INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

ProQuest Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
800-521-0600

Trellis-Based Iterative Decoding of Block Codes for Satellite ATM

Usa Vilaipornsawai

A Thesis

in

The Department

of

Electrical and Computer Engineering

Presented in Partial Fulfillment of the Requirements

for the Degree of Master of Applied Science at

Concordia University

Montréal, Québec, Canada

Febuary 2001

ABSTRACT

Trellis-Based Iterative Decoding of Block Codes for Satellite ATM

Usa Vilaipornsawai

In power-limited channels such as satellite links, error control codes are needed to provide high performance at very low Signal-to-Noise Ratios (SNRs). Turbo Codes or concatenated codes with iterative decoding, can achieve a performance very close to the Shannon limit. They are the most powerful codes known so far and are considered for use in various applications.

In this thesis, we present the turbo codes using Reed-Muller (RM) codes as their component codes, and where the decoding method used is the trellis-based iterative Maximum A posteriori Probability (MAP) algorithm which provides an optimal performance. We aim to apply our turbo coding scheme to Asynchronous Transfer Mode (ATM) applications in Digital Video Broadcast - Return Channel via Satellite (DVB-RCS). Therefore, the shortened versions of ordinary RM-turbo codes are proposed. In some cases, the shortened codes demonstrate Unequal Error Protection (UEP) property. The UEP codes are suitable for ATM applications since in an ATM cell, the cell-header is more important than its payload and requires a higher protection level.

As well, we investigate the effect of phase offset on the shortened RM-turbo code performance with Quadrature Phase Shift Keying (QPSK) modulation scheme and

the effect of preamble size used to recover carrier phase due to the problem of synchronization which is usually raised in practical systems. Moreover, in an iterative MAP decoding scheme, the channel SNR is essential in the calculation of the MAP soft-outputs. Hence, the effect of channel SNR mismatch on the performance of RM-turbo codes is explored. It is shown in this thesis that our proposed turbo coding schemes are suitable for satellite ATM application. First, it is because we use a block code as component code which is suitable for the fixed size of an ATM cell. Second, it provides good performance with special features such as UEP property. Third, it can tolerate channel SNR mismatch to a satisfactory level i.e. in a range from -2 to 6 dB channel SNR mismatch which is typical of available estimation algorithms. Fourth, a byproduct of shortening the RM-turbo code, we can send the shortened zeros as the preamble bits and can easily use them to recover carrier phase.

Dedicated to my parents, brother and sisters.....

ACKNOWLEDGEMENTS

First of all, I would like to express my sincere gratitude to my supervisor, Dr. Mohammad Reza Soleymani for giving me the opportunity to work on this interesting research topic. I really appreciate the way he devoted his precious time to discuss with me and encourage me throughout my reseach work. I could not have asked for a better, more supportive supervisor than him. He has been everything that one could want in his/her supervisor.

My heartfelt thanks to my parents, brother and sisters who supported me with their love and understanding which, I beleive is the fundamental basis for my acheivement. I hence would like to decicatc this thesis to them.

I would also like to thank all my teachers who were of great support throughout my academic life.

I also extend my thanks to National Science and Technology Developement Agency (NSTDA), Thailand and National Research Council of Canada (NRC) for giving me the opportunity to participate in the "Women in Engineering and Science" (WES) program in Canada which gave a boost to my enthusiasm to learn and to be strong. The motivation I got from my supervisor at NRC Dr. Bas Baskaran should not go without any praise. Dr. Bas has had a very significant impact on my life. He was more like a mentor to me, as I was looking upto him for opinions/suggestions whenever I needed them, be it academic or personal. He would always provide me with a good advice alongwith suggesting suitable alternatives and respecting my decision. Thus, I would like to thank Dr. Bas, his family and his sister-in-law, Bharathi for their support and encouragement.

I also would like to thank Edith Corriveau, my first room-mate, her Mom, brother, sisters and friends for their friendship and support while I was in Ottawa. Thanks

to Vlad who guided me initially by sharing his knowledge and expertise on RM and RS encoders and decoders. My thanks to Sanjeev, Wantawin, Mohammad, Vlad and Najam for their help in proof-reading my thesis. Thanks to Nicha, Jariya and other Thai friends who were very good to me. Thanks to Su, Sam and Sanjeev to be my good friends. I finally would like to thank all other friends back in Thailand and at Concordia.

TABLE OF CONTENTS

LIST OF FIGURES

Chapter 1

Introduction

1.1 Introduction

In this era, distance cannot prevent people from communicating in any part of the world. Nowadays, activities such as talking on the phone with your family in Asia, using electronic mail to keep in touch with friends, ordering a book through internet, watching live soccer game in England via satellite video broad-casting and even having tele-surgery and tele-conference are very normal. Those activities are the results of the development of communication systems that provide cost-effective, reliable and high speed data transfer. In addition, this era is the time for digital

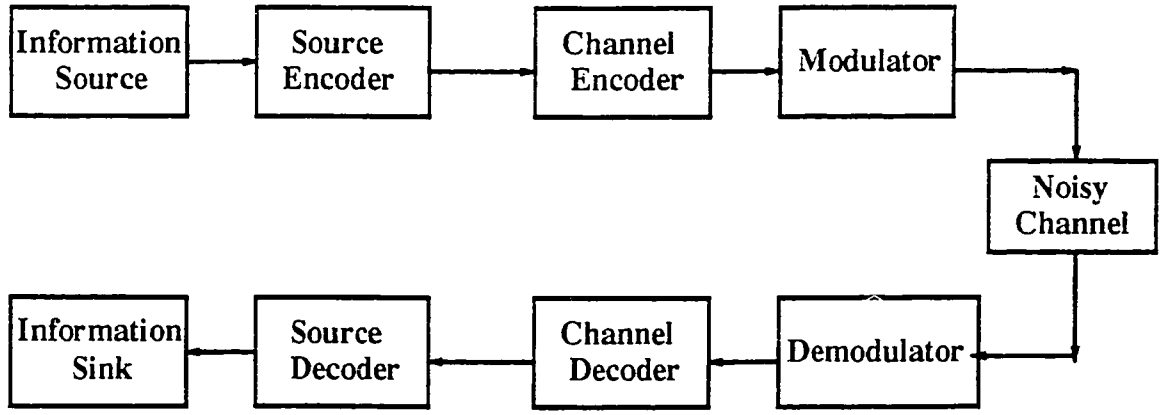

Figure 1.1: Digital communication system

signals rather than analog signals because they are superior in many ways. They can be regenerated at intermediate points along long-distance transmissions so noise can be removed at each point. This is in contrast to analog signals where noise is amplified along with the signal when the transmitted signal is amplified along the way. Moreover, the redundancy in digital data can be eliminated whereas in analog signal there is high redundancy. Furthermore, the implementation of a digital system is often cheaper than that of an analog system.

Figure 1.1 shows a typical *digital communication system*. An information source could either be a human being or a machine and it produces a source signal. There are two kinds of source signals, one is an analog signal, for example, an audio/video signal and the other is a digital signal such as a data stream from a computer. In digital communications, we can assume that all signals are in digital form because the analog data can be converted to digital data by using Analog to Digital Converter (ADC). First the digital signal is processed at the source encoder which eliminates the redundancy in the signal in order to represent data with the smallest number of bits where perfect (or acceptable) reconstruction can be achieved. We shall call the output data from the source encoder the information sequence. The information sequence is then encoded by a channel encoder where the controlled redundancy, the so-called "parity" is added. The output obtained from the encoding process is the coded sequence and it provides the error detecting and/or error correcting capability to the coded system. Following that modulation maps each symbol of the discrete coded sequence to a corresponding signal waveform that is more suitable for physical channels such as the terrestrial radio, telephone line, satellite link, etc where noise corrupts the signal.

At the receiver end, the demodulator processes the corrupted waveform and provides the estimated digital coded sequence. Then the sequence is passed to the channel decoder where it provides the estimated information sequence whose errors

are detected and/or corrected . After that the source decoder takes the estimated information sequence and uses the knowledge from the source encoder to reconstruct the original signal and passes on the data which may be converted back to analog signal by using Digital to Analog Converter (DAC) to an information sink (the user).

In this thesis, we focus on the *channel coding*. Channel coding provides protection to information resulting in higher reliability of a transmission system. Channel coding reduces transmitted power where we can consider this power saving in terms of coding gain which is the lesser amount of Signal to Noise Ratio (SNR) required to achieve the same Bit Error Rate (BER) compared to an uncoded system. The SNR is in terms of $\frac{E_b}{N_o}$ where, E_b is energy per information bit and N_o is the one-sided noise power spectral density. Since coding gain provides the reduction in transmitted power, a low cost system can be established. Therefore, a powerful coding scheme is sought to achieve such a coding gain. Nevertheless, the selection of the coding scheme for specific system depends on some constraints and requirements such as channel environment, complexity, performance, speed, bandwidth etc. For example, at the same code rate and decoding complexity, the convolutional code outperforms the Reed-Solomon (RS) code at low $\frac{E_b}{N_o}$, region ($\leq$ 4.5 dB) with moderate performance (BER down to about 10^{-5}) while the RS code obtains better performance at higher $\frac{E_b}{N_o}$. It other words, convolutional codes should be used for poor channels with some intermediate reliability and RS code should be used for less severe channels with very high performance. Also, a RS code copes well with burst error, whereas a convolutional code which has soft decision decoding algorithm performs well at low SNRs. However, convolutional codes cannot tolerate burst errors and even causes burst errors themself. Hence, the applications of RS codes are usually on bursty channels such as multi-path fading channels, Compact Disc (CD), Magnetic disks, etc., whereas, convolutional codes have impact on power-limited channels in which the SNR required is very low such as in satellite links or deep space applications.

Later on a combination taking the advantages of both codes was introduced. It is the concatenation of a RS code and a convolutional code. The code is constructed by using a RS code as an outer code and a convolutional code as the inner code, where the information is first encoded by the RS encoder and then fed to the convolutional encoder. The decoding process is in the reverse order. The received sequence is decoded by the soft decision decoding algorithm in the convolutional decoder and then decoded by an algebraic RS decoder to clean up the residual errors. Because of its good performance, concatenated codes [2] are widely used instead of convolutional codes in many applications such as in deep space, satellite and wireless communications.

The concatenated coding schemes were the dominant channel coding schemes until the most powerful codes were developed by Berrou, Glavieux and Thitimajshima [3]. They are called *Turbo Codes*, concatenated convolutional codes with *iterative Maximum A Posteriori Probability (MAP) decoding algorithm*, where the performance is very close to the channel capacity. Turbo Codes kept all coding theorists busy since the time they were introduced and they are considered to be used instead of many existing schemes for the applications where good performance with very low SNR is required.

There are two main groups of turbo codes. One of them is the *Convolutional Turbo Code* (CTC) and the other is *Block Turbo Code* (BTC). The difference between them is the component code used. The CTCs use convolutional codes as their component codes, in contrast to the BTCs which use block codes as the component code. This chapter contains the basic concept of turbo codes, some literature reviews in the area of CTCs and BTCs, the objectives, the contributions and scope of the thesis.

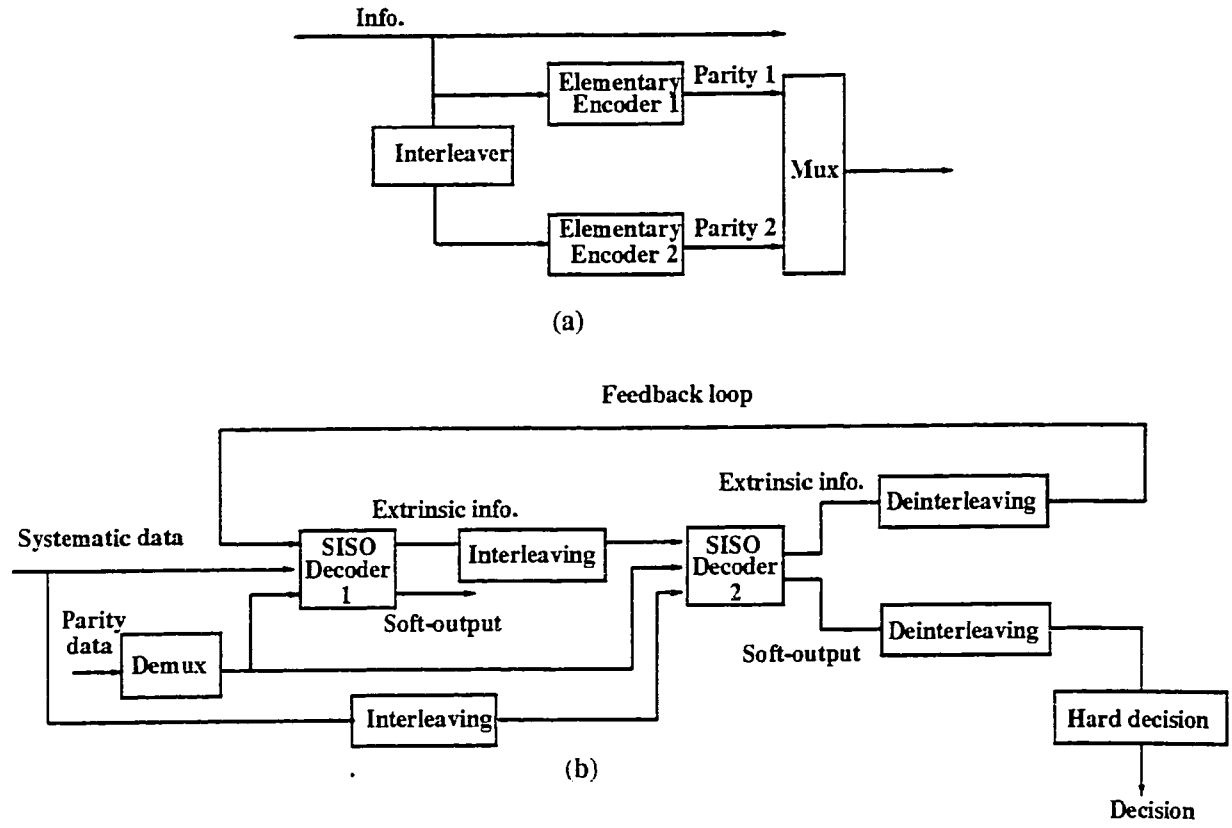

Figure 1.2: Turbo codes (a). Turbo encoder (b) Turbo decoder

1.2 Basic Concept of Turbo Codes

The main ingredients of a two-dimensional turbo code, with straight forward generalization for multi-dimensional codes, are two elementary encoders, one interleaver and two *Soft-Input Soft-Output* (SISO) decoders. The elementary codes can be either convolutional or block codes and the turbo code can be parallel or serial concatenated code. In this section we will explain the general idea of Turbo coding by giving an example of a parallel concatenated code, the details of a serial can be found in the next chapter.

In the parallel turbo encoder shown in Figure 1.2 (a), the information is encoded twice by two systematic encoders where the inputs of the encoders are the original

information and its interleaved version. Each encoder produces a code sequence consisting of information and parity subsequences whereas in a turbo coded sequence, only one information part will be sent along with the two parity parts from two encoders because the other information portion is completely redundant. The decoding scheme shown in Figure 1.2 (b) is the iterative decoding of two SISO decoders where each of them accepts soft-input and produces extrinsic information and exchanges it between them in an iterative decoding process. In the final decoding step, hard decision on information is provided based on the soft-output information. It is important to note that a deinterleaver and an interleaver are needed before the first and second SISO decoders, respectively, corresponding to the elementary encoder 1 and 2 which operate over non-interleaved and interleaved information.

In order to explain turbo coding concept in an easy way, an analogy is presented. Suppose there is an object that is observed by two people from different views and the description of the object is given by each person. The description is then passed on to another pair of people, each of them getting one of the descriptions. The discussion process starts (at the receiving end) to find out what the object has been. Each analyzes his/her data and provides the information in his/her view to the other. The discussion will go on until a consensus is reached and a final decision is made. In comparing this example with turbo coding, we can think of the first pair people as the two encoders, an object as the information sequence and the other two people as the two SISO decoders. Whereas the information gathered from the other angle could be considered as the same information obtained after interleaving and the discussion process as the iterative decoding. Actually this concept is not new for human communication, but it is, for machines because now we can make machines that communicate or talk to each other in order to share information.

1.3 Literature Review of Convolution Turbo Codes.

Since the invention of Turbo Codes, there have been many publications which relate to this topic. In this section, we will cite some papers presented on Convolution Turbo Codes.

The interleaver is an important feature behind the achievement of the turbo code performance because it produces long block length leading to large distance, it scrambles input data for the second elementary encoder resulting in more diversity information so that the iterative decoding can be used. An interleaver also influences the distance spectrum of turbo codes and the termination of the trellises. It is known that performance of a code depends on the weight distribution of the code and at high SNRs the code words with minimum distance affect the error probability of the code [6]. Since the component encoders are recursive, the problem of trellis termination arises, however, it can be eliminated by properly designing interleaver as presented in [7]-[10]. In addition, in [10] and [11], the interleaver is designed to increase minimum distance by breaking short length input patterns which are likely to generate low weight code words

So far, no tight bounds at low SNRs has been found for turbo codes. Fortunately, in this region, simulation can be conducted easily, as compared to high SNRs where simulation is highly computation intensive which makes analytical performance evaluation crucial. In performance analysis, the weight distribution of the code is needed, however, the weight distribution is hard to find because the encoder is recursive. Therefore, the average weight function independent of interleaver type is introduced in [12], [13] and the average upper bound of turbo code performance on an AWGN channel is analyzed. The analysis is later extended on a Rayleigh fading channel in [14].

