisticated.

However, in the decoding of Turbo Code, the number of state n is exponentially proportional to the memory size m .

$$n = 2^m$$

The size of trellis formed is exponentially proportional to the encoder memory size. The exponential increase in complexity of the decoding algorithm will cause the decoding time increases dramatically with the memory size. The latency will increase exponentially too.

Due to the great influence of large memory size of the encoders. It is recommended that the encoder memory size should be kept in small value (3 to 5).

4.7.3.3 Output Puncturing

If output puncturing is implemented, the code rate will be restricted to $1/2$. This is useful in circumstances which the bandwidth limitation is so great that additional redundancy of code to achieve a code rate of less than $1/2$ is undesirable.

However, as output is punctured, some information is loss. That means the performance of Turbo Code will decrease in general. Bit error rate (BER) will increase.

4.7.3.4 Decoding Iterations

One distinctive characteristic of Turbo Code is iterative decoding, in which the results of one decoder will passed to the another decoder for the next decoding iterations. The intuition of iteration is that an decoder only get part of the information of the decoding bits (the first decoder gets the systematic output and also the first encoder output, while the second decoder gets the information of the systematic output and also the second encoder output, etc.)

Iteration is useful in sharing the information from one decoding to another. In our example, the first decoder does not have the information of the second encoder output in the first iterations. After the first iterations, the output of the second decoder will feed back into the input of the first encoder. Thus first decoder have more information in the second iterations and the decoding performance should be improved.

From the above, the performance of the Turbo Code increases as the number of iterations increases. However, the time used will also increases linearly as the number of iterations. This increases in decoding time per bits will lead to increase in latency.

Therefore, designers have to justify the number iterations to accommodate the performance/time ratio.

4.7.3.5 Noise Levels

The most direct factor to affect the performance of Turbo Code is noise level. Actually, noise is the root cause of implementing of coding scheme.

Noise level can be represented by signal energy per bit to noise power spectral density (E_b/N_o). This is used to measure the power efficiency (or energy efficiency) - how favorably this tradeoff between fidelity and signal power is made of a coding scheme.

The larger the E_b/N_o , the smaller the noise level. With more favorable environment, the BER of the Turbo Code will decrease, and vice versa.

Chapter 5

COFDM

Coded Orthogonal Frequency Division Multiplexing (COFDM) is a form of modulation which is particularly well-suited to the needs of terrestrial broadcasting channels. COFDM can cope with high levels of multipath propagation, with a wide spread of delays between the received signals. This leads to the concept of single-frequency networks in which many transmitters send the same signals on the same frequency, generating “artificial multipath”. COFDM also cops well with co-channel narrowband interference, as may be caused by the carriers of existing analogue services.

COFDM has therefore been chosen for two recent new standards for broadcasting, DAB and DVB-T, both of which have been optimized for their respective applications and have options to suit particular needs.

The special performance of COFDM in respect of multipath and interference is only achieved by a careful choice of parameters and with attention to detail in the way in which the forward error-correction coding is applied.

5.1 Why we need Error Coding

In fact, we would expect to use forward error-coding in almost any practical digital communication system, in order to be able to deliver an acceptable bit error rate (BER) at a reasonably low signal to noise ratio. At a high SNR it might not be necessary, and this is also true for uncoded OFDM, but only when the channel is relatively flat. Uncoded OFDM does not perform very well in a selective channel. Its performance could be evaluated for any selective channel for any modulation scheme, by:

- Noting the SNR for each carrier
- Deducing the corresponding BER for each carriers data
- Obtaining the BER for the whole data, by averaging the BERs of all the carriers used.

Very simple examples will show the point. Clearly, if there is a zero dB echo which is delayed such that every m^{th} carrier is completely extinguished, then the symbol error rate will be of the order 1 in m , even at infinite SNR. (Here symbol denotes the group of bits carried by one carrier within one OFDM symbol). An echo delay of say $T_u/4$, the maximum for which a loss of orthogonality is avoided when the guard interval fraction is $1/4$, would thus cause the SER to be 1 in 4. Similarly, if there is one carrier amongst N carriers in all, which is badly affected by interference, then the SER will be of the order of 1 in N , even with infinite SNR.

This tells us two things:

- *Uncoded OFDM is not satisfactorily for use in such extremely selective channels*
- *For any reasonable number of carriers, CW interference that is affecting one carrier is less of a problem than a 0 dB echo.*

However, just hard decision coding to this uncoded system is not enough, either, it would take a remarkably powerful hard-decision code to cope with an SER of 1 in 4! The

solution is to use convolutional coding in conjunction with soft-decision decoding, properly integrated with the OFDM system.

5.2 Soft Decision and Channel State Information

First let us review, just for simplicity, the modulation and detection of a single carrier, one bit is transmitted per symbol, with say, a “0” being sent by a modulation signal of $-1V$ and a 1 is transmitted by a $+1V$. At a receiver, assuming that the gain is correct, we should expect to demodulate a signal always in the vicinity of either $+1$ or $-1V$, depending on whether a 0 or 1 was sent. Any departure from the exact values of $+1$ or $-1V$, would have been caused by the inevitable noise added during transmission.

A hard decision receiver would operate according to the rule that negative signals should be decoded as “0” and positive ones as “1” with 0 being the decision boundary. So if the noise amplitude remains within the boundaries this receiver will not do any mistake. But noise may have occasionally have a large amplitude, although with lower probability than with smaller values. Common sense suggests that when a large amplitude signal is received we can be more confident in the hard decision, than if the amplitude is small.

This view of a degree of confidence is exploited in soft-decision viterbi decoders. These decoders maintains a history of many possible transmitted sequences, building up a view of their relative likelihood's and finally selecting a value of 0 or 1 for each bit, according to which has the maximum likelihood. For convenience, a viterbi decoder adds logarithmic likelihood's to accumulate the likelihood of each sequence.

With other rectangular constellation modulation schemes, such as 16QAM or 64 QAM each axis carries more than one bit, usually with Gray coding. At the receiver, a soft decision can be made separately for each received bit. The metric functions are now more complicated than for QPSK, being different for each bit, but the principle, the decoder exploits its knowledge of the expected reliability of each bit, still remains.

Matrices of COFDM are slightly more complicated. We start from the understanding that the soft decision information is a measure of the confidence to be placed in the accompanying hard decision.

When data are modulated onto a single carrier in a time invariant system, then a priori all data symbols suffer from the same noise power on average; the soft decision information simply needs to take note of the random symbol-by-symbol variations that this noise causes.

When data are modulated onto the multiple OFDM carriers, the metric become slightly more complicated as the various carriers will have different SNR. For example, a carrier which falls into a notch in the frequency response will comprise mostly noise, one in a peak will suffer much less. Thus, in addition to the symbol-by-symbol variations, there is another factor to take account of in the soft decisions: data conveyed by carriers having a high SNR and a priori more reliable than those conveyed by carriers having low SNR. This extra a priori information is usually known as channel state information (CSI).

The inclusion of CSI in the generation of soft decision is the key to the unique performance of COFDM in the presence of frequency selective fading and interference.

We now return to the simple example in which there is a 0dB echo, of such a delay as to cause a complete null on one carrier in every 4. Figure 5.1 illustrates the effect of this selective channel: 1 carrier in every 4 is nulled out while every another carrier in every 4 is actually boosted, and the remaining 2 are unaffected.