In typical modern communication, a receiver consists of a series of subsystems. For example, a cascade of antenna array, equalizer, multi-user detector, channel decoder and source decoder. Before the idea of SISO and iterative technique was proposed, hard information was passed on from one subsystem to another and each subsystem was optimized independently. However, after the idea of passing soft information in an iterative fashion was realized, the combination of the subsystems to achieve a better overall performance was introduced. These include the combination of turbo coding with source coding [15], and with an equalizer [16], [17]. The application of turbo code on the transmit and receive diversity, space time codes is presented in [18], [19].

In a band limited channel like a telephone line, trellis-coded modulation (TCM) is used because high coding gain can be obtained without any loses in bandwidth. Once again turbo code is applied instead of convolution code to obtain high spectral efficiency. This idea was first introduced in [20] and has been investigated more in [21], [22]. In addition, some applications of turbo codes and their variations are found in [10], [23]-[26] .

1.4 Literature Review of Block Turbo Codes.

Along with the development of the CTCs, the concept of iterative decoding was also applied to block codes. Two different Soft-Input Soft-Output (SISO) decoding methods for BTCs are the *trellis-based algorithm* [31] and the *Augmented List Decoding algorithm* [27]. Therefore, we will classify our literature review on the topic of BTCs into two parts as follows:

1.4.1 Trellis-Based Algorithm

In trellis-based algorithm, which is based on the Bahl-Cocke-Jelinek-Raviv (BCJR) algorithm [5], the state sequence of discrete-time finite state Markov process in memoryless channel is estimated, and such a process can be represented by a trellis diagram. Both convolutional and block code have a trellis diagram representation. Lodge et al. [28] presented the separable MAP-filters approach for decoding the multi-dimensional product codes and extended to concatenated convolutional codes with interleaver where the extrinsic information, called refinement factors, is passed from one decoding process to another through an iterative process. This paper was one year before the introduction of Turbo Code by Berrou et al. [3]. It was also presented at the same conference [29] that Berrou et al. presented their famous paper, which means that BTC was developed even before CTC. In [30], the MAP decoding algorithm in log domain was developed. Then, Hagenauer et al. [31] presented a clear concept and solid mathematical framework for iterative decoding of both convolutional and linear block codes for MAP algorithm and its variants. For linear block codes, the MAP decoding algorithm can be performed using either the trellis of a code or the trellis of its dual, where the decoding complexity can be reduced if the dual code trellis is used when $n - k < k$ where n and k are the code length and the information length, respectively. Moreover, the stopping criterion using cross-entropy is investigated for sake of reducing complexity. The use of BTCs in a concatenated scheme was presented by Y. Liu and S. Lin in [32] using Reed-Solomon code as the outer code and Hamming turbo product code as the inner code. Also, the new stopping criterion is proposed and applied to inner iterative decoding and the effect of parallel and serial decoding of turbo decoder is investigated and it is shown that the parallel one outperforms the serial one. In addition, in [33] Y. Liu et al. propose the trellis-based MAP decoding algorithm based on the sectionalized trellis of linear block codes. An optimal sectionalized trellis is considered the best trellis under some conditions such as minimizing the

number of multiplication operations. In this paper the optimal sectionalization of Reed-Muller (RM) codes are found. The analysis of computational complexity and storage space are investigated. It is important to note that the permutation of encoding sequence is essential for the technique of bidirectional decoding algorithm that computes the backward and forward recursion simultaneously resulting in delay reduction. Also the parallel MAP decoding algorithm is considered for RM codes by decomposing the trellis structure into identically parallel sub-trellises without cross connections among them and without exceeding the maximum state complexity of the trellis. By doing this , the decoding delay is reduced and the decoding process is speeded up making it suitable for hardware implementation.

1.4.2 Augmented List Decoding

In this turbo decoding category, a list of candidate code words are produced with different methods such as the Chase-II algorithm [4] used in [34], the Pseudo-Maximum-Likelihood (PLM) algorithm used in [40] and the Fang-Battail-Buda-Algorithm (FBBA) used in [27]. List decoding is soft decision decoding of linear block codes because the list of candidate code words are provided rather than just one alternative.

Pyndiah and his co-authors published a series of papers [34]-[38]. They used product code or a serial concatenation of block codes with a block interleaver using SISO decoder based on modified Chase algorithm. The main idea of this algorithm is to reduce number of the reviewed code words in a set of highly possible code words by using channel information. First, the set of error patterns, E is produced based on the reliability of received sequence, then the set of test patterns, T is generated, where $\vec{t} = \vec{e} + \vec{h}$, $\vec{t} \in T$, $\vec{e} \in E$ and $\vec{h}$ is the hard decision vector of a received sequence. Each test pattern is decoded by an algebraic decoder and the set of candidate code words is generated. The decision is the code word with the

highest correlation with received sequence among the candidate code words. The soft-output of a given bit is calculated from received vector, the decision code word and the completing code word or the higher correlation among code word in the candidate set with bit at a given position is different from the bit at that position of decision code word.

The performance of BTCs using BCH codes as their component codes over a Gaussian channel was presented in [34]. The results show the attractiveness of BTCs for the applications that require very good performance with high code rate $R > 0.8$. The extension of this paper is presented in [35] with results for both AWGN and Rayleigh channels. It is shown that more than 98% of channel capacity can be achieved with a high code rate. A further investigation using Reed-Solomon codes as component codes was presented in [36] with an attempt to apply BTC to data storage applications.

The most significant drawback of turbo decoder is its complexity; thus, [37] presents the methods of reducing complexity of turbo product code by reducing the number of test patterns and using the previous decision code word as the competing code word for the next iteration. Results show that the complexity is reduced almost a factor of ten compared to [34] with a performance degradation of 0.7 dB. In [39] fast Chase algorithm is proposed by ordering test patterns before feeding to algebraic decoder in such a way that the operations in syndrome and metric calculation are reduced without performance degradation. Some recent improvements on the BTC in performance and implementation matters are presented in [38], [41] and [42]. The application of block turbo codes in wireless packet transmission is presented in [43], [27]. In [43], the PLM algorithm is used, whereas in [27] the FBBA algorithm is applied and Unequal Error Protection (UEP) property of Generalized Turbo Product Code (GTPC) is also introduced.

1.5 The Objectives

The main objectives of this thesis is to develop the alternatives to the current baseline coding scheme used for the satellite ATM in Digital Video Broadcast - Return Channel via Satellite (DVB-RCS). Moreover, the sensitivity of the developed coding schemes to channel impairments is also our focus.

1.6 The Main Contributions of Thesis

1. Development of block turbo coding schemes having Reed-Muller (RM) codes as their component codes, suitable for ATM transmission over wireless and satellite links. This includes DVB-RCS.

2. Performance evaluation of the developed coding schemes over AWGN and Rayleigh-fading channels using trellis-based iterative MAP decoding.

3. Design of shortened RM-turbo codes with different shortening patterns fitting into satellite ATM. In some cases, the shortened versions of Turbo-RM codes obtain Unequal Error Protection (UEP) property. This property is suitable for connection-oriented networks such as ATM network because cell-header contains routing information, flow control information, etc., and is more important than its payload.

4. Investigation of the effect of channel impairments including channel SNR mismatch and phase offset on the performance of shortened RM turbo codes using QPSK modulation scheme as well as the effect of preamble size used to estimate carrier phase.

1.7 The Scope of Thesis

This thesis is organized as follows:

Chapter 2 provides a brief introduction to error control coding. The code construction and detail of two classical codes: the linear block code and convolutional code are discussed. Also the details of new class of code, "Turbo Codes" is presented, particulary, in parallel and serial turbo encoder and decoder.

Chapter 3 contains the construction of a RM code and its minimal trellis using Massey algorithm. The Maximum-Likelihood (ML) or the Viterbi decoding for linear block code is discussed. It also includes the example of how to construct and draw trellis of RM (8,4) with the simulation results for both hard and soft decision decoding. Finally, the performance of the concatenated code with Reed-Solomon (RS) (73,57) and the RM (8,4) inner codes as opposed to the existing system which uses Convolutional rate of $\frac{1}{2}$ as an inner code is shown.

Chapter 4 covers the topics related to RM turbo code, including the encoder and the iterative MAP decoder of a two-dimensional linear block code, particularly for the RM-turbo codes. The simulation results of the RM turbo codes both in Gaussian and Rayleigh-fading channels are presented.

Chapter 5 presents the shortened RM turbo code proposed for use in cell-based transmission. First, an overview of ATM networks is presented. Then the design of shortened RM-turbo codes with different shortened patterns is presented because in order to fit turbo-RM code in an ATM cell, the codes must be shortened. Finally, simulation results of different shortened RM-turbo codes are shown as well as the performance of different regions of the shortened patterns is presented.

Chapter 6 contains the investigation of the effect of channel impairments including the channel SNR mismatch and phase offset as well as the effect of preamble size used to recover carrier phase.

Chapter 7 presents the conclusion of the results and suggestions for future research.

Chapter 2

Background of Error Control Coding

2.1 Introduction

The two classical theorems in *information theory* proven by Shannon were published in his famous paper "A Mathematical Theory of Communication" [1].

The first theorem, the *source coding* theorem, is concerned with the compression of the source information. According to this theorem, the smallest number of bits required to represent a given source without any loss is its *entropy*.

The second theorem, the *channel coding* theorem, shows that in a communication channel, the error-free transmission or communication with an arbitrarily low bit error rate can be achieved, as long as the data is transmitted using a error correcting code at a rate less than the *channel capacity*. It was contrary to the general belief at that time that there was no way that information could be passed through a noisy channel without error. The concept of channel coding is to add redundancy, in a controlled manner, to protect information when it is sent through a noisy channel for reliability.

In this thesis, channel coding is our subject of interest. A brief preview of the two basic classes of codes used in communication systems for error control coding, *block* and *convolution* codes [44], [45] is presented. Also a new class of code called *Turbo Code* will be discussed. We start with the discussion of the overall concept of turbo codes and some mathematical background used in decoding algorithm, followed by the idea of parallel and serial turbo encoders and decoders.

2.2 Linear Block Codes

An (n, k) *block code* over the Galois Field , $GF(q)$[1]where q is a prime number, consists of $M = q^k$ code words of length n with components taking values from $GF(q)$. The code can be thought of as a k-dimensional subspace of the n-dimensional space. The ratio of $\frac{k}{n}$ is called *code rate*.

A block code has *linear* property when linear combination of any two code words is another code word where the operations are done i.e., modulo-q. Here, we consider only binary linear block code, i.e., $q = 2$ is assumed. In this case, linearity property implies that the modulo-2 sum of any two code words is another code word.

A linear block code can be constructed from a matrix, G with k rows as the bases of code words for the (n, k) code C. We call this matrix, a *generator* matrix. A linear block code, $\overrightarrow{c} = (c_0, c_1, ..., c_{n-1})$ obtained from the input message sequence of length k, $\overrightarrow{m} = (m_0, m_1, ..., m_{k-1})$ over $GF(2)$ is given as,

$$\overrightarrow{c} = \overrightarrow{m} \times G \tag{2.1}$$

where G is the $k \times n$ generator matrix with k linearly independent code words as its rows and the multiplication is performed over $GF(2)$.

[1]**Galois Field q is the finite field of the integer modulo q**

The code is considered to be *systematic* when the generator matrix is of the form

$$\widetilde{G} = \left[I_k \vdots P \right] \tag{2.2}$$

where I_k is a $k \times k$ identity matrix and P is a $k \times (n-k)$ parity matrix.
The systematic code of the input sequence $\overrightarrow{m}$ generated by $\widetilde{G}$ contains information bits located as the first k bits of the code words. The advantage of a systematic code is that at the decoder, the first k bits are simply the estimation of the k information bits. No attempt is needed to recover information bits from a code word. In a systematic code, the code word bits are given by

$$c_i = \begin{cases} m_i & 0 \leq i \leq k-1 \\ \sum_{j=1}^{k} p_{ji} m_j & k \leq i \leq n-1 \end{cases} \tag{2.3}$$

However, it is not necessary that the positions of the information bits be at the first k bits. They can be at other coordinates. The generator matrix used to construct code word where we can specify information positions is considered to be in systematic form.

2.3 Convolutional Codes

The concept of a *convolutional code* is different from that of a block code. A *continuous data sequence* is considered rather than *fixed block information.* A convolutional code with rate $\frac{k}{n}$ is described. The portion of k bits in information sequence is fed to an encoder using linear shift registers where n-bit output or code word is provided. First, a continuous data sequence M which can be demultiplexed in to k subsequences $M^{(i)} = \left(m_0^{(i)}, m_1^{(i)}, m_2^{(i)}, ..., m_p^{(i)}, ... \right)$, $i = 0, 2, ..., k-1$, is fed to convolutional encoder. The encoder produces n subsequences $U^{(j)} =$

$\left(u_0^{(j)}, u_1^{(j)}, u_2^{(j)}, ..., u_p^{(j)}, ...\right)$, $j = 0, 2, ..., n-1$. The output subsequences are multiplexed to obtain the code word $\overrightarrow{c} = \left(u_0^{(0)}, u_0^{(1)}, ..., u_0^{(n-1)}, ..., u_1^{(0)}, u_1^{(1)}, ..., u_1^{(n-1)}, ...\right)$. The convolutional encoder consists of k shift registers for each input subsequence. For $1 \leq i \leq k$, the i^{th} shift register has length l_i which is also the number of memory elements. The *total memory* in the encoder is

$$L = \sum_{i=0}^{k-1} l_i \tag{2.4}$$

The *memory order* is defined as the maximum number of memory elements in an input register and given by

$$v = max_{0 \leq i < k-1} l_i \tag{2.5}$$

Constraint length is defined as the number of bits in an output subsequence which are affected from any input bit. The *constraint length* of a code is given by

$$K = 1 + v \tag{2.6}$$

A set of parameters (n, k, v) is usually used to specify a convolutional code with rate $\frac{k}{n}$ and memory order v. The generator matrix of a convolutional code is

$$G = \begin{bmatrix} G_0 & G_1 & G_2 & \cdots & G_v & & \\ & G_0 & G_1 & \cdots & G_{v-1} & G_v & \\ & & G_0 & \cdots & G_{v-2} & G_{v-1} & G_v \\ & & & \ddots & & & \ddots \end{bmatrix} \tag{2.7}$$

where G_0 and G_v are non zero sub-matrixes and the sub-matrix, G_i, for $0 \leq i \leq v$

is given by

$$G_i = \begin{bmatrix} g_{0,i}^{(0)} & g_{1,i}^{(0)} & \cdots & g_{n-1,i}^{(0)} \\ g_{0,i}^{(1)} & g_{1,i}^{(1)} & \cdots & g_{n-1,i}^{(1)} \\ \vdots & \vdots & \vdots & \vdots \\ g_{0,i}^{(k-1)} & g_{1,i}^{(k-1)} & \cdots & g_{n-1,i}^{(k-1)} \end{bmatrix} \tag{2.8}$$

Where $g_{out,mem}^{(in)}$ is an element in the generator sequence $g_{out}^{(in)} = (g_{out,o}^{(in)}, g_{out,1}^{(in)} \cdots, g_{out,v}^{(in)})$ which is the impulse response obtained at output *out* by applying a single 1 at input *in* followed by zeros. The convolutional code word, $\overrightarrow{c} = (c_0, c_1, ..., c_{n-1})$ generated from message sequence $\overrightarrow{m}$ and generator matrix G is given by

$$\overrightarrow{c} = \overrightarrow{m} \times G \tag{2.9}$$

The s^{th} output code word at time p for $0 \leq s \leq n$ and $p \geq 0$ is given by

$$c_p^{(s)} = \sum_{i=0}^{k-1} m_p^{(i)} g_{s,0}^{(i)} + \sum_{i=0}^{k-1} m_{p-1}^{(i)} g_{s,1}^{(i)} + \cdots + \sum_{i=0}^{k-1} m_{p-v}^{(i)} g_{s,v}^{(i)} \tag{2.10}$$

2.4 Turbo Codes

In general, a turbo encoder can be considered as a two- or multi- step encoder. It is the machine to construct either parallel or serial concatenated codes with interleavers. This encoding technique provides a diversity of information which will be useful for iterative decoding scheme.

For a turbo decoder, the important element is the *Soft-Input Soft-Output* (SISO) decoder where the soft-outputs are obtained from the *Log-Likelihood-Ratio* (*LLR*) of a posteriori probability which we will discuss below.

2.4.1 Log-likelihood Ratio of A Posteriori Probability

Let the null element of $GF(2)$ with element $\{+1, -1\}$ be -1. The natural logarithm is used through out for the formulae in this thesis.