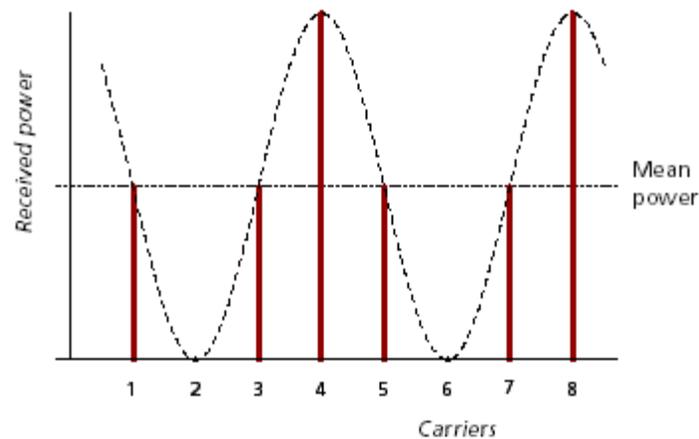


Figure 5.1: the effect of channel with a single 0dB echo of long delay, such that exactly 1 carrier out of 4 is nulled out.

Note that received power is shown, to which the SNRs of the carriers will be proportional if the receiver noise is itself flat, as in usual. The mean power marked on the diagram is

the mean of all carriers. It is equal to the total received power (via both paths), shared equally between all carriers.

Although a few OFDM carriers are illustrated, the pattern repeats cyclically for all of them. The dotted curve represents the power frequency response of the channel formed by the two paths.

In COFDM, the viterbi matrices for each bit should be weighted according to the SNR of the carrier by which it traveled. Clearly, the bits from the nulled carriers are effectively flagged as having “no confidence”. This is essentially the same thing as erasure, the viterbi decoder in effect just records that it has no information about these bits.

There is another well-known case of regularly-occurring erasures, namely punctured codes. Typically, convolutional codes intrinsically have code rates expressed as simple fractions such as $\frac{1}{2}$. When a code having a higher rate is needed, then one of these low rate “mother” codes is punctured, that is to say certain of the code bits are just not transmitted, according to a regular pattern known to the receiver. At the receiver dummy bits are again reinserted to replace the omitted ones, but are marked as erasures, bits having zero confidence, so the viterbi decoder treats them accordingly. Punctured codes are obviously less powerful than the mother code, but there is an acceptable steady trade off between performance and code rate, as the degree of puncturing is increased.

Now return to our simple COFDM example, in which one out of 4 is nulled out by the channel, but the corresponding bits are effectively flagged as erasures, thanks to the application of CSI. 2 out of 3 of the remaining carriers are received at the same SNR as that of the overall channel, while 1 carrier is actually boosted, having an improved SNR. Suppose that rate $\frac{1}{2}$ coding is used for the COFDM signal. It follows that the SNR performance of COFDM with this selective channel should be very slightly better (because 1 carrier is boosted) than that for a single carrier system using the corresponding rate $\frac{2}{3}$ code in a flat channel. In other words, the effect of this very selective channel on COFDM can be directly estimated from knowledge of the behavior of puncturing the same code when used in an SC system through a flat channel.

This explains how the penalty in the required CNR, for a COFDM system subject to 0 dB echoes, may be quite small, provided a relatively powerful convolutional code is used together with the application of CSI.

5.3 Interleaving

So far we have considered a very special example in order to make it easy to explain, by invoking the close analogy with the use of code puncturing. But what of the other delay values?

If the relative delay of the echo is rather shorter than we have just considered, then the notches in the channels frequency response is broader, affecting many adjacent carriers. This means that the coded data we transmit should not simply be assigned to the OFDM carriers in the sequential order, since at the receiver this would cause the viterbi soft-decision decoder to be fed with clusters of unreliable bits. This is known to cause a serious loss of performance, which we avoid by interleaving the coded data before assigning them to OFDM carriers at the modulator. A corresponding de-interleaver is used at the receiver before decoding. In this way, the cluster of errors occurring when adjacent carriers fail simultaneously is broken up, enabling the viterbi decoder to perform better. As just described, the process could be called frequency interleaving. This is all that is needed if the channel only varies slowly with time, and that is why it is used in DVB-T. In mobile operation (a key application of DAB), we may expect the various paths to be subjected to different and significant Doppler shifts, making the frequency response vary with time.

5.4 More Coding

DAB conveys audio data which, despite compressed in source coding, is relatively robust to the effects of transmission errors. The BER remaining after correction by the viterbi decoder is adequate. On the other hand, the compressed data of DVB-T is more susceptible to errors so that the residual BER at the output of the viterbi decoder is too high.

Thus DVB-T includes a second stage of error coding, called the outer coding, since in an overall block diagram it sandwiches the “inner” convolutional coding. Data to be transmitted are first coded with a Reed-Solomon code, interleaved with an additional “outer” interleaver, then passed to the “inner” convolutional coder. At the receiver, the

viterbi decoder is followed by an “outer” interleaver and the “outer” R-S decoder. The R-S decoder uses hard decisions, but is able to reduce the BER substantially, despite very modest extra redundancy having been added to the transmitter.

5.5 Conclusion

Hence I conclude this chapter simply, that, COFDM as used in DAB and DVB-T is very well matched to the terrestrial channel, being able to cope with severe multipath and the presence of co-channel narrowband interference.

COFDM is also adaptable to various uses by making an appropriate choice of parameters. Both DAB and DVB-T have a range of options to facilitate this.

COFDM only works because all the key elements are correctly integrated. These elements include many orthogonal carriers, added guard intervals, interleaving, soft decision viterbi decoding and the use of channel state information.

Chapter 6

Literature Survey

6.1 Survey

The first OFDM scheme was proposed by Chang [4] in 1966 for dispersive fading channels, which has also undergone a dramatic evolution due to the efforts of Weinstein, et. al. Recently OFDM was selected as the high performance local area network transmission technique. A method to reduce the ISI is to increase the number of sub carriers by reducing the bandwidth of each subchannel while keeping the total bandwidth constant [30]. The ISI can instead be eliminated by adding a guard interval at the cost of power loss and bandwidth expansion [31]. These OFDM systems have been employed in military applications since the 1960's for example by Bello [5], Zimmerman [6], Powers and Zimmerman [7], Chang and Gibby [8] and others. The employment of discrete

Fourier transform (DFT) to replace the banks of sinusoidal generators and the demodulators was suggested by Weinstein and Ebert [9] in 1971, which significantly reduces the implementational complexity of OFDM modems. In 1980, Hirosaki [10], suggested an equalization algorithm in order to suppress both intersymbol and intersubcarrier interference caused by the channel impulse response or timing and frequency errors. Simplified model implementations were studied by Peled [11] in 1980. From Erlangen University, Kolb, Schubler, and Preuss conducted further research into the applications of OFDM. Cimini [12] and Kalet [13] published analytical and early seminal experimental results on the performance of OFDM modems in mobile communication channels.

Most recent advances in OFDM transmission were presented in the impressive state of art collection of works edited by Fazel and Fettweis [14]. OFDM transmission over mobile communications channels can alleviate the problem of multipath propagation. Recent research efforts have focused on solving a set of inherent difficulties regarding OFDM, namely peak-to-mean power ratio, time and frequency synchronization, and on mitigating the effects of the frequency selective fading channels.

Now one of the most important question regarding with OFDM is, why do we need error coding with OFDM? In fact, at low signal to noise ratios, in order to maintain an acceptable BER, we need some sort of error-correcting coding. At high SNR and when the channel is relatively flat, we don't need any coding, practically. More recently COFDM has been studied and implemented for digital television and terrestrial broadcasting by a number of researchers [25,26]. The successful demonstration of DAB gave researchers encouragement to further explore OFDM and COFDM for television broadcasting.