The log-likelihood ratio (LLR) of a binary random variable X, $L_X(x)$ is defined as

$$L_X(x) = log\frac{P_X(x=+1)}{P_X(x=-1)} \tag{2.11}$$

Where $P_X(x)$ is the probability that the random variable X takes on the value $x \in \{+1, -1\}$. When the (n, k) systematic code is sent through a Gaussian channel, the log-likelihood ratio of bit x condition on the received bit y at the detector is given by,

$$\begin{aligned} L(x \mid y) &= log\frac{P(x=+1 \mid y)}{P(x=-1 \mid y)} \\ &= log\left(\frac{p(y \mid x=+1)}{p(y \mid x=-1)} \cdot \frac{P(x=+1)}{P(x=-1)}\right) \\ &= log\frac{e^{\left(-\frac{E_s}{N_o}(y-a)^2\right)}}{e^{\left(-\frac{E_s}{N_o}(y+a)^2\right)}} + log\frac{P(x=+1)}{P(x=-1)} \\ &= L_c \cdot y + L(x) \end{aligned} \tag{2.12}$$

where $L_c = 4 \cdot a \cdot \frac{E_s}{N_o}$ is called the *reliability value* of the channel, a is the fading attenuation whereas for Gaussian channel a equals one and $\frac{E_s}{N_o}$ is the channel SNR estimated at the receiver. The $L(x \mid y)$ is a posteriori log-likelihood ratio or the soft output which is the combination of LLR of channel measurement and of priori probability. We consider a memoryless channel. The soft output of bit x_k, $L(\widehat{x}_k)$ is a posteriori log-likelihood ratio of a bit x_k conditioned on the received sequence, $\overrightarrow{y}$, $L(x_k \mid \overrightarrow{y})$ which is given below

$$
\begin{aligned}
L(\widehat{x}_k) &= log\frac{P(x_k=+1 \mid \overrightarrow{y})}{P(x_k=-1 \mid \overrightarrow{y})} \\
&= log\left(\frac{p(\overrightarrow{y} \mid x_k=+1)}{p(\overrightarrow{y} \mid x_k=-1)} \cdot \frac{P(x_k=+1)}{P(x_k=-1)}\right) \\
&= log\left(\frac{p(y_k \mid x_k=+1)}{p(y_k \mid x_k=-1)} \cdot \prod_{i=1,i\neq k}^{n} \frac{p(y_i \mid x_k=+1)}{p(y_i \mid x_k=-1)} \cdot \frac{P(x_k=+1)}{P(x_k=-1)}\right) \\
&= log\left(\frac{p(y_k \mid x_k=+1)}{p(y_k \mid x_k=-1)}\right) + log\left(\prod_{i=1,i\neq k}^{n} \frac{p(y_i \mid x_k=+1)}{p(y_i \mid x_k=-1)}\right) + log\left(\frac{P(x_k=+1)}{P(x_k=-1)}\right) \\
&= L_c \cdot y_k + L_e(\widehat{x}_k) + L(x_k) \qquad (2.13)
\end{aligned}
$$

where $L_c = 4 \cdot \frac{E_s}{N_o}$. The estimated soft-output of the information bit x_k consists of three terms which are the direct estimate of this bit obtained from the channel i.e. the *channel value*, $L_c \cdot y_k$, the *extrinsic information* given to this bit from other bits in the code word, $L_e(\widehat{x}_k)$ and a *priori value*, $L(x_k)$ which equals zero at the first iteration because at first it is assumed that the information bits have equal a priori probabilities. A priori value, $L(x_m)$ and received sequence, $\overrightarrow{y}$ are input to MAP decoder where it provides the soft output of bit x_m which consists of three terms as follows,

- The channel value, $L_c \cdot y_m$, which depends on the received bit y_m and channel reliability L_c.
- The extrinsic information $L_e(\widehat{x}_m)$, that is the extra information obtained from all other coded bits except x_m .
- A priori value $L(x_m)$ that is the a priori log-likelihood ratio of bit x_m. At first iteration, we assume that bit x_m has equal probability to be +1 or -1, so that $L(x_m) = 0$.

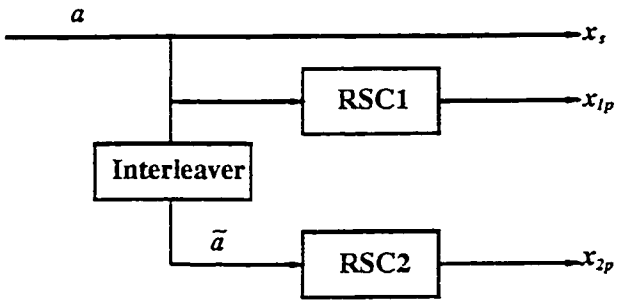

Figure 2.1: Structure of a turbo encoder.

2.4.2 Parallel Turbo Code

2.4.2.1 Encoder

In a two-dimensional *parallel turbo encoder*, information bits are encoded twice by using the original information and its interleaved version supplied to elementary encoder 1 and 2 respectively. In order to give credit to Berrou et al., the example of the original turbo code, the parallel convolution concatenated code, is presented. Figure 2.1 shows the turbo encoder consisting two *Recursive Systematic Convolutional* (RSC) encoders, RSC1 and RSC2, respectively. The information and its scrambled version are encoded by RSC1 and RSC2. The over all code rate is $\frac{1}{3}$, however, higher code rate can be obtained by puncturing some of its parity bits.

The RSC encoder is chosen to be an elementary encoder because of its systematic form which provides clear estimation on information bits for MAP decoding. Furthermore, the recursive encoder provides high weight parity or a *infinite impulse response* (IIR) even though the low weight information is encoded. Figure 2.2 shows RSC encoder for a rate 1/2 convolutional code with constraint length K and memory $\upsilon = K - 1$. The RSC is specified by two code generators $G_1 = \{g_{1i}\}$ and $G_2 = \{g_{2i}\}$ which is usually represented in octal form. One is used for forward encoding and the other is used for feedback encoding. The output at time j , $X_j = (x_{s,j}, x_{p,j})$ consists of the systematic bit, $x_{s,j}$ and the parity bit, $x_{p,j}$. Where systematic and

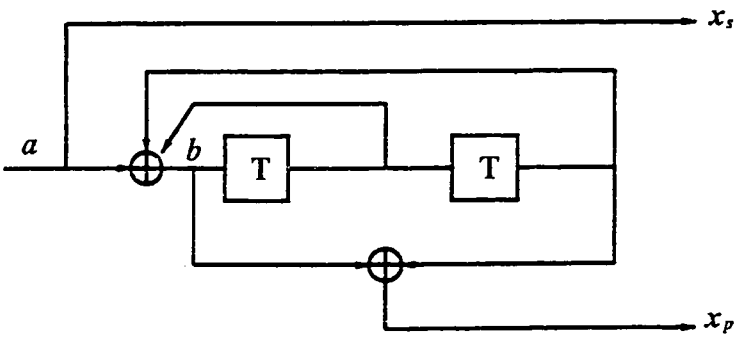

Figure 2.2: Example of Recursive Systematic Convolutional (RSC) encoder with $v = 2$, $R = 1/2$ and $G_1 = 7, G_2 = 5$

parity bits are given by

$$x_{s,j} = a_j \tag{2.14}$$

$$x_{p,j} = \sum_{i=0}^{v} g_{2i} b_{j-i} \quad g_{2i} = 0, 1 \tag{2.15}$$

Since the code is recursive, the input to the shift register is no longer the information input but the recursive input b_j which is given by

$$b_j = a_j + \sum_{i=1}^{v} g_{1i} a_{j-1} \tag{2.16}$$

Assume the Gaussian channel with BPSK modulation, the input to the decoder or output of the receiver filter are $Y = (y_{s,j}, y_{p,j})$ with

$$\begin{aligned} y_{s,j} &= (2x_{s,j} - 1) + n_{s,j} \\ y_{p,j} &= (2x_{p,j} - 1) + n_{p,j} \end{aligned} \tag{2.17}$$

where, $n_{s,j}$ and $n_{p,j}$ are two independent white Gaussian noise with zero mean and variance $\frac{N_o}{2}$. We consider just a *memoryless channel* in this thesis.

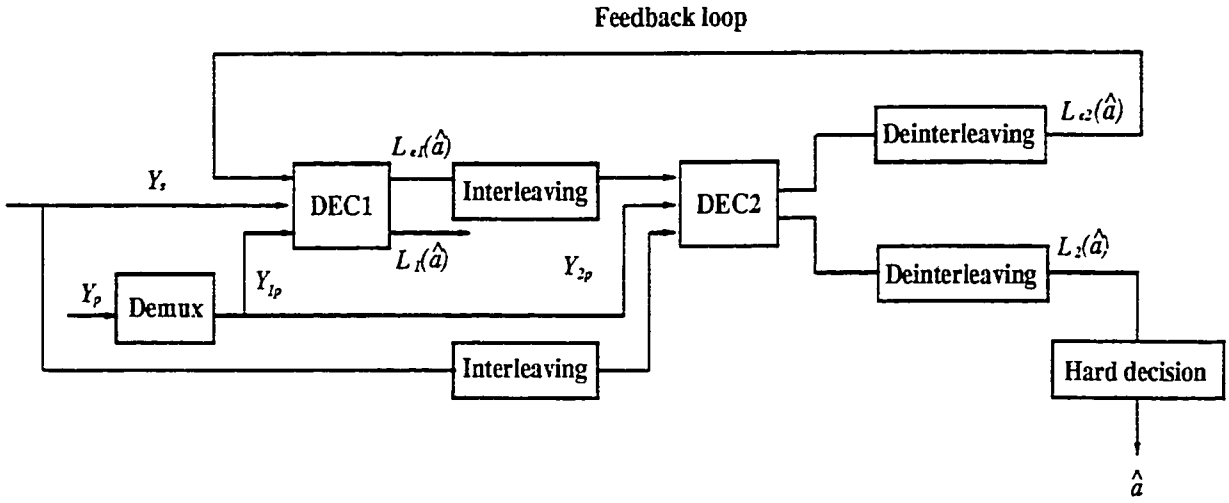

Figure 2.3: Parallel iterative decoder

2.4.2.2 Decoder

Figure 2.3 shows the *parallel iterative decoding* of a turbo code. The received parity sequence Y_p is demultiplexed to Y_{1p} and Y_{2p} which are the received parity subsequences for DEC1 and DEC2 repectively. DEC1 performs MAP decoding on the received information, Y_s, the received parity sequence, Y_{1p}, and the *a priori value*, $L(a)$ where the a priori value $L(a) = 0$ at first decoding step and then DEC1 provides the extrinsic information, $L_{e1}(\widehat{a})$. Similarly, the interleaved version of the noise corrupting sequence as well as a priori value obtained from DEC1 are fed into the MAP decoder, DEC2 where it provides the extrinsic value, $L_{e2}(\widehat{a})$. Keep in mind that DEC1 and DEC2 operate over non-interleaved and interleaved versions of the received sequence respectively. Thus, the input of DEC1 have to be de-interleaved if it is needed before being fed to the decoder. On the other hand, the inverse operation is applied to the input of DEC2. The soft-output provided by DEC1 and DEC2 for a_k at iteration $t \geq 1$ is given by the relation,

$$\begin{aligned} L_1^{(t)}(a_k) &= L_c \cdot y_k + L_{e2}^{(t-1)}(a_k) + L_{e1}^{(t)}(a_k) \\ L_2^{(t)}(a_k) &= L_c \cdot y_k + L_{e1}^{(t-1)}(a_k) + L_{e2}^{(t)}(a_k) \end{aligned} \tag{2.18}$$

Figure 2.4: Serial turbo encoder

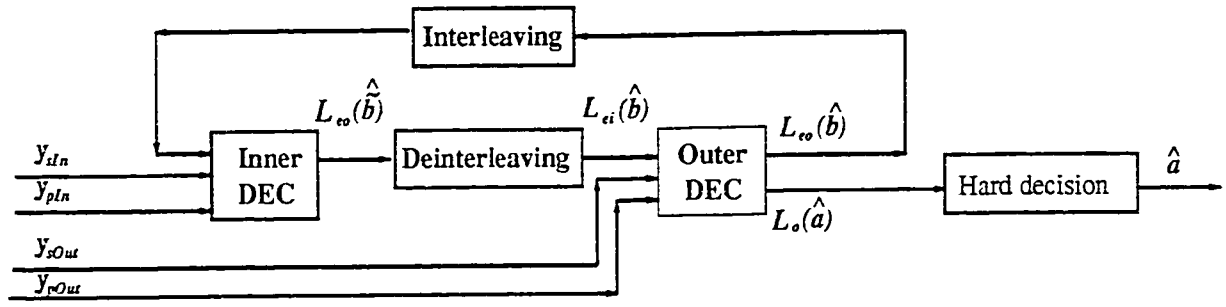

Figure 2.5: Serial iterative decoder

For the final iteration the decision is obtained from the sign of de-interleaved soft-output of DEC2.

2.4.3 Serial Turbo Code

2.4.3.1 Encoder

Figure 2.4 shows a *serial turbo encoder* consisting of systematic outer and inner encoders. The information data is encoded by an outer encoder where the output is a coded sequence. Then the sequence is scrambled by an interleaver and fed to an inner encoder.

2.4.3.2 Decoder

The difference of the *serial turbo decoder* from the parallel turbo decoder is that the $LLRs$ of both information and code symbols are calculated, not only the $LLRs$ of information symbols. Figure 2.5 shows the serial turbo decoders where the L denotes LLR and the subscript e denotes the extrinsic information and the subscripts i and o denote the input to and output from a $SISO$ decoder respectively.

Each $SISO$ decoder can provide extrinsic and soft-output values for both information and code symbols. At the first iteration, the inner received symbols are fed to the inner decoder, i.e., Inner DEC, where the soft-output of the information part of an inner code, $L_{eo}(\widehat{\widehat{b}})$, is obtained and de-interleaved, which is then input to the Outer DEC as a priori value, $L_{ei}(\widehat{b})$ for the code symbols of outer code. The Outer DEC processes the outer received symbols with the a priori value and introduces the extrinsic information of the code symbols of outer code $L_{eo}(\widehat{b})$ which are interleaved and fed back to the Inner DEC as the priori value. At the last iteration the soft-output of the information symbols $L_o(\widehat{a})$ are calculated at the Outer DEC and the sign of the soft values is the final decision. More details on how to obtain soft output is divided into two cases for convolutional and block turbo codes. The trellis-based decoding is used for convolutional turbo codes where more on this matter can be found in [3], [31]. Whereas for block turbo codes, the decoding is separated into two main approaches. One is based on list decoding and the other is based on the trellis-based decoding. Our emphasis is on the latter and we will discuss it in Chapter 4.

2.5 Conclusion

In this chapter, the basic classes of error correcting code, including block and convolutional codes, were described. In block code, both the input and output are of the fixed-size sequence. On the contrary, continuous data is used in convolutional codes. A code is constructed by multiplying the block input or sequence of data with a generator matrix.

Furthermore, the new class of concatenated codes called "Turbo Code" was presented. At the encoder, the two systematic encoders are used with interleaver between them to provide parallel or serial concatenated codes. The corresponding iterative MAP decoding scheme of parallel and serial concatenated codes are also

presented. The iterative decoding scheme used at the decoder is a real novelty because there are two MAP decoders where the extrinsic information obtained from decoding process is exchanged between them to improve the overall performance of the code.

Chapter 3

Reed-Muller Codes

3.1 Introduction

In this chapter, the *Reed-Muller* (RM) codes are discussed. We present their definition, properties, the trellis-based *Maximum Likelihood* (ML) decoding used and its application. Since we will use *trellis-based decoding*, the *minimal trellis construction* of linear block codes is also presented. The chapter is organized as follows:

In the second section, the definition and properties of the Reed-Muller codes are presented. In the third section, we present the definitions related to the trellis diagram of block codes. Then the construction of a trellis diagram of a linear block code using Massey algorithm is discussed. In particular, the construction of trellis diagram of a RM code is presented. The trellis-based Maximum Likelihood (ML) decoding or Viterbi decoding with soft-input is presented in section 4. In addition, the simulation result of RM (8,4) using ML decoding is shown. The performance analysis of the concatenated code with RS (73,57) and RM (8,4) as outer and inner codes, respectively is discussed. The performance comparison of this coding scheme with the concatenated code of the same outer code and the convolutional inner code with rate of $\frac{1}{2}$ and constraint length, $K = 7$ which is usually used in satellite ATM

is shown in Section 5.

3.2 Reed-Muller Codes.

Reed-Muller codes can be defined in terms of boolean functions. In order to define codes of length $n = 2^m$, we need m basis vectors $\overrightarrow{v_1}, \overrightarrow{v_2}, ..., \overrightarrow{v_m}$ which take the values 0 or 1 with 2^m in length. Let $\overrightarrow{V} = (\overrightarrow{v_1}, \overrightarrow{v_2}, ..., \overrightarrow{v_m})$ range over V^m, the set of all binary $m - tuples$ in increasing or decreasing order. Let $\overrightarrow{a} \cdot \overrightarrow{b}$ be the boolean product of vector $\overrightarrow{a}$ and $\overrightarrow{b}$ where $\overrightarrow{a} = (a_1, a_2, ..., a_n)$, $\overrightarrow{b} = (b_1, b_2, ..., b_n)$ and '$\cdot$' is AND operation.

$$\overrightarrow{a} \cdot \overrightarrow{b} = (a_1 \cdot b_1, a_2 \cdot b_2, ..., a_n \cdot b_n) \tag{3.1}$$

For simplicity, $\overrightarrow{a} \cdot \overrightarrow{b}$ is denoted by $\overrightarrow{ab}$. The vector obtained from a boolean product of l vectors is said to be a polynomial of degree l. Boolean function, $f(\overrightarrow{V}) = f(\overrightarrow{v_1}, \overrightarrow{v_2}, ..., \overrightarrow{v_m})$, is defined as any function that takes on values 0 or 1 from the AND operation of its arguments. The following are the definitions and code parameters of RM code.

Definition 1 : Let $0 \leq r \leq m$, the binary Reed-Muller code $\Re(r, m)$ of order r and length 2^m consists of the vectors $\overrightarrow{f}$ associated with all Boolean functions f, that are polynomials of degree less than or equal to r in m variables.

Code parameters :

- The code length, n is 2^m.
- The dimension or message length, k of $\Re(r, m)$ is defined as:

$$k = 1 + \begin{pmatrix} m \\ 1 \end{pmatrix} + \begin{pmatrix} m \\ 2 \end{pmatrix} + ... + \begin{pmatrix} m \\ r \end{pmatrix} \tag{3.2}$$

- The minimum distance, d_{min} of $\Re(r,m)$ is 2^{m-r}.

The generator matrix, G of $\Re(r,m)$ with order r and length 2^m consists of vectors with polynomial degree less than or equal to r and can be constructed as follows:

$$G = \begin{bmatrix} \overrightarrow{1} \\ \hline \overrightarrow{v_m} \\ \overrightarrow{v_{m-1}} \\ \vdots \\ \overrightarrow{v_1} \\ \hline \overrightarrow{v_{m-1}v_m} \\ \overrightarrow{v_{m-2}v_m} \\ \vdots \\ \overrightarrow{v_1v_2} \\ \hline \vdots \\ \hline \overrightarrow{v_{m-r+1}v_{m-r+2}\cdots v_m} \\ \overrightarrow{v_{m-r}v_{m-r+2}\cdots v_m} \\ \vdots \\ \overrightarrow{v_1v_2\cdots v_r} \end{bmatrix} \begin{array}{ll} \text{---}> & \textit{degree } 0 \\ & \\ \text{---}> & \textit{degree } 1 \\ & \\ \text{---}> & \textit{degree } 2 \\ & \\ \vdots & \vdots \\ & \\ \text{---}> & \textit{degree } r \end{array} \qquad (3.3)$$

In an example of $\Re(2,4)$, the 4 basis vectors of length 16 where the 16 columns

represent the 4 binary tuples in increasing order and they are given by

$$\begin{aligned}
\overrightarrow{v_4} &= 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1 \\
\overrightarrow{v_3} &= 0\ 0\ 0\ 0\ 1\ 1\ 1\ 1\ 0\ 0\ 0\ 0\ 1\ 1\ 1\ 1 \\
\overrightarrow{v_2} &= 0\ 0\ 1\ 1\ 0\ 0\ 1\ 1\ 0\ 0\ 1\ 1\ 0\ 0\ 1\ 1 \\
\overrightarrow{v_1} &= 0\ 1\ 0\ 1\ 0\ 1\ 0\ 1\ 0\ 1\ 0\ 1\ 0\ 1\ 0\ 1
\end{aligned} \tag{3.4}$$

A generator matrix for this code is constructed from vectors obtained from all boolean functions of 4 basis vectors with polynomial degree less than or equal to 2.

$$G_{\Re(2,4)} = \begin{bmatrix}
\overrightarrow{v_0} &= 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1 \\
\overrightarrow{v_4} &= 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1 \\
\overrightarrow{v_3} &= 0\ 0\ 0\ 0\ 1\ 1\ 1\ 1\ 0\ 0\ 0\ 0\ 1\ 1\ 1\ 1 \\
\overrightarrow{v_2} &= 0\ 0\ 1\ 1\ 0\ 0\ 1\ 1\ 0\ 0\ 1\ 1\ 0\ 0\ 1\ 1 \\
\overrightarrow{v_1} &= 0\ 1\ 0\ 1\ 0\ 1\ 0\ 1\ 0\ 1\ 0\ 1\ 0\ 1\ 0\ 1 \\
\overrightarrow{v_3 v_4} &= 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 1\ 1\ 1 \\
\overrightarrow{v_2 v_4} &= 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 1\ 0\ 0\ 1\ 1 \\
\overrightarrow{v_1 v_4} &= 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 1\ 0\ 1\ 0\ 1 \\
\overrightarrow{v_2 v_3} &= 0\ 0\ 0\ 0\ 0\ 0\ 1\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 1 \\
\overrightarrow{v_1 v_3} &= 0\ 0\ 0\ 0\ 0\ 1\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 1 \\
\overrightarrow{v_1 v_2} &= 0\ 0\ 0\ 1\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 1
\end{bmatrix} \tag{3.5}$$

The dimension of the code, k equals 11 because the constraint of the code; the degree of polynomial of all boolean functions have to be less than or equal to 2.

$$k = 11 = 1 + \binom{4}{1} + \binom{4}{2} \tag{3.6}$$

The code which is generated by the generator matrix constructed above is not in a systematic form. However, the RM code is a linear block code, so the generator

matrix can be modified by using linear operations on its rows to make it a systematic-like code. We will describe this in the next section.

For simplicity, as it is common to describe a code with parameters n and k, we will use the notation RM(n, k) instead of $\Re(r, m)$ in this thesis.

3.3 Minimal Trellis for Linear Block Codes

In [46], the trellis diagram construction and the Maximum-Likelihood (ML) decoding of block codes were presented. One obvious factor that determines the complexity of a trellis-based decoder for a block code is the structure of its trellis (the number of states and branches). However, it has been found that there are many trellis representations for a given block code. Thus, one way to reduce the complexity of the decoder is to seek the "*Minimal Trellis*" which means the best-trellis representation in the sense of having the smallest number of states and branches than any other trellis-representations. Recently, there has been a lot of attention on the trellis structure of block codes [47]-[50]. As it is stated in [47], different trellis representations are obtained from different orderings (permutatings) of symbol positions of any given block code. Up to now, the problem of finding minimal trellis of a block code attained by any permutation has not been solved in general and has been stated to be an NP-complete problem [52]. However, there are some codes whose minimal trellis is known. These include the RM code [53] and Goley code [47].

In this section, we first introduce some basic notation and definition related to trellis representation of a linear block code [55]. Next we will discuss the minimal trellis construction using Massey method [54], [55]. We use Massey algorithm because it constructs the trellis diagram of a *systematic* linear block code.

3.3.1 Notation and Definition

Some preliminary notations and definitions used to explain trellis of block code are defined as follows:

Definition 1: A n-depth trellis, $T = (S, B, L)$, is a directed graph of length n. It consists of three sets of the elements as follows:

The states, S, the branches, B and the labels, L. Where, each set can be decomposed into subsets as given below:

$$\begin{aligned} S &= S_0 \cup S_1 \cup \cdots \cup S_n \\ B &= B_0 \cup B_1 \cup \cdots \cup B_{n-1} \\ L &= L_0 \cup L_1 \cup \cdots \cup L_{n-1} \end{aligned} \tag{3.7}$$

whereas at time i, a subset S_i consists m_i states. In S_o and S_n subsets, each has only one state called original state s_o and final state $\dot{s}_f$, respectively. At section i, each branch b_{ij} in B_i connects a state in S_i to a state in S_{i+1} with a branch-label l_{ij} in L_i.

Definition 2: A trellis of depth n, $T = (S, B, L)$ represents a linear block code C of length n, if the sequence of branch labels at each path is uniquely corresponding to a code word in C.

Definition 3: A trellis T for a code C is called minimal, if the number of states at each time $i \ : \ i = 0, 1, ..., n$ is minimal among all possible trellis representations of C.

Definition 4: Let $\overrightarrow{a} = (a_1, a_2, ..., a_n)$ be a non zero vector over $GF(q)$. $L(\overrightarrow{a})$ is called the left index of $\overrightarrow{a}$ which equals to the smallest index i such that $a_i \neq 0$.

Definition 5: Let $G = (\overrightarrow{g}_0, \overrightarrow{g}_1, ..., \overrightarrow{g}_{k-1})^T$ be a $k \times n$ matrix with k row vectors of length n, $\overrightarrow{g}_i \ : \ i = 0, 1, ..., k-1$ over $GF(q)$. G is said to be in a reduced echelon

form if,

$$L(\overrightarrow{g}_0) < L(\overrightarrow{g}_1) < \cdots < L(\overrightarrow{g}_{k-1}) \tag{3.8}$$

and k columns of G at positions $L(\overrightarrow{g}_i)$: $i = 0, 1, ..., k-1$ have weight one.

3.3.2 Minimal Trellis Construction of Linear Block Codes.

In this part, we first describe the Massey algorithm used to construct minimal trellis of linear block codes. Then, the minimal trellis construction of systematic-like RM code will be discussed.

3.3.2.1 Massey Construction

A code C is constructed from a $(k \times n)$ generator matrix G of reduced echelon form. Let γ_i : $i = 0, 1, ..., k-1$ be left indexes of matrix G. As it is stated in the properties of a reduced echelon matrix, that k columns at the left indexes have weight one so it implies that in a code word, the information bits can be found at positions of the left indexes. In trellis $T = (S, B, L)$ of the block code, C over $GF(q)$ is constructed by specifying the set of states S_i at time i : $i = 0, 1, ..., n$. The states in S_i are identified by the knowledge of information symbols already observed at time i, thus, all other information symbols are assumed to be zero. Let p be the largest index such that $\gamma_p \leq i$. States in S_i are labeled by

$$s_j = \{(c_{i+1}, ..., c_n) \ : \ (c_1, ..., c_n) = (m_1, ..., m_p, 0, ..., 0)G\} \tag{3.9}$$

The original and final states are $S_o = \{0\}$ and $S_n = \{\phi\}$ by tradition, where ϕ is an empty string.

The branches in B_i of T are defined as follows :

- When $i = \gamma_p$, there is a branch $b \in B_i$ connects a state, $s \in S_{i-1}$ with a state, $s' \in S_i$, if and only if there exist code words $c = (c_1, c_2, ..., c_n)$ and $c' = (c'_1, c'_2, ..., c'_n)$ in C such that,

$$\begin{aligned} (c_i, c_{i+1}, ..., c_n) &= s \\ (c'_{i+1}, ...c'_n) &= s' \end{aligned} \tag{3.10}$$

where it is either $c' = c$, or $\beta(c' - c)$ equals the p^{th} row of G with $\beta \in GF(q)$. The branch label is c'_i and the number of out-going branches at each state in S_{i-1} are q.

- When $i > \gamma_p$, there is a branch $b \in B_i$ connects a state $s \in S_{i-1}$ with a state $s' \in S_i$ if and only if there exists a code word $(c_1, c_2, ..., c_n) \in C$, such that,

$$\begin{aligned} (c_i, c_{i+1}, ..., c_n) &= s \\ (c_{i+1}, ..., c_n) &= s' \end{aligned} \tag{3.11}$$

The branch label is c_i . In this case there is only one out-going branch from each state in S_{i-1}.

3.3.2.2 Trellis Diagram of a RM Code.

A generator matrix of RM code is constructed as defined in section 2 and modified to be in a row-reduced echelon form. After that, the Massey algorithm is applied on the matrix to construct the minimal trellis of the code. As stated in the previous part that the positions of information bits can be indicated in a code word even though they are not at first or last k positions of a code word as in a systematic code. Thus, we can consider the code as a systematic-like RM code. The following is a generator matrix, G of a RM (8,4) code and a trellis diagram of the code drawn by Massey algorithm is shown in Figure 3.1. The example of a code word $C = (1, 1, 0, 0, 0, 0, 1, 1)$ is represented by a sequence of branch labels of the path

with dashed lines as shown in the same figure.

$$G = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix} \tag{3.12}$$

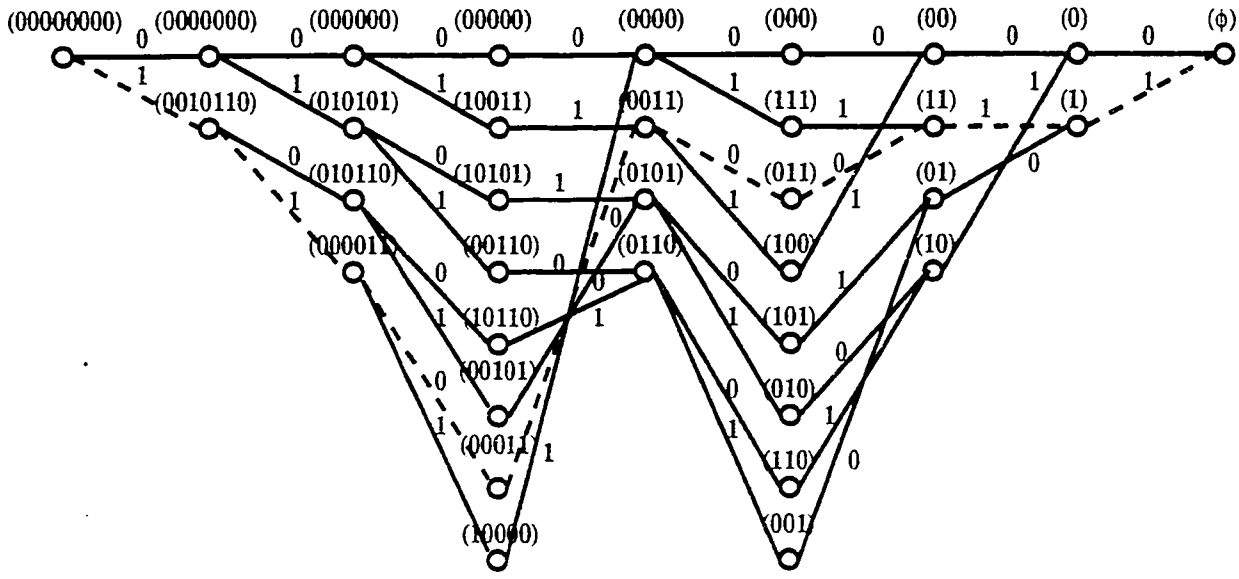

Figure 3.1: Trellis diagram of the RM (8,4)

3.4 Maximum Likelihood Decoding.

We assume that *Binary Phase Shift Keying* (BPSK) modulation scheme is used. A code word, $C = (c_0, c_1, ..., c_n)$ where each bit takes value 0 or 1 is mapped to transmitted code word $Y = (y_0, y_1, ..., y_n)$, where $y_i : 0 < i < n$ takes value +1 and -1. In a Gaussian channel, the received sequence, $R = (r_0, r_1, ..., r_n)$ obtained from a detector is given by

$$r_i = y_i + n_i \quad , 0 < i < n \tag{3.13}$$

where n_i is the white Gaussian noise random variable with zero mean and $\frac{N_o}{2}$ variance. In this decoding scheme, the likelihood function is the probability density function of the received bit r_j conditioned on the transmitted y_i that has been sent. It is given as follows:

$$p(r_i \mid y_i) = \frac{1}{\sqrt{\pi N_o}} e^{\frac{-(r_i - y_i)^2}{N_o}} \quad , \; 0 < i < n \tag{3.14}$$

In a memoryless channel, the likelihood function of the received sequence $\overrightarrow{r}$ and the transmitted code word, $\overrightarrow{y}$ is given by

$$p(\overrightarrow{r} \mid \overrightarrow{y}) = \Pi_{i=0}^{n-1} p(r_i \mid y_i) \tag{3.15}$$

For calculation simplicity that the addition is used instead of multiplication, we will consider log-likelihood function instead of likelihood function. Thus, the log-likelihood function is given by

$$\begin{aligned} log\,(p(\overrightarrow{r} \mid \overrightarrow{y})) &= \sum_{i=0}^{n-1} log\,(p(r_i \mid y_i)) \\ &= \sum_{i=0}^{n-1} \left[\frac{-(r_i - y_i)^2}{N_o} - log\left(\sqrt{\pi N_o}\right) \right] \\ &= \sum_{i=0}^{n-1} \left[\frac{-r_i^2 + 2r_i y_i - y_i^2}{N_o} - log\left(\sqrt{\pi N_o}\right) \right] \\ &= \alpha_1 \sum_{i=0}^{n-1} r_i y_i + \alpha_2 \end{aligned} \tag{3.16}$$

where α_1 and α_2 are constants. In order to perform trellis-based maximum likelihood decoding, the maximum of log-likelihood function over all paths in trellis is sought. The constants can be omitted because a relatively maximal value is required, thus we shall present the branch metric, $m_i(r \mid y)$ and path metric, $M(\overrightarrow{r} \mid \overrightarrow{y})$ which

are a reduced form of log-likelihood functions as follows:

$$m_i(r \mid y) = r_i y_i \quad , 0 < i < n-1 \tag{3.17}$$

$$M(\overrightarrow{r} \mid \overrightarrow{y}) = \sum_{i=0}^{n-1} m_i \tag{3.18}$$

The Viterbi algorithm [44], [45] is a trellis-based maximum likelihood decoding which can perform hard or soft- input decoding. However, the soft-input decoding scheme known as soft decision decoding has advantage over the hard-input one because it uses the real channel information or the multi-level quantization on the data which gives more information on how strong the data is. Whereas in hard-input decoding, the input data is quantized into two levels. The following is a detailed procedure of Viterbi algorithm.

3.4.1 Viterbi Algorithm.

Let N_{ij} be a node at state i and level j, $V(N_{ij})$ be a node value of node N_{ij}. The node value of each node in trellis is computed as follows:

1. Set $V(N_{oo}) = 0$ and $j = 1$.

2. Compute the partial path metrics obtained from summing the node value at time $j-1$ and branch metric of branch connects the node at time $j-1$ to time j of all paths entering each node at time j.

3. Set $V(N_{ij})$ of each node at level j equals to the maximum partial path metric entering the node. The partial path with maximum metric is called survivor path, whereas others partial paths are deleted from the trellis.