Initially trellis coded modulation TCM [27] combined with frequency and time interleaving is considered most effective means for frequency selective channel. Leonard J. Cimini investigate the performance of a Reed-Solomon coded OFDM system [28], for Advanced Cellular Internet Services. In [29], J. Kim and L. J. Cimini proposed that with Reed-Solomon codes there is a problem of random bits errors in symbols. So than they considered convolutional codes which are robust again random bit errors, but not so effective against bursty errors unless significant interleaving is applied. Then, they

combine the attributes of both RS and Convolutional codes to devise a scheme which is robust against both random and bursty errors. In [12], an OFDM system was introduced in mobile channel and its performance was investigated in flat fading channel. In [32], performance of RS Coded 16 QAM OFDM system was investigated, when the number of sub carriers varied for various channels under the condition of fixed total bandwidth and information rate.

In [23], a powerful coding technique, Turbo Coding, has been shown to perform near the Shannon capacity limit in an additive white Gaussian noise (AWGN) channel. In [33], Performance of turbo coded OFDM were analyzed under slow fading conditions and studied the effects of changing word size. In [35], Alister Burr and George White also analyzed the performance of coded OFDM using turbo codes, in digital broadcasting system, they considered particularly the effect of excess multipath delay, greater than the OFDM guard period. In [36], comparison of convolutional and turbo codes for OFDM with antenna diversity was analyzed.

When the channel state information (CSI) is not available at the transmitter, space time coding is an optimal coding strategy. In [34], it was demonstrated that the serial concatenation of an inner space time trellis code and an outer reed Solomon code employed with OFDM transmit diversity in terms of both bandwidth and power efficiency. More extensive code search provided improved versions of STTrC's [37,38] but no significant breakthrough has been achieved.

Chapter 7

Comparison of Convolutional and Turbo codes For OFDM in Wireless Applications

7.1 Introduction

Universal Mobile Telecommunications System (UMTS) is a third generation mobile radio communications system that supports wide range of services with different bit rates and quality of service on a single mobile terminal and wideband DS-SS-SS has been selected for the UMTS radio interface standard.

For UMTS, data transmission services require BER of 10^{-6} or lower with very high bit rates up to 2 Mbit/s is allowed. To satisfy both BER and high data rate transmission, very powerful forward error correction codes should be used.

From up to now, convolutional codes have been widely used in mobile radio communication systems due to their relatively good performance with reasonably simple Viterbi decoding algorithm. However, the complexity of Viterbi algorithm grows exponentially for increasing the number of memory.

In 1993, parallel concatenated convolutional codes (Turbo codes) introduced in [23] have been shown to perform near Shannon limit on AWGN channel with relatively simple iterative decoding technique. As a powerful coding technique, since their introduction, Turbo codes have been proposed for any communication system where a significant lower saving is required or the operating signal to noise ratio is very low such as deep-space and satellite communications, microwave links, paging as well as for interference limited applications such as third generation cellular and personal communication services.

Serial concatenated convolutional code is a serial concatenation of two convolutional codes cascaded through an interleaver that permuting the outer codewords symbols. The decoding of SCCCs can be done iteratively by decoding the convolutional encoders separately and sharing the bit reliability information in an iterative procedure. SCCCs differ from PCCCs only in the way constituent convolutional codes are combined, but have been shown to yield performance comparable and in some case superior to Turbo codes. At high enough SNR, SCCCs achieves much higher coding gain than that of Turbo codes, so that, using SCCCs can avoid the “floor effect” behavior, which is typical of Turbo codes [39,40]. This suggests the use SCCCs for those applications where a very low BER is required as data communication and future wireless applications.

7.2 Performance of Turbo Codes and Serially Connected Convolutional Codes In Wideband CDMA

In [41], performance of Turbo codes and serial concatenated convolutional codes (SCCCs) with the different number of memory in component convolutional codes ($m=2$; $m=4$) over frequency selective Rayleigh multipath fading channel in a wideband DS-SS-CDMA is investigated through simulation. System parameters are closely matched to UMTS requirements. Only low mobile speed of 5 km/h is considered. The comparison of Performance between Turbo codes and SCCCs is also made based on the same number of states or the equal decoding complexity.

7.2.1 Brief Overview of Turbo and Serially connected Convolutional Codes

Parallel concatenation of at least two component convolutional codes with interleaver between them forms a Turbo codes. A Turbo encoder with two component convolutional codes is shown in Fig. 7.1. Note that, the component convolutional codes must be Recursive Systematic Convolutional Codes.

The encoders ENC 1 and ENC 2 encode the same input information bit u_k but in different order due to the interleaver in front of ENC2. Appropriate puncturing of parity bits from two encoders can create a Turbo code of desired rate.

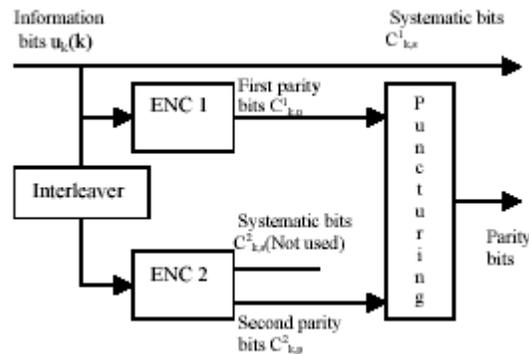


Figure 7.1: Block diagram of Turbo code encoder

The block diagram of an SCCC encoder is shown in Fig.7.2.



Figure 7.2: Block diagram of SCCC encoder

Firstly, information bits are encoded by an outer convolutional encoder. The output codeword of it is then permuted by interleaver before being encoded by an inner convolutional encoder.

Performance of Turbo codes and SCCCs depends so much on the selection of component codes as well as interleaver.

7.2.2 System Model

Block diagram of low pass equivalent representation of the overall simulation model is presented in Fig.7.3.

The random data generator generates digital information bits frame-by-frame. Since the information bit rate is 64 kbit/s, the frame size is set to 640 bits, in order to maintain the 10 ms information frame size recommended for UMTS (Universal Mobile Telecommunications System) [7], [8]. The information frame is then encoded by the channel encoder (Turbo code encoder or SCCC encoder) of overall rate $\frac{1}{2}$. Fixed pseudorandom interleaver is used in the channel encoder. The encoded data is interleaved by using an external interleaver for time diversity to reduce bursts of error. Assuming the available bandwidth of about 4.4 MHz, spreading of coded 128 kbit/s data stream is done using length-31 Gold code sequence. The resulting signal at chip rate of 3.968 Mchip/s is modulated using BPSK modulation. Band limiting transmit and receiver filters are omitted in the simulation work, assuming ideal timing. Hence sampling rate is at the chip rate. Synchronous downlink transmission is assumed. The transmitted signal is then passes through a multipath frequency selective Rayleigh fading channel. Channel is modeled as six-tap delay line with tap spacing equal to chip duration. Each path generates correlated Rayleigh fading channel coefficients by filtering complex Gaussian samples through the classical Doppler filter that introduces channel correlation. Exponential power delay profile is assumed among the multipaths, with rms delay spread of 0.4 μ s assumed for a typical urban environment. Additive White Gaussian Noise will be added to the transmitted signal at the receiver front end.

At the receiver, a three-finger Rake receiver is used for demodulation and despreading of the received signal. Coherent detection assuming ideal channel estimation is used with maximal ratio combining to extract energy from the three strongest paths. Soft channel outputs are calculated and passed to the channel decoder after deinterleaving. Decoding of Turbo code and SCCC using Log-MAP iterative decoding algorithm. Lastly, the decoded output data stream will be compared with the originally generated bit stream to compute the BERs correspondingly to various SNR by using Monte Carlo method.

The system configuration and parameters used in the simulation are summarized in Table 7.1.