4. If $j \leq n$, increase j by one and return to step 2, otherwise the the maximum

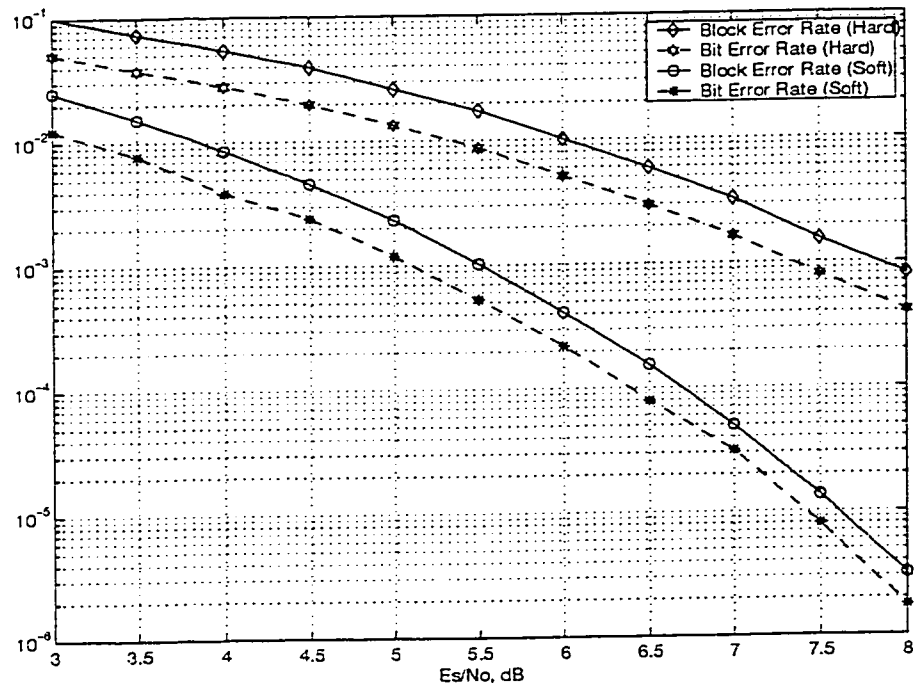

Figure 3.2: Comparison of performance of the RM code with soft and hard decision decoding.

likelihood code word is obtained from the sequence of branch labels of the survivor path.

3.4.2 Simulation Results of a RM (8,4) using ML Decoding

The Viterbi algorithm is applied to trellis diagram of RM(8,4) drawn in section 3.3. Figure 3.2 shows the simulation result in terms of block error rate and bit error rate of RM (8,4) code using soft-input and hard-input decoding at different $\frac{E_b}{N_o}$. It shows that soft decision decoding has a better performance than that of hard decision decoding with a coding gain of about 2.3 dB at block error rate and bit error rate of 10^{-3}.

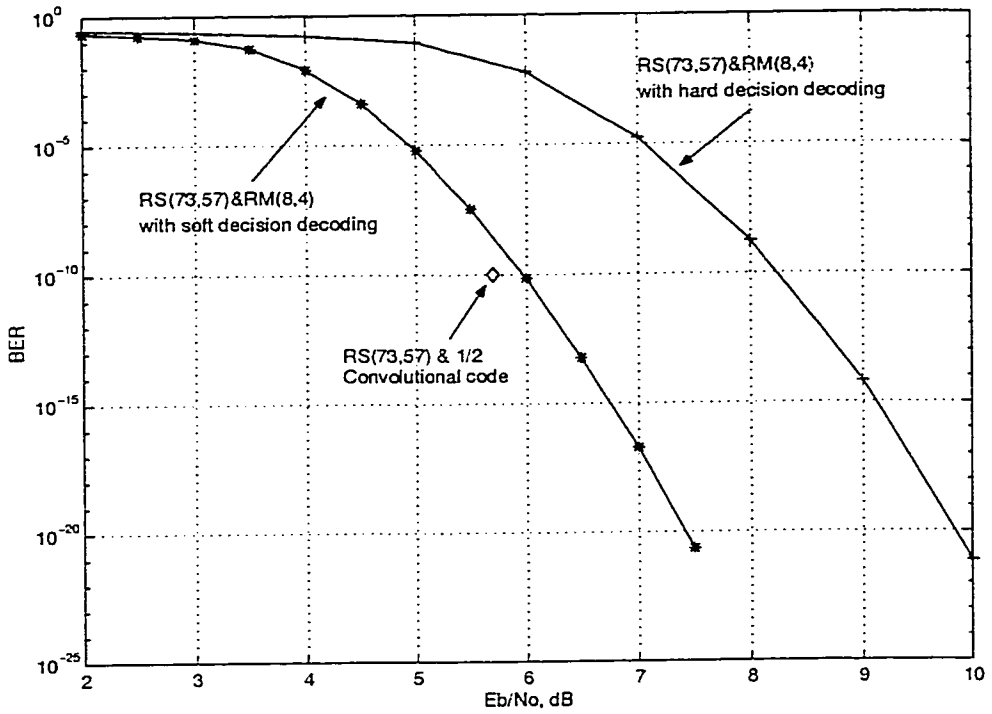

Figure 3.3: Performance of a concatenated code using RS(73,57) and RM(8,4) codes with soft and hard decision decoding.

3.5 Performance of a Reed-Solomon/Reed-Muller Concatenated Code.

In this concatenation scheme we use RM (8,4) as an inner code and RS(73,57) as an outer code. The RS (73,57) is shortened from RS(255,239) and its symbols are taken from $GF(256)$ so that one symbol is represented by 8 bits or 1 byte. Thus, one symbol of outer code will be encoded by two RM (8,4) code. By using block error rate, P_{RM} , of soft decision decoding presented in the previous section, the probability of symbol error of RS code, P_e is given by

$$P_e = 1 - (1 - P_{RM})^2 \tag{3.19}$$

The bit error probability of RS (73,57) with 8-error correcting capability is given

below:

$$P_b \leq \frac{128}{255} \sum_{i=9}^{73} \left[\frac{8+i}{73} \right] \begin{pmatrix} 73 \\ i \end{pmatrix} P_e^i (1-P_e)^{73-i} \tag{3.20}$$

A good reference on the performance analysis of linear block code can be found in [56] where we also followed the method presented in [56]. Figure 3.3 shows the analytical BER versus $\frac{E_b}{N_o}$ of the RS(73,57) and RM(8,4) concatenated code both with soft decision and hard decision decoding of the RM code. We can see that in order to achieve BER of 10^{-10} , $\frac{E_b}{N_o}$ of 6 and 8.2 dB are required for the RM code with soft decision decoding and with hard decision decoding respectively. It is clearly shown that the combination of RS(73,57) and RM(8,4) with soft decision decoding is giving a better deal of BER than that with hard decision decoding with moderate cost of complexity. It is also shown the analytical performance of RS (73,57) and rate $\frac{1}{2}$ convolutional code at BER of 10^{-10} [57] which has just slightly better performance, 0.3 dB gain, than that of RS(73,57) and RM (8,4) with soft decision decoding, but has much higher complexity. The trellis representation of the convolutional code rate of $\frac{1}{2}$ with constraint length 7 has 64 states at each time index, whereas trellis diagram of RM(8,4) shown in section 3.3 has maximum states of 8. And these two coding scheme use the same Viterbi algorithm, we can say that the decoding complexity of the RM (8,4) inner code is 8 times less than that of the convolutional inner code rate of $\frac{1}{2}$.

3.6 Conclusion

In this chapter, first the definitions and the code construction of RM code were explained. Then, the minimal trellis construction using Massey algorithm was presented. The minimal trellis is the best trellis representation of a block code in terms of the lowest number of states at each time index. Following this the Maximum

Likelihood (ML) algorithm of block was discussed. Along this chapter, the example of the construction of RM(8,4) code and its trellis representation were given. As well as the simulation result of RM(8,4) code using soft decision ML decoding, which is better than that of the hard decision ML decoding was shown. Finally, the performance analysis of concatenated code using RS(73,57) and RM(8,4) as outer code and inner code respectively was calculated and comparing this coding scheme with the existing scheme which is the concatenated RS(73,57) and convolutional code rate $\frac{1}{2}$. It obtains 8 times less complexity with penalty of 0.3 dB performance degradation.

Chapter 4

Reed-Muller Turbo Code

4.1 Introduction

In a block turbo code, the concept of iterative decoding as in Berrou's work is applied to two-dimensional block codes [31], [34]. There are two approaches used in optimal and sub-optimal MAP decoding of block codes. In sub-optimal MAP decoding, the algorithm based on the list decoding e.g. the Chase algorithm is used [34]. The optimal MAP algorithm uses the trellis-based MAP decoding for linear block codes is proposed by Hagenauer et al. [31]. We follow the same line as Hagenauer et. al. because of its optimality. In addition, the minimal trellis of the selected code, the RM code, is known so that we can obtain optimal performance with low complexity.

In this chapter, the turbo encoder and decoder will be discussed. In turbo encoder, the parallel concatenated code constructed from two elementary encoders with interleaver between them is presented. In turbo decoder, first, the *trellis-based MAP decoding*, followed by *iterative MAP decoding*. Then, the system model used for the simulation purpose is given. Finally, the simulation results of RM turbo codes on Additive White Gaussian Noise (AWGN) and Rayleigh-fading channels are shown.

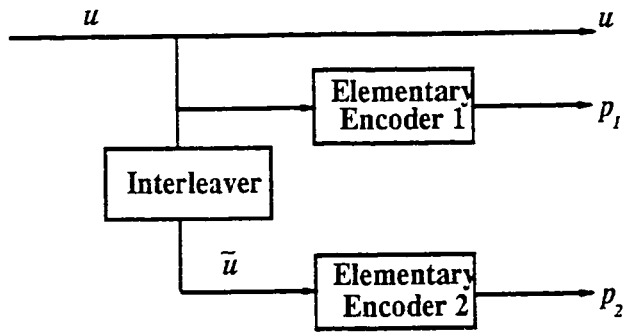

Figure 4.1: RM-turbo encoder

4.2 RM Turbo Encoder

In this thesis, we consider a parallel two-dimensional block turbo encoder shown in Figure 4.1. The block of information u and $\tilde{u}$, the permuted version of u with *block interleaver*, are encoded by two elementary RM encoders. In block interleaver, data is written in row wise from left to right and top to bottom and read out column wise from top to bottom and left to right. This turbo code, shown in Figure 4.2., consists of k_2 (n_1, k_1) and k_1 (n_2, k_2) linear systematic block codes. Where n_i and k_i are code length and information length of code C_i : $i = 1, 2$. This code can be considered as *product code* [44] without parity on parity even though a product code is serial concatenated code, whereas this code is parallel codes. Therefore, the decoding of a complex and long code can be broken up into the decoding steps of shorter codes. The extrinsic information L_e^- and $L_e^|$ produced from horizontal and vertical decodings are also shown in Figure 4.2.

The information part, $u = k_1 \times k_2$ bits, is encoded horizontally using the elementary encoder 1 which consists of k_1 (n_2, k_2) RM encoders. These RM encoders produce parity block, p^-. Then, u is encoded vertically which it can be thought of as the interleaved version of horizontal one resulted from block interleaver. By using k_2 (n_1, k_1) RM encoders, a vertical parity block , $p^|$ is generated.

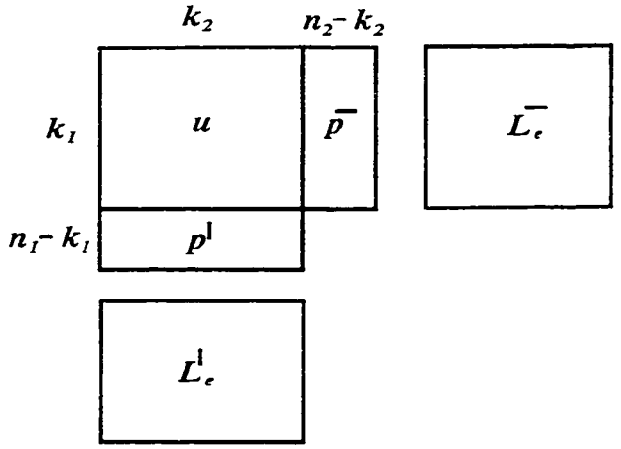

Figure 4.2: Two-dimensional block code

We consider the two-dimensional RM code, which use the same RM (n, k) code in each dimension denoted as RM $(n, k)^2$ code. The overall code rate is given below:

$$R = \frac{k^2}{n^2 - (n-k)^2} \tag{4.1}$$

Figure 4.2. demonstrates the idea of a two-dimensional codes in the case of systematic component codes. The generalization of non-systematic case is straightforward. In this thesis, we use codes in which the information part, although distinct, is not placed at the beginning of the code word. In this case, some modification on horizontal codes has been made by reordering code bits to obtain the sequence whose the first k positions are information bits. Along similar lines, the vertical codes are also reordered. It is noted that the reordering sequences are not code words, thus the inverse operation has to be done before the decoding process starts. Figure 4.2 shows the reordered sequences which are obtained from the systematic-like code.

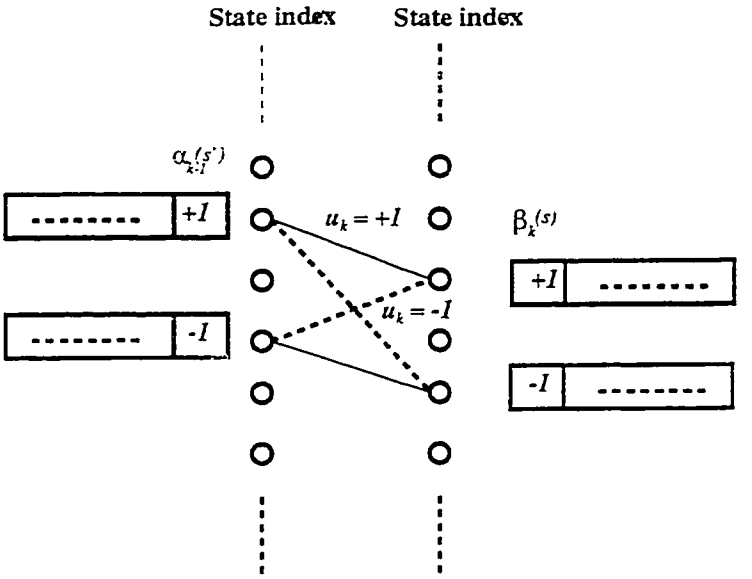

Figure 4.3: Trellis structure of a systematic block code

4.3 Turbo Decoder

Turbo decoder is referred to as the iterative decoding scheme. The *Maximum A Posteriori* (MAP) decoder is used in this iterative decoding because it accepts soft-input and provides soft-output for information bits. In this section, the trellis-based MAP algorithm and iterative decoding for two-dimensional block code are discussed in depth by following the Hagenour's paper [31].

4.3.1 Trellis-Based MAP Algorithm for Linear Block Codes

The binary trellis of block codes is shown in Figure 4.3. The coded bit at time m, x_m is the label of branch which connects from time $m-1$ to time m. The s' and s are the trellis states at time $m-1$ and at time m. The soft-output of a given information bit, x_m from the MAP decoder is defined as the a posteriori log-likelihood ratio for "$x_m = +1$" and "$x_m = -1$" which are transmitted when sequence $\overrightarrow{y}$ is received.

The soft-output is given by

$$L(\widehat{x}_m) = L(x_m \mid \overrightarrow{y}) = log\frac{P(x_m = +1 \mid \overrightarrow{y})}{P(x_m = -1 \mid \overrightarrow{y})} = log\frac{\sum_{(s',s),x_m=+1} p(s', s, \overrightarrow{y})}{\sum_{(s',s),x_m=-1} p(s', s, \overrightarrow{y})} \quad (4.2)$$

In the memoryless channel, the joint probability $p(s', s, \overrightarrow{y})$ can be written as follows:

$$\begin{aligned} p(s', s, \overrightarrow{y}) &= p(s', \overrightarrow{y}_{j<m}) \cdot p(s, \overrightarrow{y}_m \mid s') \cdot p(\overrightarrow{y}_{j>m} \mid s) \\ &= p(s', \overrightarrow{y}_{j<m}) \cdot P(s \mid s') \cdot p(\overrightarrow{y}_m \mid s', s) \cdot p(\overrightarrow{y}_{j>m} \mid s) \\ &= \alpha_{m-1}(s') \cdot \gamma_m(s', s) \cdot \beta_m(s) \end{aligned} \quad (4.3)$$

where $\overrightarrow{y}_{j<m}$ represented the portion of received sequence from bit 0 up to bit $m-1$. Similarly, the received sequence from bit m up to bit $n-1$ is denoted by $\overrightarrow{y}_{j>m}$. And $\alpha_m(s)$ and $\beta_{m-1}(s')$ are the forward and backward recursions of the MAP decoding respectively.

$$\alpha_m(s) = \sum_{s'} \gamma_m(s', s) \cdot \alpha_{m-1}(s') \quad (4.4)$$

$$\beta_{m-1}(s') = \sum_{s} \gamma_m(s', s) \cdot \beta_m(s) \quad (4.5)$$

where $\alpha_0(0) = 1$ and $\beta_n(0) = 1$. The branch transition probability is provided by

$$\gamma_m(s', s) = P(s|s') \cdot p(y_m|s', s) = p(x_m; y_m) \quad (4.6)$$

We assume that the information bits are statistically independent. In (n, k) systematic block codes, the transition probability is given by

$$p\left(x_m; y_m\right) = \begin{cases} p(y_m|x_m) \cdot p(x_m) & 1 \leq m \leq k \\ p(y_m|x_m) & k+1 \leq m \leq n \end{cases} \tag{4.7}$$

A priori probability $p(x_m)$ and the condition probability $p(y_m \mid x_m)$ are given below:

$$p(x_m = \pm 1) = \frac{e^{\pm L(x_m)}}{1 + e^{\pm L(x_m)}} = \left(\frac{e^{-L(x_m)/2}}{1 + e^{-L(x_m)}}\right) \cdot e^{L(x_m) \cdot x_m/2} = A_m \cdot e^{L(x_m) \cdot x_m/2} \tag{4.8}$$

$$p(y_m \mid x_m = \pm 1) = \left(\frac{P(x_m) \cdot (1 + e^{-L(x_m)}) \cdot e^{-L_c \cdot y_m/2}}{1 + e^{-(L(x_m) + L_c \cdot y_m)}}\right) \cdot e^{L_c \cdot y_m \cdot x_m/2} = B_m \cdot e^{L_c \cdot y_m \cdot x_m/2} \tag{4.9}$$

where the detailed derivation can be found in Appendix A. And the log-likelihood ratio associated with $p(x_m; y_m)$ can be written as

$$L\left(x_m; y_m\right) = \begin{cases} L_c y_m + L(x_m) & 1 \leq m \leq k \\ L_c y_m & k+1 \leq m \leq n \end{cases} \tag{4.10}$$

A_m and B_m in equations 4.11 and 4.12 are equal for all transition from time $m-1$ to time m and can be omitted due to the ratio in equation 4.5. So, the reduced version of branch transition probability can be expressed as follows:

$$\gamma_m(s', s) = e^{\left(\frac{1}{2} x_m (L_c \cdot y_m + L(x_m))\right)} = e^{(L(x_m; y_m) \cdot x_m/2)} \tag{4.11}$$

In systematic block code, a priori probability $L(x_m)$ equal zero if x_m is a parity bit.