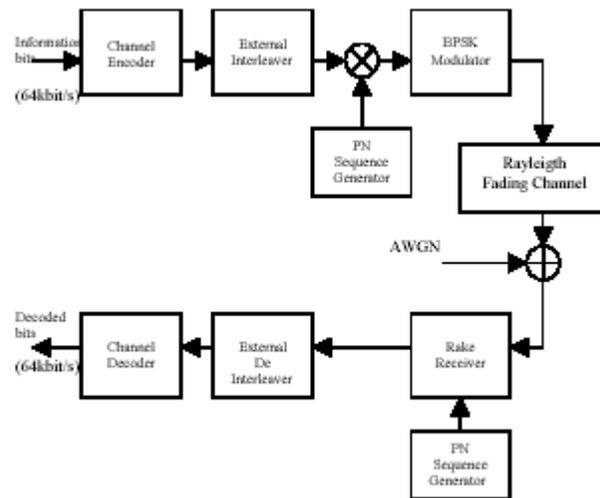


Figure 7.3: Block diagram of system model

And the table 7.1 is,

		Turbo Code	SCCC
Source data rate		64 kbit/s	
Channel Encoder i	Overall rate	1/2	
	Component Codes	Two identical rate 2/3 RSC codes obtained by puncturing (matrix [10]) of rate 1/2 , m=2 (4 states) RSC codes with generator $(1,5/7)_{\text{total}}$	Outer code: Rate 1/2, non-recursive convolutional code, m=2 (4 states), $G(D) = (7, 5)_{\text{total}}$ Inner code*: Rate 1/1, Punctured recursive Convolutional code, 2 states
Channel Encoder II	Overall rate	1/2	
	Component Codes	Two identical rate 2/3 RSC codes obtained by puncturing (matrix [10]) of rate 1/2 , m=4 (16 states) RSC codes with generator $(1,21/37)_{\text{total}}$	Outer code: Rate 1/2, non-recursive convolutional code, m=4 (16 states), $G(D) = (37, 21)_{\text{total}}$ Inner code*: Rate 1/1, Punctured recursive Convolutional code, 2 states
Input information block		640 bits	
Internal interleaver		Fixed Pseudorandom Interleaver ($N=640$)	Fixed Pseudorandom Interleaver ($N=1280$)
External Interleaver		Block Interleaver 1280x12 (120 ms)	
Chip rate		3.968 Mchip/s	
PN sequence		Gold code of length 31	
Modulation		BPSK	
Sampling rate		Chip rate	
Channel model		6 taps, exponential power delay profile, 0.4 μ s rms delay spread.	
Carrier frequency		2 GHz	
Mobile speed		5 km/h (modeled by Doppler filters)	
Receiver		3 finger Rake Receiver, maximal ratio combining	
Iterative decoding		Log-MAP Algorithm	
BER simulation		Monte Carlo method	

* The inner code is obtained by not sending the systematic bits of the rate 1/2 recursive convolution code with $G(D) = [1, 1/1+D]$.

Table 7.1: Summary of Simulation Configuration

7.2.3 Simulation Results

Now, in [41], by using above system, they were made lot of simulations, to demonstrate the performance of turbo codes and serially connected convolutional codes, and just to make our discussion complete, I uses their results directly.

There simulations only considers the performance at low mobile speed of 5 km/h with 120 ms (1280x12) external block interleaver. Since the coherent time of channel is long (19.3ms) and the interleaving depth is not large enough, burst errors due to deep fades can cause a lot of errors, which is unpredictable. So, the only way to get confident results is to simulate with large enough number of input blocks, especially in high range of SNRs and the time spent for simulation is too long.

Figures 7.4 and 7.5 show the performance of Turbo codes over Rayleigh multipath fading channel with the number of memory in component convolutional codes m=2 and m=4

respectively. We can see that, at the second iteration, the error performance can be improved with about 0.5 dB coding gain. The BER cannot be improved much after further iterations. The floor effect is not observed; hence lower BER at high SNR is expected.

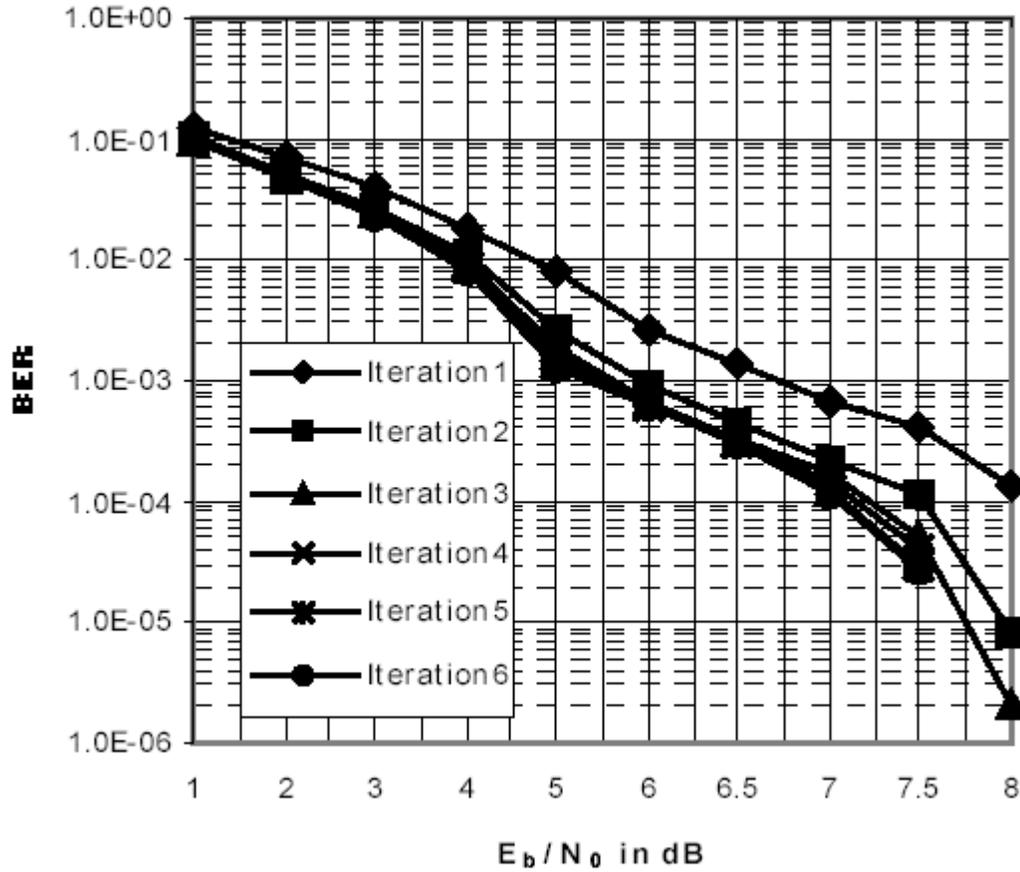


Figure 7.4: Performance of Turbo code ($m=2$) over Rayleigh multipath fading channel