4.3.2 Soft-Output Calculation

Soft-output from log-MAP and Max-log-MAP decoding are given in this section.

- The optimal soft-output using log-MAP decoder can be written as

$$L(\widehat{x}_m) = L_c y_m + L(x_m) + log\frac{\sum_{(s',s),x_m=+1} \alpha_{m-1}(s') \cdot \beta_m(s)}{\sum_{(s',s),x_m=-1} \alpha_{m-1}(s') \cdot \beta_m(s)} \tag{4.12}$$

  where forward and backward are given in equations 4.7 and 4.8. The last expression is the extrinsic information

- The sub-optimum soft-output using Max log-MAP for systematic block codes is the approximated version of log-MAP algorithm. Since, the $log(e^{a_1} + e^{a_2} + \cdots + e^{a_m}) \approx max_{0 \leq i \leq m} a_i$ so that the third term in equation is approximated in terms of maximum and given by

$$\begin{aligned} L_{log-MAP}(\widehat{x}_m) &= L_c \cdot y_m + L(x_m) + max_{(s',s),x_m=+1} \left(log(\alpha_{m-1}(s')) + log(\beta_m(s))\right) \\ &\quad - max_{(s',s),x_m=-1} \left(log(\alpha_{m-1}(s')) + log(\beta_m(s))\right) \end{aligned} \tag{4.13}$$

  with

$$log(\alpha_m(s)) = max_{s'} \left(log(\alpha_{m-1}(s')) + \frac{1}{2} L(x_m; y_m) \cdot x_m \right) \tag{4.14}$$

$$log(\beta_{m-1}(s')) = max_s \left(log(\beta_m(s)) + \frac{1}{2} L(x_m; y_m) \cdot x_m \right) \tag{4.15}$$

  The soft-output can also be calculated from the modified trellis and the trellis of dual code. For further detail, it is referred to [31].

4.3.3 Iterative Decoding of a Two-Dimensional Code

Figure 4.4 shows the iterative decoding procedure of the code shown in Figure 4.2 for K iterations.

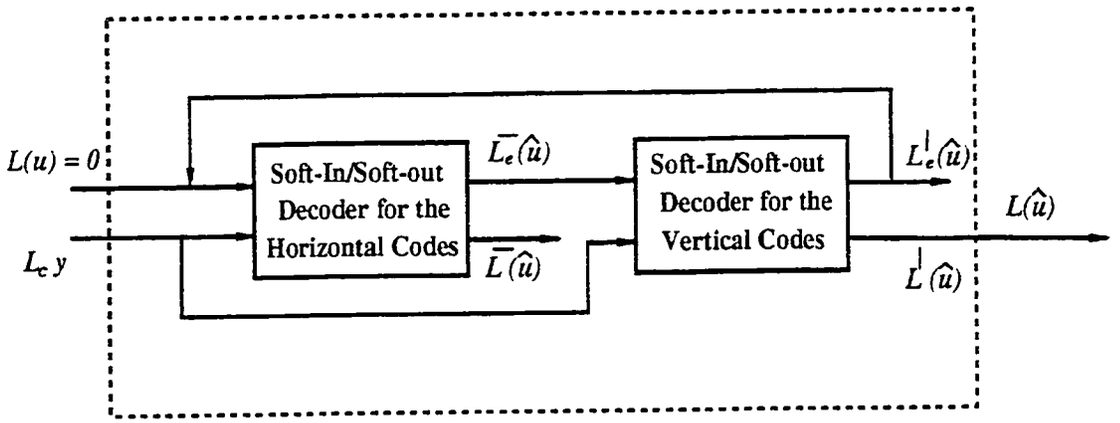

Figure 4.4: Iterative decoding procedure of two-dimensional block code

1. Set the a priori value $L(u) = 0$ because the equally likely probability of input data is assumed. And set the number of iteration $I = 0$.

2. Decode the information u horizontally and obtain the horizontal extrinsic information for information bits as follows:

$$L_e^{-}(\widehat{u}) = L(\widehat{u}) - L_c \cdot y - L(u) \tag{4.16}$$

3. Set $L(u) = L_e^{-}(\widehat{u})$. Where the extrinsic information from the horizontal decoder is passed to the vertical one as a priori value of information bits.

4. Decode the information u vertically and obtain the vertical extrinsic information for information bits as follows:

$$L_e^{|}(\widehat{u}) = L(\widehat{u}) - L_c \cdot y - L(u) \tag{4.17}$$

5. Set $L(u) = L_e^{|}(\widehat{u})$. The reason is similar as in step 3.

6. If $I < K$, set $I = I + 1$ otherwise go to step 7.

7. The soft output is:

$$L(\widehat{u}) = L_c \cdot y + L_e^{-}(\widehat{u}) + L_e^{1}(\widehat{u}) \tag{4.18}$$

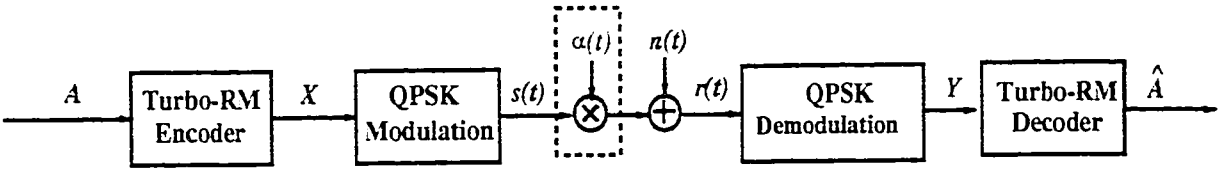

Figure 4.5: System model

4.4 System Model.

Figure 4.5 shows the channel model used for simulation purposes. The received signal can be written as

$$r(t) = \alpha(t)s_i(t) + n(t)$$

where $\alpha(t)$ is a Rayleigh process that at any specific time satisfies $E(\alpha^2) = 1$ where α is a Rayleigh random variable with probability density function,

$$p_\alpha(\alpha) = 2\alpha e^{-\alpha^2} \quad , \; a \geq 0$$

$n(t)$ is white Gaussian noise process with two-sided power spectral density $N_o/2$. We assume QPSK modulation is used. $s_i(t) \;\; : i = 1, 2, 3, 4$ is the modulated waveform for the symbol s_i . The channels are modeled as follows:

AWGN channel : $\alpha(t) = 1$

Fading channel: $\alpha(t)$ is Rayleigh process

4.5 Simulation Results

Figure 4.6, 4.7 and 4.8 show the BER versus E_b/N_o of RM $(8,4)^2$, RM $(16,11)^2$, RM $(32,26)^2$-turbo codes at different number of iterations. The coding gain obtained by increasing the number of iterations is high at the first incremental step and it becomes lower with successive steps. Moreover, the coding gains are saturated at 2^{nd} , 4^{th} and 5^{th} iteration in RM $(8,4)^2$, RM $(16,11)^2$, RM $(32,26)^2$-turbo codes with neglect performance degradation. The reason for the faster saturation of the performance in the element code with shorter code length is that a code provides the smaller size of the interleaver which implied the less diversity of information which is the heart of iterative decoding. It is noted that from now, the number of iterations used in the simulations of RM $(8,4)^2$, RM $(16,11)^2$ and RM $(32,26)^2$-turbo codes are two, four and five. The BER versus E_b/N_o curves of RM-turbo codes with different lengths in an AWGN channel are shown in Figure 4.9. The coding gains of 3.6. 5.3 and 6.2 dB for RM $(8,4)^2$, $(16,11)^2$, $(32,26)^2$ turbo codes are obtained over uncoded QPSK at BER of 10^{-5}. The results are better than the ones reported in [31] as expected since we are using stronger component codes. In addition, the performance of RS(73,57) and convolutional code rate of $\frac{1}{2}$ concatenated code at BER of 10^{-6} is given and compared with RM $(32,26)^2$ turbo code where the turbo coding scheme obtains coding gain of about 0.5 dB with higher code rate of about 1.7 times than that of the concatenated code which is equivalent to an additional coding gain of about 2.4 dB .

In parallel to the results for AWGN channel, Figure 4.8 depicts the BER versus $E_b/N_o$of RM-turbo codes in Rayleigh fading channel. We obtain at least 25 dB coding gain over uncoded QPSK for RM $(8,4)^2$, $(16,11)^2$, $(32,26)^2$- turbo codes at BER of 10^{-4}. It is illustrated in Figure 4.9 that the higher coding gain is obtained for more iterations and saturated after 5 iterations in the case of RM $(32,26)^2$-turbo

code.

4.6 Conclusion

In this chapter, the RM turbo encoder was described which could be considered as the encoding process of product code without parity on parity. The information was encoded both horizontally and vertically, where the vertical version of information could be considered as the permuted version of information with block interleaver. In the decoding process, the basic concept related to a posteriori probability used in trellis-based MAP decoding and iterative MAP decoding was discussed. The simulation results on AWGN and Rayleigh-fading channels were presented. It was shown that the turbo decoding improves the performance when the number of iterations increases, although it saturates after a few iterations. The number of iterations needed depends on the interleaver size. The longer the interleaver the more gain obtained from increasing the number of iterations. The saturation is a result of the extrinsic information exchanged between two decoders being highly correlated so that no extra information could be provided after a few iterations.

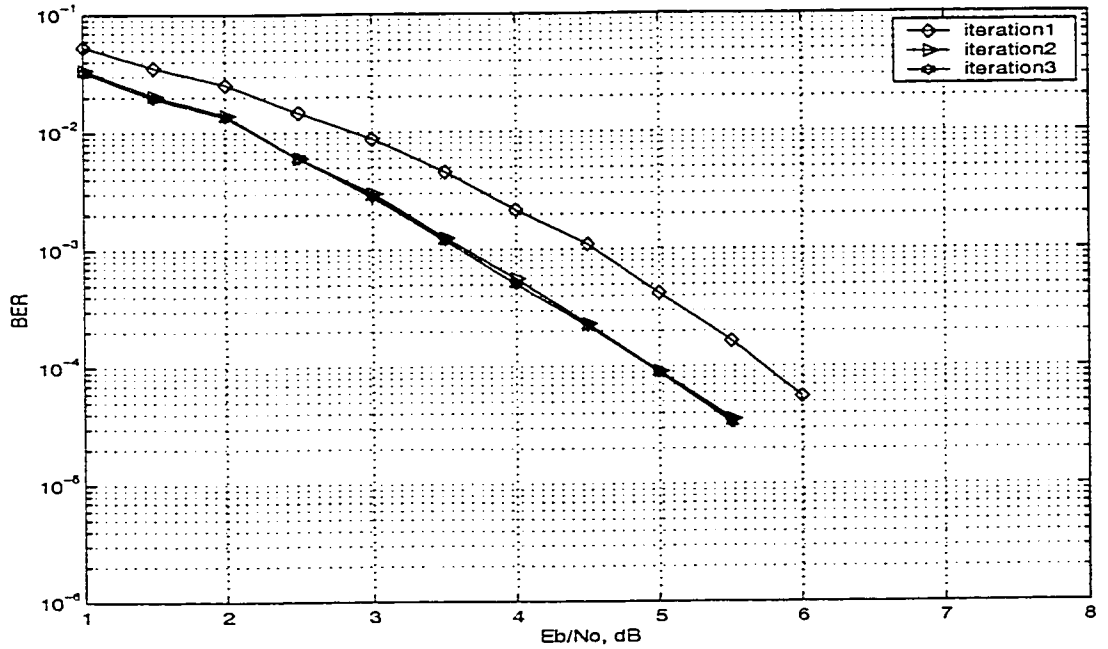

Figure 4.6: The performance of a RM $(8,4)^2$- turbo code with different iterations on an AWGN channel

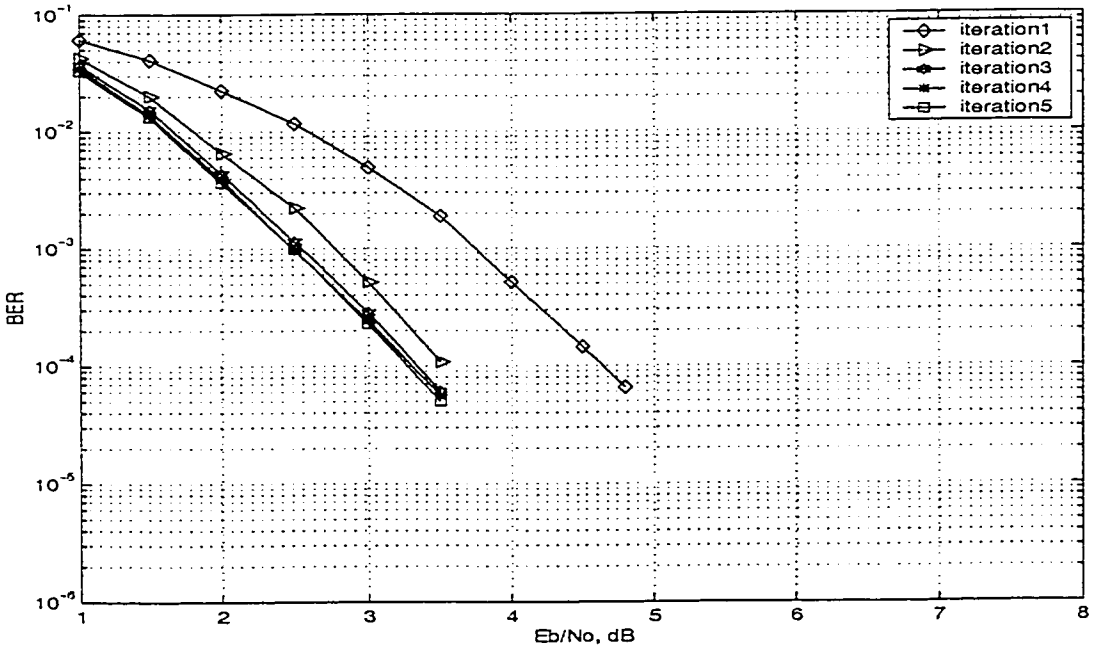

Figure 4.7: Performance of a RM $(16,11)^2$- turbo code with different iterations on an AWGN channel

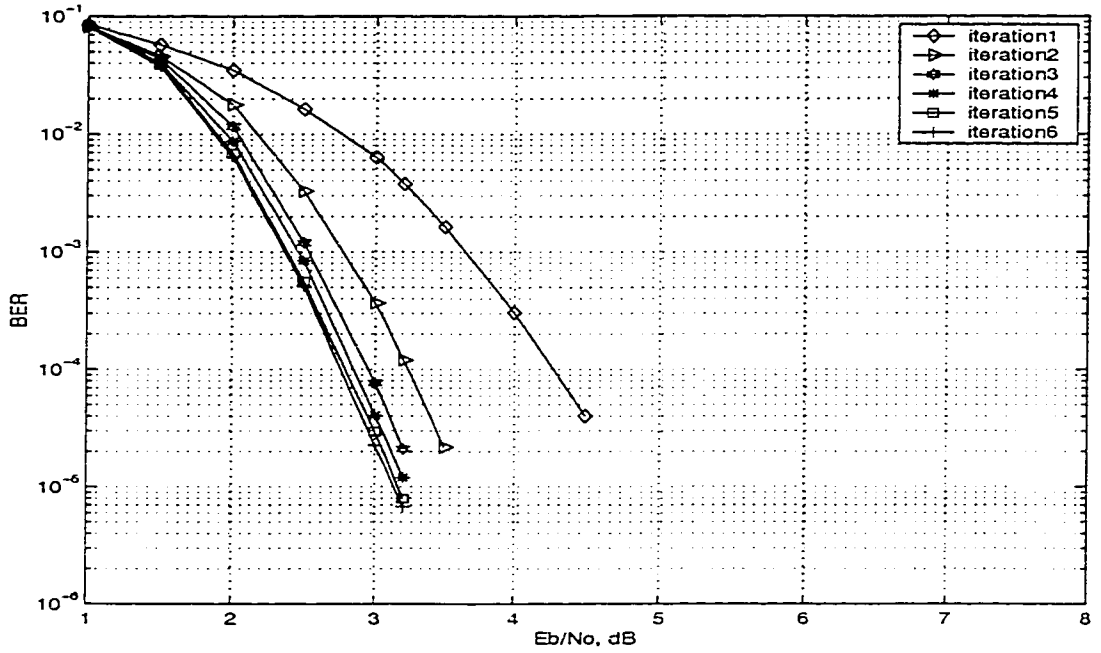

Figure 4.8: Performance of a RM $(32, 26)^2$- turbo code with different iterations on an AWGN channel