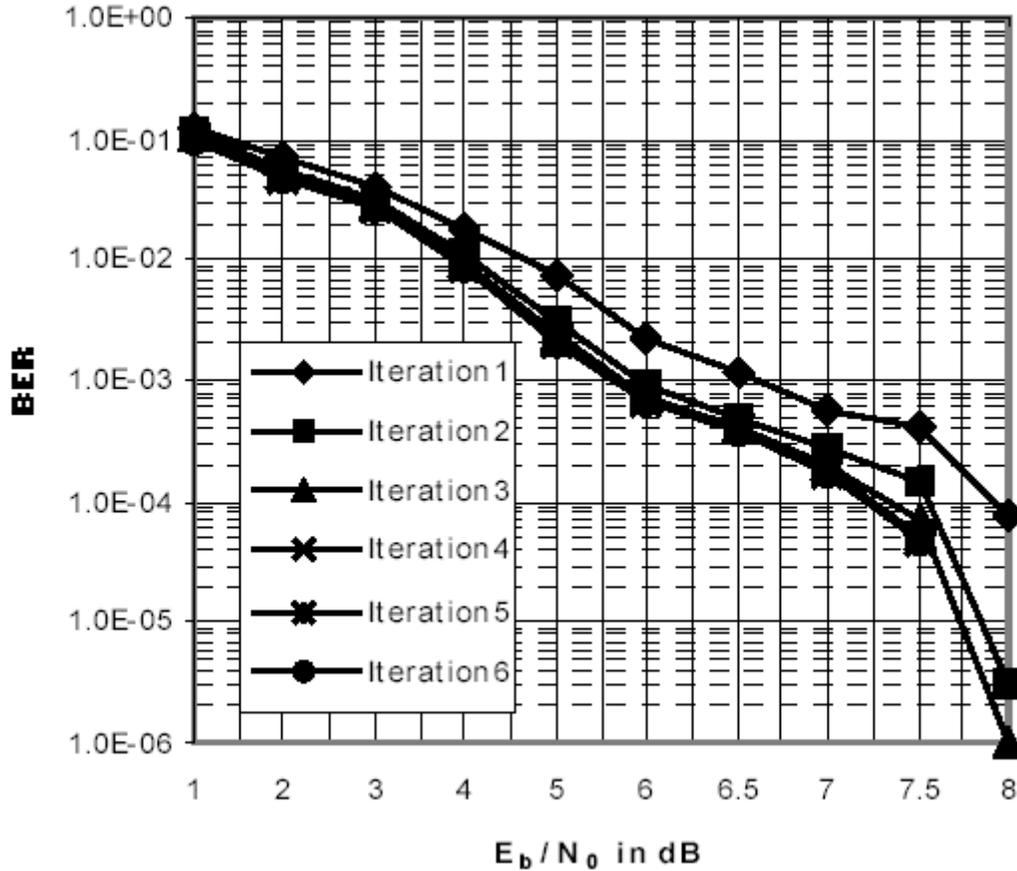


Figure 7.5: Performance of Turbo code ($m=4$) over Rayleigh multipath fading channel

The comparison between Turbo codes having $m=2$ and $m=4$ at 3rd iteration is illustrated in Fig.6. It can be observed that the performance of $m=4$ Turbo code does not improve for the case of $m=2$ in low and medium SNR. However, for BER less than 10^{-5} , $m=4$ Turbo code is little better.

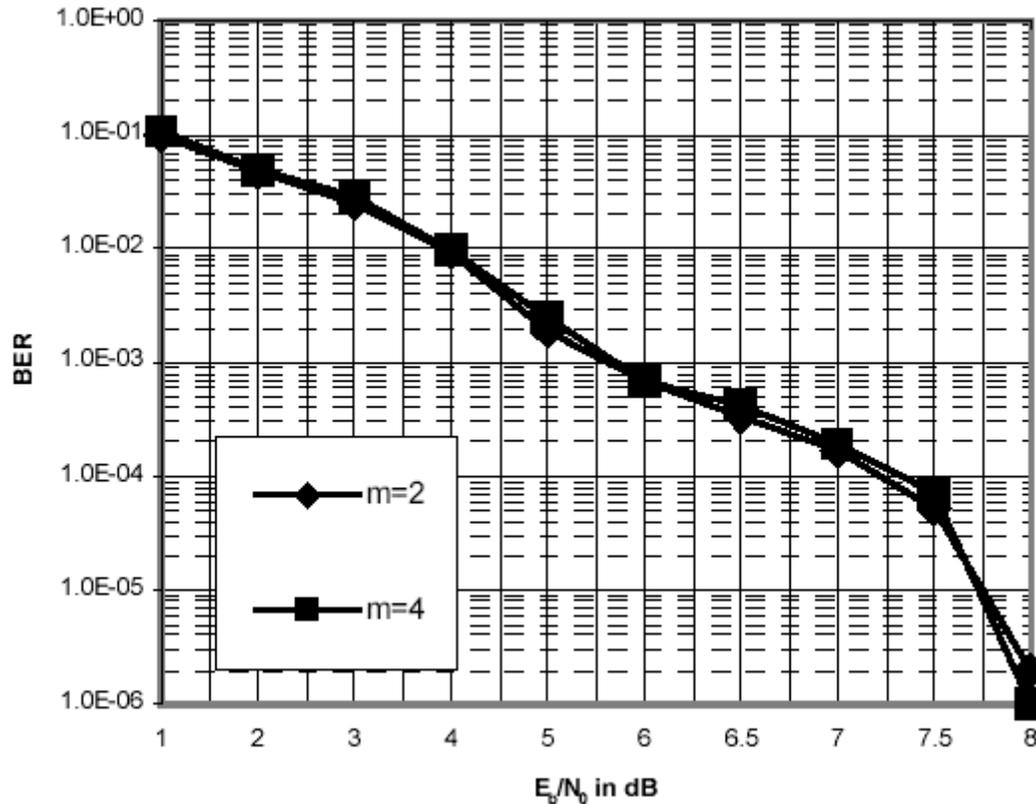


Figure 7.6: Comparison of performance of Turbo codes having $m=2$ and $m=4$ over Rayleigh multipath fading channel at 3rd iteration

Figures 7 and 8 show the performance of SCCCs over Rayleigh multipath fading channel with the number of memory $m=2$ and $m=4$ in the outer convolutional codes respectively. It is also can be seen that a significant coding gain is obtained after the second iteration and very little gain can be observed after the third iteration, but the BER curves seem to diverge at low BERs.

Figure 7.9 shows the comparison between $m=2$ SCCC and $m=4$ SCCC. In the range of whole SNRs corresponding BER above 10^{-6} , the performance of case $m=4$ is worse than $m=2$.

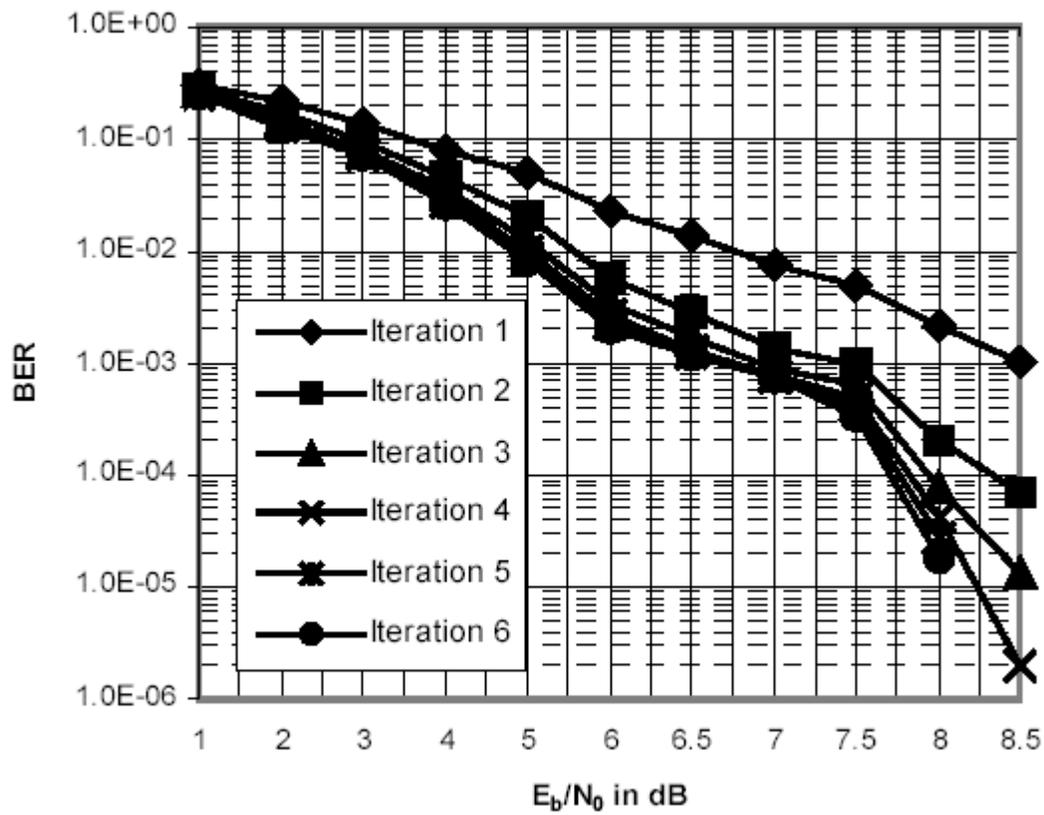


Figure 7.7: Performance of SCCC having the number of memory $m=2$ in the outer convolutional code over Rayleigh multipath fading channel

Finally, he showed the comparison of the two proposed coding schemes for different values of m . so here, I use,

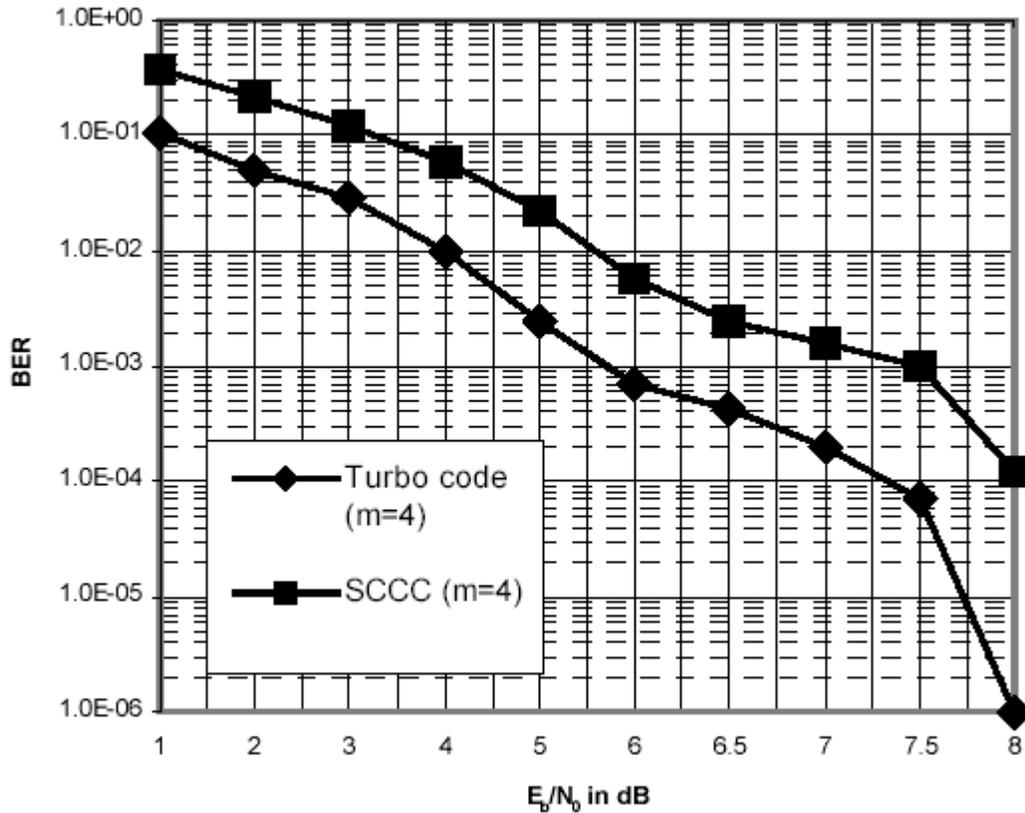


Figure 7.8: Comparison of performance of Turbo code and SCCC for $m=4$ over Rayleigh multipath fading channel at 3rd iteration

7.2.4 Conclusion

The performance of rate 1/2 Turbo codes and SCCC with $m=2$ and $m=4$ in a wideband DS-CDMA system, which matches to the third generation mobile radio communications systems-UMTS is investigated in this research study. Due to the time limitation only low mobile speed of 5 km/h is considered.

It was found that, both Turbo codes and SCCC can be expected to reach the BER level of 10^{-6} . Therefore, they can be suitable for data transmission in third generation wideband DS-CDMA systems. When the number of memory is increased, in low and medium SNRs, performance of $m=4$ Turbo code is little worse than $m=2$ Turbo code and in case of SCCC, approximately 0.5 dB coding gain of $m=2$ SCCC is obtained over $m=4$ SCCC. However, at high SNRs, a little better performance is achieved only for case of Turbo

codes, and the performance of $m=4$ SCCC is still worse than $m=2$ SCCC. Therefore, to some extent, using $m=2$ Turbo code is more effectively than using $m=4$ Turbo code since the better performance have to pay for more decoding complexity. For SCCCs, using $m=2$ SCCC is better.

The comparison between Turbo codes and SCCCs shows that, for both $m=2$ and $m=4$ the Turbo codes schemes outperform SCCCs schemes for BER above 10^{-6} .

7.3 Comparison Between Convolutionally and Turbo Coded OFDM

In [42], a nice comparison between convolutional-reed Solomon codes and Turbo Codes was presented. Firstly, results were presented for convolutional and Reed–Solomon codes and then they showed that even with short constraint lengths, convolutional codes have the potential to outperform Reed–Solomon codes, provided that sufficient precision is used in the soft decoder. Then, they evaluate the performance of Turbo codes under slow fading conditions and study the effects of varying codeword size. In [42], they conclude that increasing codeword size theoretically provides better interleaving between the two component codes. However, this advantage is less clear when the fading rate is significantly lower than the symbol rate, which is typical of the high-data-rate systems. Under such conditions, the advantage of using two component convolutional codes in Turbo codes is limited. So, it is preferred that a single convolutional code with a long constraint length may be a better choice.

In this paper, they studied and compared the performance of different coding strategies for a coded OFDM system with diversity as a means to provide high-speed wireless data services. The Key point of this paper is, that decoder design plays a critical role in system performance. By using convolutional codes with good precision in the soft decoding, excellent performance can be attained even with relatively short constraint lengths. By employing a long constraint length, up to, convolutional coding can reduce the link budget by more than 3 dB relative to what can be achieved by Reed–Solomon coding.