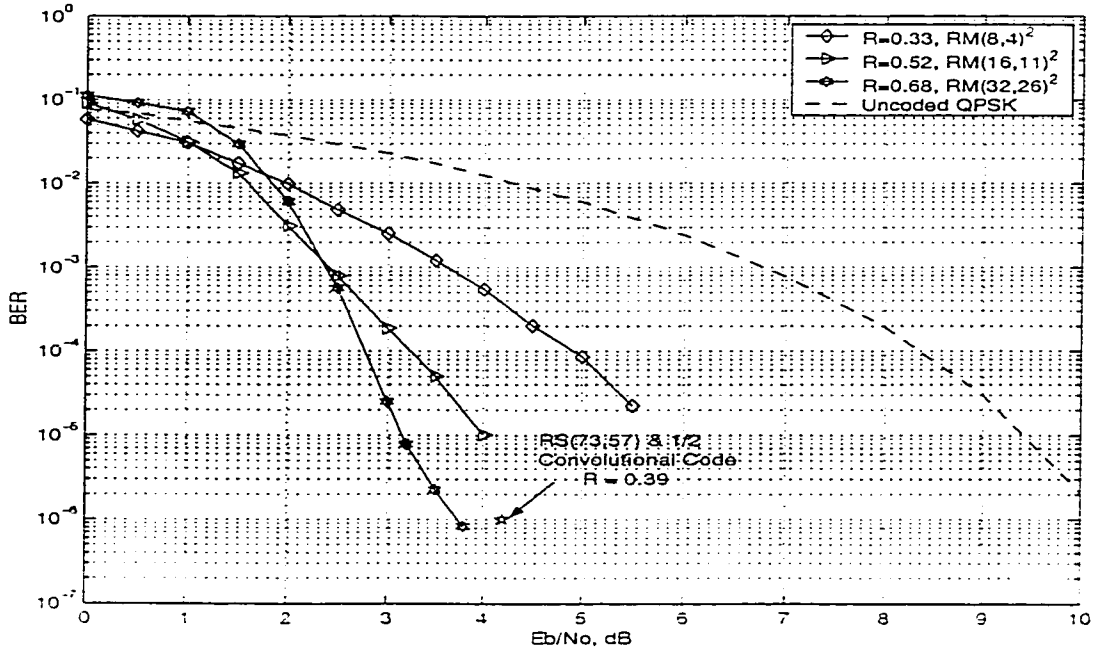

Figure 4.9: Performance of RM-turbo codes with different code lengths after 5 iterations on an AWGN channel

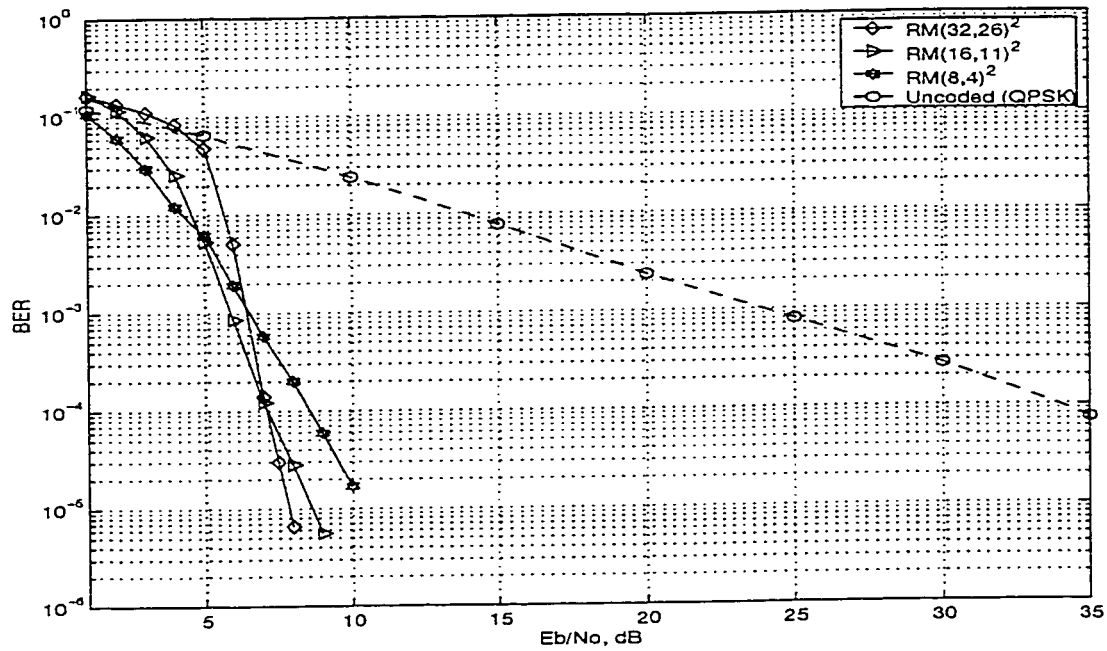

Figure 4.10: Performance of a RM-turbo code with different code lengths after 5 iterations on a Rayleigh channel

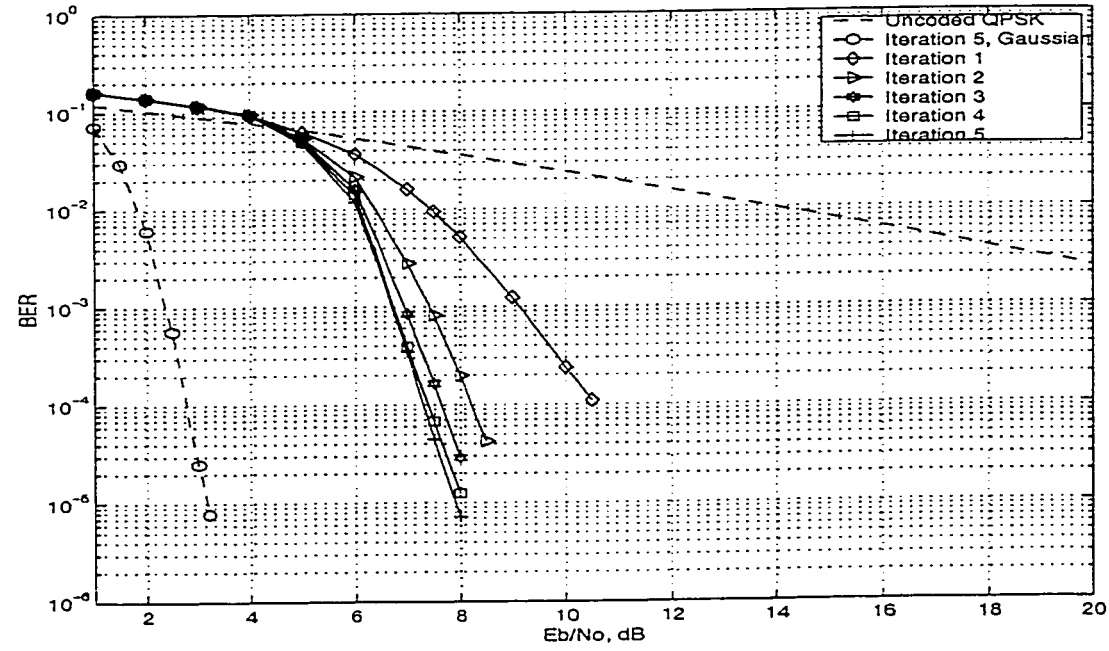

Figure 4.11: Performance of a RM $(32, 26)^2$- turbo code with different iterations on a Rayleigh channel

Chapter 5

Turbo Codes for Cell-Based Transmission

5.1 Introduction

On account of their excellent performance, turbo codes are proposed to be used in many applications in which the power is limited. One application, also in our interest, is the *cell-based transmission* such as Asynchronous Transfer (ATM) used in DVB-RCS. In this thesis, we attempt to provide an alternative for the existing system which is a Satellite ATM used in Return Channel which currently uses a concatenated code with a shortened RS (73,57) code and a convolutional code rate of $\frac{1}{2}$ as outer and inner codes, respectively. This RS code is shortened from RS(255,239) to fit in a 57-byte satellite ATM cell. The shortened version of the code has some advantages; first we can match the longer code in some specific smaller size code, second we can use the same encoder and decoder for different shortened codes which derived from the same mother code.

This chapter is organized as follows. Primary, a brief review of ATM is presented. Then, the design of shortened RM-turbo codes with different shortening patterns

will be discussed. Also the performance of the shortened version of the proposed coding scheme is investigated. It is shown that some shortened patterns obtain *Unequal Error Protection* (UEP) property, whereas a UEP code is more suitable for the structure of ATM cell since cell-header is more important than its payload. However, as our attempt is solely to express our new idea to obtain UEP codes by shortening a two-dimensional code, the possibility to improve and/or to control the performance at different regions of the codes can be done in future.

5.2 A Brief Review of an ATM network.

In this section, we briefly explain an Asynchronous Transfer Mode (ATM) in general and some specific issues related to our work. An ATM network is defined as a cell-relay network, a connection-oriented network and a hardware switch [58].

On a cell-relay network, data in small fixed-size packets, or cells, is transmitted. It has several advantages over long and variable-length packets or frames network such as Ethernet. In particular, ATM requires less storage space and it can be processed faster. It is also more suitable for interactive applications. However, the size of the cell must be small enough to reduce latency but large enough to minimize overhead.

In fact that an ATM network is a connection-oriented network, means the connection from source to destination must be established before transmitting data. The delay at connection time is a disadvantage of connection-oriented network, whereas its guarantee to the required amount of bandwidth, a certain Quality of Service (QoS) and a better network traffic management are its advantages.

On an ATM network in the sense of hardware switch, all devices, such as workstations, servers, routers and bridges, are attached to a switch. When one ATM switch sends a connection request to a destination switch, the ATM-attached devices set

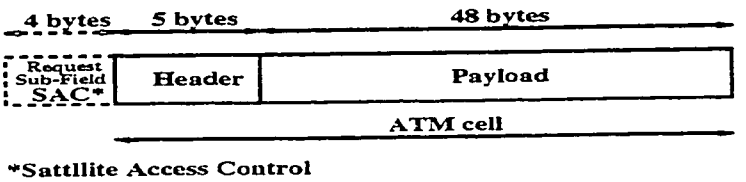

Figure 5.1: Satellite ATM cell

up the connection. The best path is determined by switches while the connection is being set. Thus, we can say that an ATM network covers the areas of switching device, the transfered protocol between two devices, the communication architecture which is a structured set of protocols that performs the communication tasks and finally the interface which is a protocol providing a set of functions for accessing a network. ATM network has both advantages of *circuit switching*, including guaranteed capacity and constant transmission delay and *packet switching*, including flexibility and efficiency. Therefore, it can send various types of traffic such as audio, video and data over the same switches.

5.2.1 ATM Cell Format

The basic format of an ATM cell is shown in Figure 5.1. The ATM cell consists of 53 bytes which is divided into two parts. The first 5-byte part is the cell-header, and the 48-byte part is the payload or user information. For satellite application, an ATM Traffic cell contains extra header part which is the request sub-field of Satellite Access Control (SAC) [59].

5.2.2 ATM Cell-Header Format

There are two cell-header formats in an ATM network : User Network Interfaces (UNI) or Network Node Interfaces (NNI). The UNI header is used for communication between ATM endpoints and ATM switches in private ATM networks. The NNI header is used to communicate between ATM switches. The UNI, NNI cell-header

formats and SAC-subfield are shown in Figure 5.2. The fields in an ordinary ATM cell-header are briefly described as follows:

- Generic Flow Control (GFC)-Provides local functions, such as identifying multiple stations that share a single ATM interface. This field is typically not used and is set to its default value.
- Virtual Path Identifier (VPI)-Constitutes a routing field for the network
- Virtual Channel Identifier (VCI)-Constitutes a routing to and from the end user.
- Payload Type (PT)-Indicates the type of information in the information field.
- Congestion Loss Priority (CLP)-Provide guidance to the network in the event of congestion.
- Header Error Control (HEC)-Calculates checksum only on the header itself.

In satellite applications, the header may be extended to contain extra routing and management information peculiar to satellitte systems. The extension of the ATM header may be done either by adding extra bytes to the existing 5 header bytes or by mapping the 5-byte header to longer than 5-byte header. In the former case, the prefix method [59] of the satellite ATM, a cell is optionally prefixed by a 4-byte SAC request sub-field. The additional prefix obtains control and management information, mainly for capacity requests. Thus, in the specific case when the prefix is used, we can consider that the satellite ATM cell has the 9-byte header.

5.3 Shortened RM Turbo Codes for Satellite ATM

As shown in Figure 5.2, the ATM cell-header contains the connection information and, therefore, is more important than the cell payload. To fit an ATM cell in the

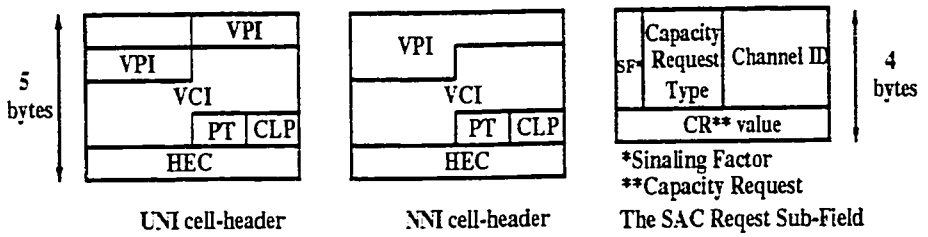

Figure 5.2: ATM UNI and NNI cell-header formats and the SAC request sub-field

$(k \times k)$ information section of a two-dimensional block code we need to shorten the code. In this section, we present the shortened RM turbo code with different shortening patterns. Moreover, the idea of shortening the two-dimensional code to give UEP property for different regions of the code word in order to make it more suitable for ATM applications is discussed. The design of shortening patterns and simulation results for different regions is presented.

5.3.1 Shortening Patterns for RM turbo Codes.

The ordinary shortening pattern of a given code is to set the shortened message bits to be zero at the end of the code. However, it is not mandated to be, so in this part we propose different patterns for shortening the RM-turbo codes.

In our study, we consider ATM cells for satellite applications. These cells have 48-byte payload and 9-byte header. Thus, we use two dimensional code with RM (32,26) component codes. Each information section of these codes is $26^2 = 676$ bits long which exceeds the length of a satellite ATM cell, i.e., $57 \times 8 = 456$ bits by 220 bits. That is, we have to shorten the RM-turbo codes by setting the 220 information bits to be zero. These zeros will not be transmitted, hence, the overall rate of the code, R_c will be roughly as follows:

$$R_c = \frac{456}{32^2 - 6^2 - 220} \cong 0.594 \tag{5.1}$$

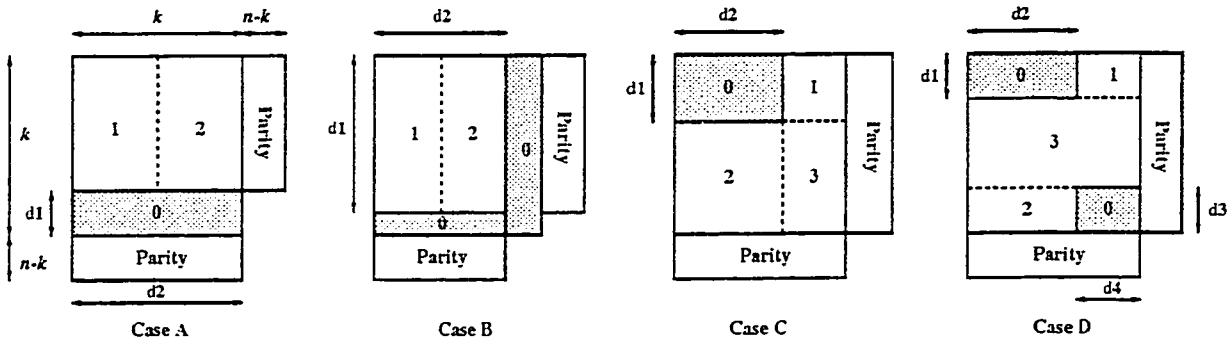

Figure 5.3: Shortening patterns

Of course, the exact rate depends on the particular shortening scheme. Figure 5.3 shows four shortening patterns. The following are the different shortening patterns with their corresponding rates :

$$\text{Case A}: \quad \text{d1} = 8\ ,\ \text{d2} = 26,\ R_c = 0.64$$

$$\text{Case B}: \quad \text{d1} = 24,\ \text{d2} = 19,\ R_c = 0.64$$

$$\text{Case C}: \quad \text{d1} = 11\ ,\ \text{d2} = 18\ ,\ R_c = 0.61$$

$$\text{Case D}: \quad \text{d1} = 8\ ,\ \text{d2} = 17 \text{ and } \text{d3} = 8\ ,\ \text{d4} = 9\ ,\ R_c = 0.60$$

where d1, d2, d3 and d4 are the dimensions of shortening block shown in Figure 5.3.

In case A and B, we attempt to design the shortening patterns in such a way that we can reduce the number of parity bits to obtain higher code rate. However, in case C and D, we design the shortening pattern to make the codes having special property which is Unequal Error Protection (UEP) property by surrounding the highly protected part with more zeros than others as shown in region 1 of case C and D.

5.4 Simulation Results

Figure 5.4 and 5.5 show the performance at different regions in the shortened codes. The shortening pattern in case A is an ordinary shortening pattern, whereas case

B, C and D are modified shortening patterns. In case A and B, the performance of different regions which shows Equal Error Protection (EEP) property of the codes is almost the same . In contrast, the performance of case C and D at different regions is different depending on the number of zeros (shortened bits) surrounding the region. The lower BER in the regions that have more zeros around them is obtained. The UEP property is obviously seen in case C and D where the region 1 gets the best performance followed by region 2 and 3 respectively. Figure 5.6 compares the BER of two different regions in case C with the performance of case B. The results show that the BER of the best region in UEP code is lower than that of EEP code. Figure 5.7 shows the overall performance of the EEP, UEP and original (non-shortened) codes, where the EEP codes provide coding gain about 0.2 dB over UEP codes. Nevertheless, the performance of the EEP codes and UEP codes are about 0.2 and 0.4 dB worse than that of the original RM-turbo codes. This is because the shortening process affects the distance spectrum of the two-dimensional codes.