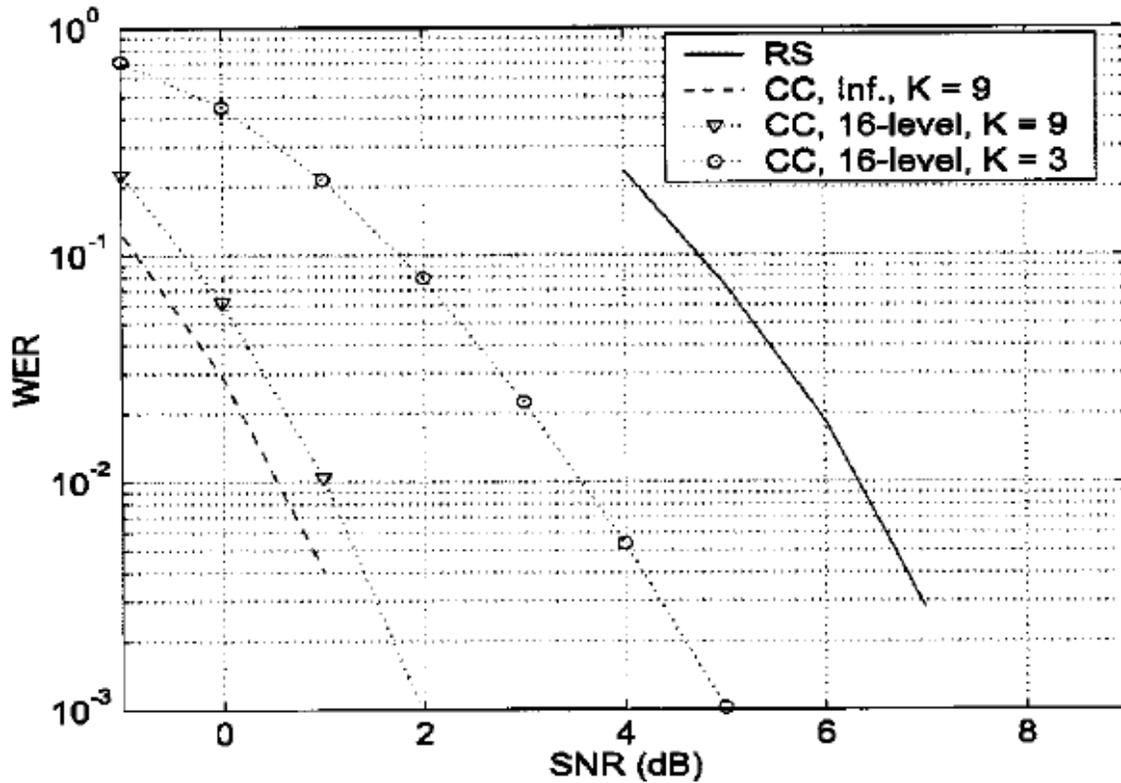


Figure 7.9: Performance of Reed Solomon codes and Convolutional codes with coherent demodulation

The performance of convolutional codes also compared favorably with that of Turbo codes with similar complexity. Turbo codes can potentially outperform convolutional codes at high SNR but their performance is severely limited at low SNR due to error propagation in the iterative decoding. However, if the fading is relatively slow compared to the symbol rate, the advantage of Turbo codes with two identical recursive systematic convolutional component codes is limited. A single convolutional code with a long constraint length may be a better choice.

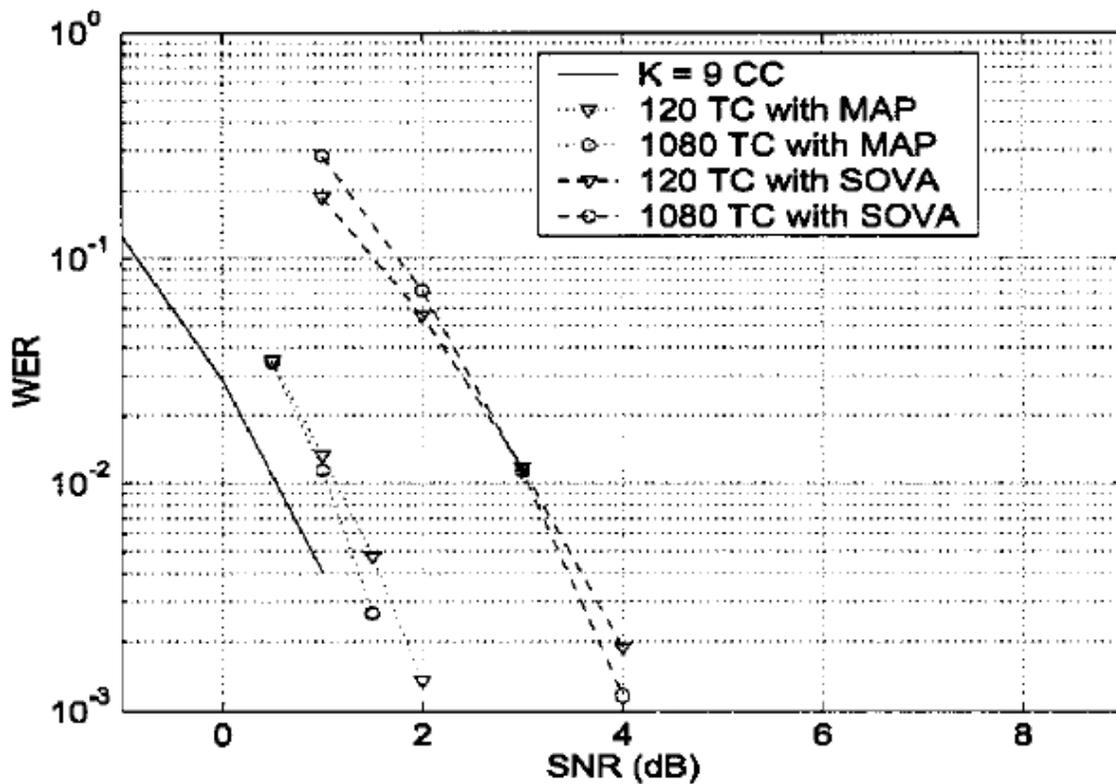


Figure 710: Performance of Turbo codes with coherent demodulation

7.4 Conclusion

The current status of the research is that COFDM appears to be a suitable technique as a modulation technique for high performance wireless telecommunications. As research suggests that an OFDM link has been confirmed to work by using computer simulations, and people also performed some practical tests on a low bandwidth baseband signal. So far only four main performance criteria have been tested, which are OFDM's tolerance to multipath delay spread, channel noise, peak power clipping and start time error. Several other important factors effecting the performance of OFDM have only been partly measured. These include the effect of frequency stability errors on OFDM and impulse noise effects.

One important major area which hasn't been investigated is the problems that may be encounter when OFDM is used in a multiuser environment. One possible problem which may be encountered is the receiver may require a very large dynamic range in order to

handle the large signal strength variation between users. Still there are some other areas of OFDM like synchronization which require research, because in past they are not investigated too much.

Most practical systems uses forward error correction coding to improve the system performance. Thus more work needs to be done on studying forward error correction schemes that would be suitable for telephony applications, and data transmission. So here in this term project I studied different coding schemes with OFDM. Normally UMTS data transmission services require BER or 10^{-6} or lower with very high data rates up to 2 Mbits/s. To satisfy both BER and high data rate transmission, very powerful forward error correction codes should be used. From up to now, convolutional codes have been widely used in mobile communication systems due to their good performance and simple decoding algorithm (viterbi algorithm). However, after the advent of turbo codes, they replace the convolutional codes due to their marvelous performance. So here I studied both codes with OFDM and make a comparison under which conditions which codes perform better with OFDM. We conclude here that turbo codes can potentially outperform convolutional codes at high SNR, but at lower SNR the performance of convolutional codes is better than turbo codes.

REFERENCES

- [1] C.G. Giinther, J. E. Padgett, and T. Hattori, "Overview of wireless communications," IEEE Communication Mag., Vol. 33, No. 1, pp. 28-41, 1995.
- [2] K. Ben Lataief, J. C-I Chang and R. D. Murch, "A high transmission method for wireless personal communication," Kluwer Academic Publisher, Vol. 3, No. 3, pp.299-317, 1996.
- [3] J. C-I Chaung, K. Ben Lataief, K. F. Cheung, C. C. Ling, R. D. Murch, C. T. Nguyen and R. Yu, "Wireless Personal communications in Hong Kong: A university perspective," IEEE Persoanl Commu. Mag., Vol. 4, No.2, pp. 30-43, Apr. 1997.
- [4] R. Chang, "Synthesis of band-limited orthogonal signals for multichannel data transmission," BSTJ, vol. 46, pp. 1775-1796, December 1966.
- [5] P. Bello, "Selective fading limitations of the KATHRYN modem and some system design considerations," IEEE transaction on Communication Technology, vol. COM-13, pp. 320-333, September 1965.
- [6] M. Zimmerman and A. Kirsch, "The AN/GSC-10/KATHRYN/ variable rate data modem for HF radio," IEEE transaction Communication Tech. Vol. CCM-15, pp. 197-205, April 1967.
- [7] R. Chang and Gibby, "A theoretical study of performance of an orthogonal multiplexing data transmission scheme," IEEE trans. Communication Tech. Vol. COM-16, pp. 529-540, August 1968.
- [9] S. Weinstein and P. Ebert, "Data transmission by frequency division multiplexing using the discrete Fourier transform," IEEE Trans. Commun. Tech., vol. COM-19, pp. 628-634, October 1971.
- [10] B. Hirosaki, "An analysis of automatic equalizers for orthogonally multiplexed QAM systems," IEEE Trans. On Commun. vol. COM-28, pp. 73-83, January 1980.
- [11] A. Peled and A. Ruiz, "Frequency domain data transmission using reduced computational complexity algorithms," in Proceedings of International Conference on Acoustics, speech and signal processing, ICASSP'80, vol. 3, pp. 964-967, IEEE, 9-11 April 1980.

- [12] L. Cimini, "Analysis and simulation of a digital mobile channel using OFDM", IEEE Trans. On Commun. vol. 33, pp. 665-675, July 1985.
- [13] I. Kelet, "The Multitone Channel" IEEE Trans. On Commun. Vol. 37, pp.119-124, February 1989.
- [14] K. Fazel and G. Fettweis, "Performance of an efficient parallel data transmission system," IEEE Trans. Commun. Technol., pp. 805-813, December 1967.
- [15] L.F. Chang, and J. C-I Chaung, "Diversity selection using coding in a portable radio communication channel with frequency selective fading" IEEE journal on selected areas In Commun., vol. 7, pp. 89-97, Jan. 1987.
- [16] K. Ben Letaief, K. Muhammad, and J. S. Sadowsky, "Fast simulation of DS/CDMA with and without coding in multipath fading channels," IEEE Journ. on selected areas in communication, Vol. JSAC-15, No. 4, pp. 626-639, May 1997.
- [17] B.R. Salzberg, "Performance of an efficient parallel data transmission system", IEEE Trans. Commun. Technol., vol. COM-15, pp.805-813, Dec. 1967.
- [18] W. Warner and C. L. Leung, "OFDM/FM frame synchronization for mobile radio data communication," IEEE Transactions on Vehicular Tech. Vol.42, pp. 302-313, August 1993.
- [19] A. R. S. Bahai and B. R. Saltzberg, "Multi-Carrier Digital Communications: Theory and Applications of OFDM. Kluwer Academic Publishers, Dordrecht, the Netherlands, 2000.
- [20] P. Moose, " A Technique for orthogonal frequency division multiplexing offset correction," IEEE Trans. Communication Magazine, pp. 100-109, February 1999.
- [21] F. Daffara and O. Adami, "A new frequency detector for orthogonal multicarrier transmission techniques," in proceedings of IEEE Vehicular Technology Conference, pp. 804-809, IEEE, July 15-28 1995.
- [22] M. Sandell and J. J. Van de Back, "Timing and Frequency synchronization in OFDM systems using the cyclic prefix," in proceedings of International symposium on synchronization, pp. 16-19, 14-15 Dec. 1995.
- [23] C. Berrou, A. Glavieux and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding, Turbo codes," Proc. 1993 Conf. Comm., pp. 1064-1070.
- [24] G. Ungerboeck, "Channel coding with multilevel/phase signals," IEEE Trans. Inf. Theory, pp. 55-67, Jan. 1982.

- [25] “The COFDM system intended for Digital Audio Broadcasting takes benefit from time, frequency and space diversity,” French Contribution to CCIR WP10, Oct. 1991.
- [26] J. F. Helard and B. Le. Flock, “Trellis-Coded Orthogonal Frequency Division Multiplexing for Digital Video Transmission”, Proc. Of Globecom 91, Dec. 1991.
- [27] G. Ungerboec, “Trellis coded modulation with redundant signal set,” IEEE Commun. Magazine, vol. 27, pp. 5-21, Feb. 1987.
- [28] L. J. Cimini, Jr., and N. R. Sollenberger, “OFDM with diversity and coding for advanced cellular internet services,” Proc. Of Globecom '97, pp. 305-309, November 1997.
- [29] J. Kim, L. Cimini, and J. C. Chuang, “Coding strategies for OFDM with antenna diversity for high bit rate mobile data applications,” in Proc. VTC'98, pp. 743-747.
- [30] L. J. Cimini Jr., “Multicarrier Modulation for data transmission: An idea whose time has come,” IEEE Commun. Mag., vol. 28, pp. 5-14, May 1990.
- [31] M. Okada, S. Hara and N. Morinaga, “Bit error rate performances of orthogonal Multicarrier modulation radio transmission schemes, “IEICE Trans. Commun. vol. E76-B, pp. 113-119, Feb. 1993.
- [33] Leonard J. Cimini Jr., and Justin C. -I. Chuang, “Turbo Codes for OFDM with Antenna Diversity”. IEEE Publications 1999.
- [34] D. Agarwal, V. Tarokh, A. Naguib, N. Seshadri, “Space time coded OFDM for high data rates wireless communication over wideband channels”, in Proc. VTC'98, Ottawa, Canada, May 18-22, 1998.
- [35] Alister Burr and George White, “Performance of Turbo-Coded OFDM” European Cooperation in the field of scientific and technical research, January 2000.
- [36] Leonard J. Cimini Jr., Justin C.,-I., Chuang, “Comparison of convolutional and turbo codes for OFDM with Antenna Diversity in high bit rate wireless applications”, IEEE Commun. Letters, Vol. 4, No. 9, September 2000.
- [37] J. Grim, M. P. Fitz, J. V. Krogmeier, “Further results on space time coding for Rayleigh fading” presented at 36th Annual Allerton conf. On Commun. Control and Computing, Monticello, USA, September 1998.
- [38] S. Baro, G. Bauch, A. Hansmann, “New trellis codes for space time coded modulation”, ITG conference, Munich 1999.

- [39] Benedetto, S., Divsalar, D., Montorsi, G. and Pollara, F. (1996), "Serial Concatenated of Interleaved Codes: Performance Analysis, Design, and Iterative Decoding", *The TDA Progress Report*, Jet Propulsion Laboratory, Aug., 1996, pp. 1-26.
- [40] Benedetto, S., Divsalar, D., Montorsi, G. and Pollara, F. (1998), "Serial Concatenated of Interleaved Codes: Performance Analysis, Design, and Iterative Decoding", *IEEE Transactions on Information Theory*. Vol. 44, No. 3, May 1998, pp. 909-926.
- [41] Le Nhat Thang*, R.M.A.P Rajatheva "PERFORMANCE OF PARALLEL CONCATENATED CONVOLUTIONAL CODES (TURBO CODES) AND SERIAL CONCATENATED CONVOLUTIONAL CODES IN WIDEBAND DS-CDMA" Posts and Telecommunication Institute of Technology (PTIT), Hanoi, Vietnam, Asian Institute of Technology (AIT), Bangkok, Thailand
- [42] "Comparison of Convolutional and Turbo Codes for OFDM with Antenna Diversity in High-Bit-Rate Wireless Applications" Lang Lin, *Student Member, IEEE*, Leonard J. Cimini, Jr., *Fellow, IEEE*, and Justin C.-I. Chuang, *Fellow, IEEE*