The performance comparison among different coding schemes for ATM transmission is illustrated in Figure 5.8. The performance of the proposed coding scheme case A is compared with the existing scheme, RS(73,57) and convolutional rate of $\frac{1}{2}$ concatenated code [57]. It is shown that the coding gain of about 0.2 dB is obtained by the proposed coding scheme over the concatenated code at BER of 10^{-6} and the code rate of the proposed codes are higher than that of the concatenated code by about 1.6 times which is equivalent to an additional coding gain of about 2 dB. Furthermore, the lower bound of BER of the double-binary Circular Recursive Systematic Convolutional (CRSC) turbo code at 8 iterations with 4-bit quantization is given. It is important to note that this lower bound of BER is calculated from the block error rate presented in [26]. The CRSC code obtains coding gain of about 0.8 dB over the RM-turbo code case A. Finally, the performance of the shortened

version of the extended Hamming $(32, 26)^2$ code at BER of 10^{-5} is given. This coding scheme performs worse than the proposed coding scheme by about 0.3 dB with a lower code rate or it is equivalent to overall degradation of about 0.5 dB.

5.5 Conclusion

In this chapter, we presented the concept of an ATM network, especially for satellite application. Then, the shortened RM turbo code with four different shortening patterns were discussed. Two of shortening patterns were designed to reduce the number of parity bits of the codes resulting in higher rate codes. The others were designed to obtain special property, i.e. the UEP property. In an UEP code, the information portion of the two-dimensional block code was divided into a few regions, each having a different level of error protection. These codes are suitable for connection-oriented networks such as ATM where a cell-header contains information about the connection path and status of a cell which is more important than its payload.

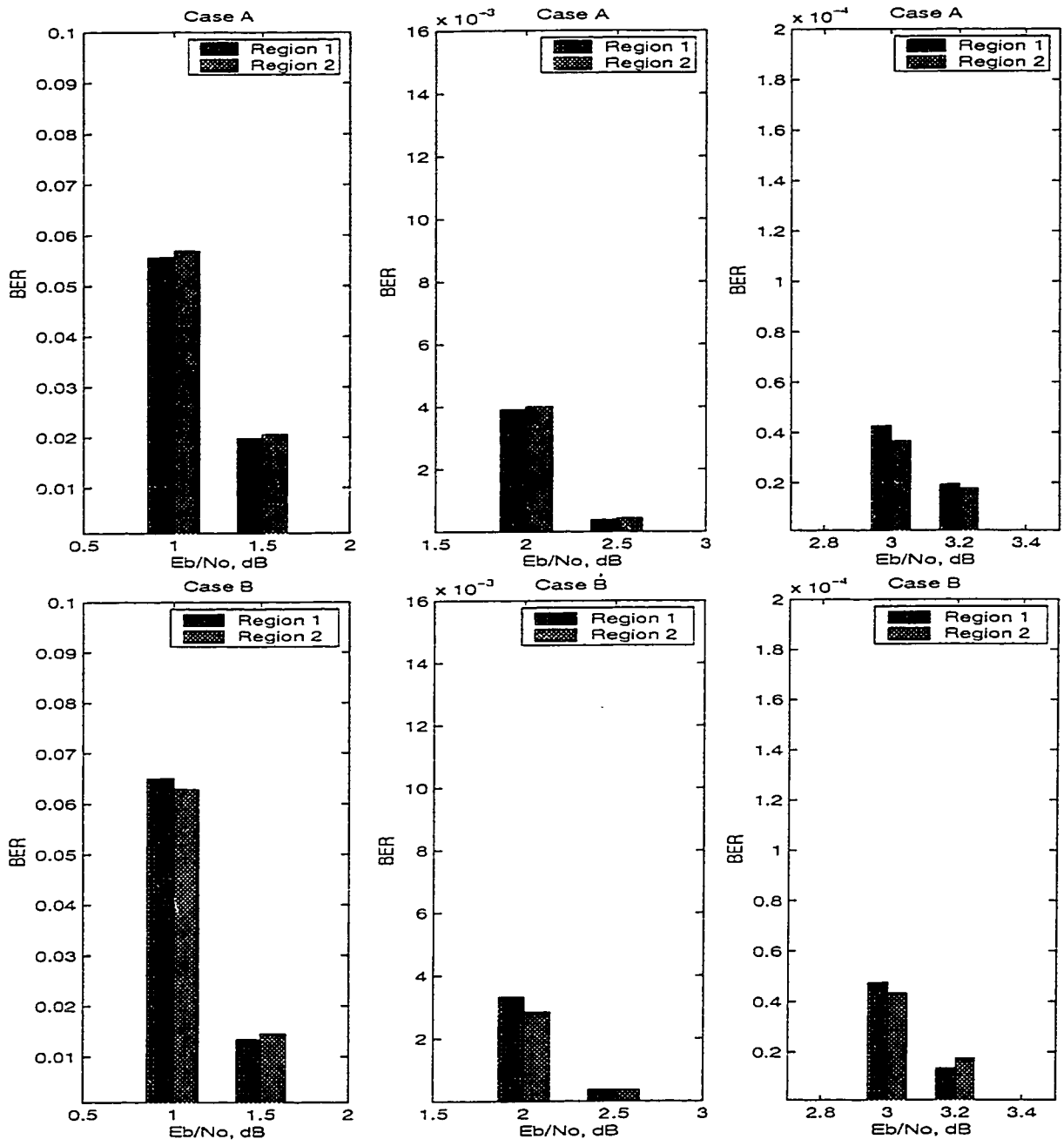

Figure 5.4: Performance of shortening patterns A and B at different regions.

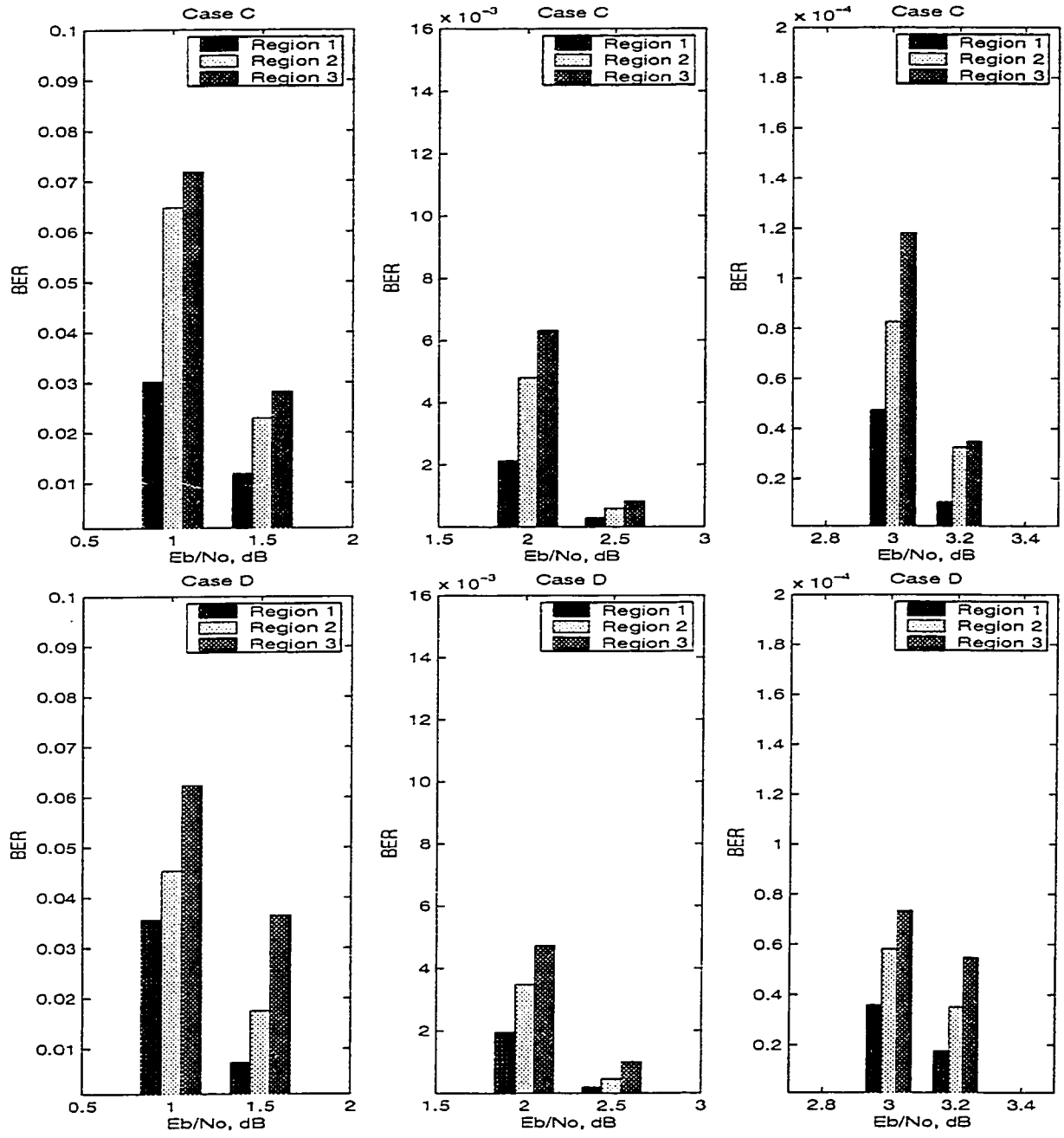

Figure 5.5: Performance of shortening patterns C and D at different regions.

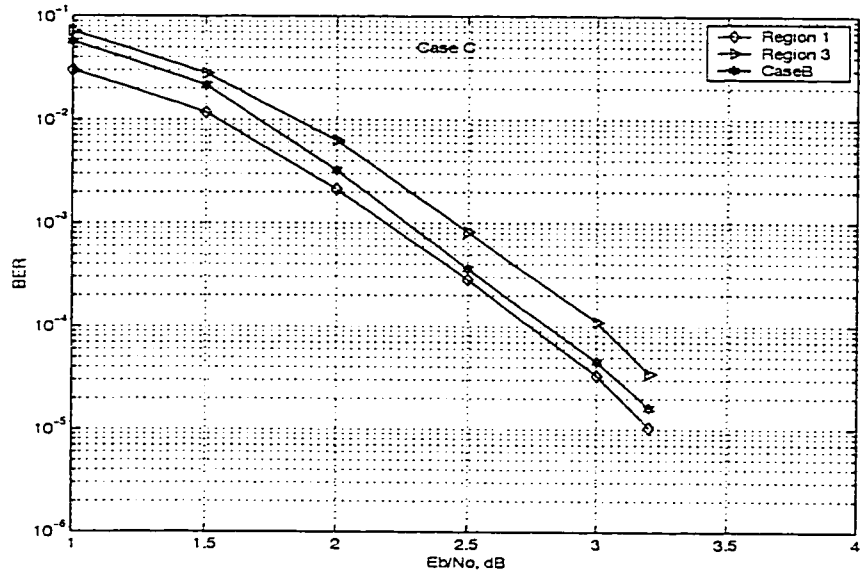

Figure 5.6: Performance of a shortening pattern B at Region 1 and 3

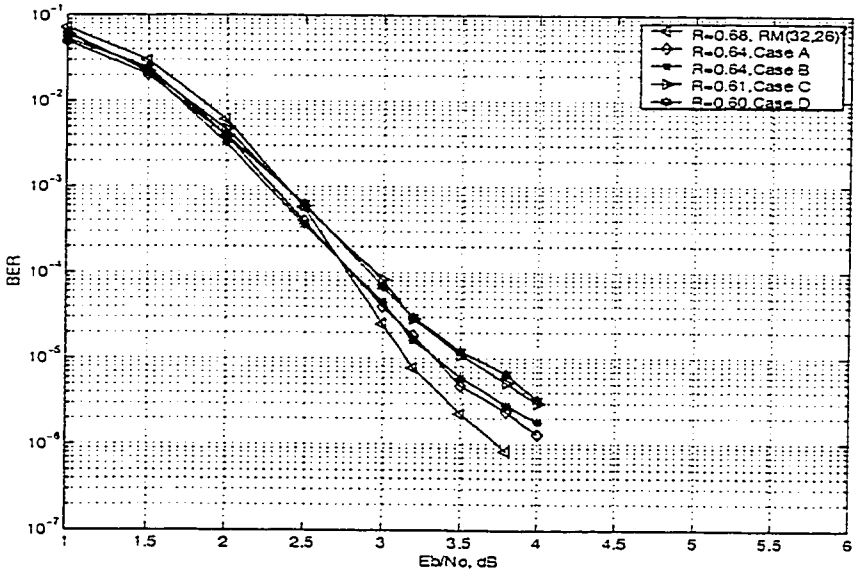

Figure 5.7: Overall performance of shortened RM-turbo codes with different shortening patterns

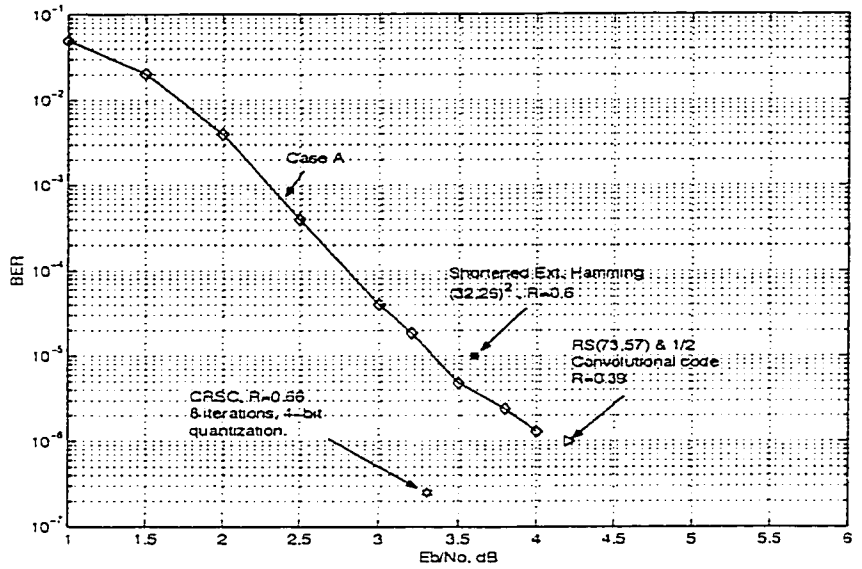

Figure 5.8: Performance comparison of different coding schemes for ATM transmission

Chapter 6

Effect of Channel Impairments

6.1 Introduction

So far, we have assumed the perfect carrier phase and channel SNR estimation in our simulations. However, this does not truly represent a practical system when the *channel impairments* caused by noise and attenuation occur. Some examples of the channel impairments are carrier phase offset and channel SNR mismatch, i.e., difference between the assumed and actual SNR.

In this chapter, we investigate the effect of channel SNR mismatch on the RM-turbo code performance. Additionally, the effect of phase offset on the shortened RM-turbo codes is investigated. In order to recover carrier phase, we send a preamble in place of the shortened bits and use this preamble to estimate the carrier phase. We also investigate the impact of preamble size on the performance of the codes.

6.2 System Model for the Investigation of Channel Impairments

The channel model used to investigate the channel impairments is illustrated in Figure 6.1. The received signal can be written as

$$r(t) = \alpha(t)e^{j\triangle\phi}s_i(t) + n(t) \tag{6.1}$$

where $\alpha(t)$ is a Rayleigh process that satisfies $E(\alpha_i^2) = 1$, $\triangle\phi$ is a random phase offset uniformly distributed over $[-a, a]$, $(\triangle\phi \sim U\ [-a, a])$ where a depends on variance of $\triangle\phi$, $n(t)$ is a Gaussian noise process with two-sided power spectral density $N_o/2$. We assume that QPSK modulation is used and $s_i(t)\ : i = 1, 2, 3, 4$ is the modulated waveform for the symbol s_i .

For the case of an Additive White Gaussian Noise (AWGN) channel we consider two cases :

AWGN channel : without phase offset ; $\alpha(t) = 1$ and $\triangle\phi = 0$

with phase offset ; $\alpha(t) = 1$ and $\triangle\phi \sim U\ [-a, a]$

For the case of Rayleigh-fading channel, we assume that the slow fading is applied so at the receiver, the phase can be recovered using standard techniques and the coherent receiver can be used.

Fading channel: $\alpha(t)$ is Rayleigh process ; $\triangle\phi = 0$

According to the above condition, in the case of Rayleigh fading channel, we do not address the carrier recovery issue.

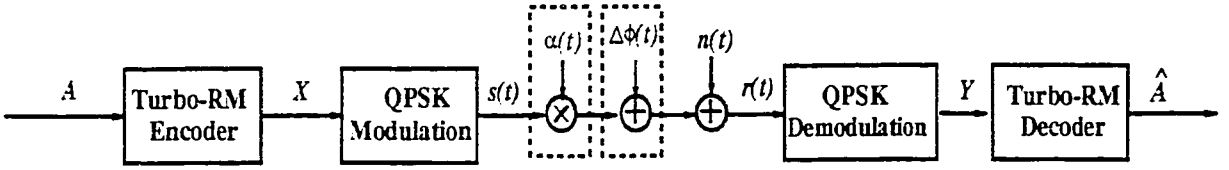

Figure 6.1: System model used to investigate the channel impairments

6.3 Channel SNR Mismatch

Knowledge of the channel SNR plays an important role in the iterative MAP decoding through the soft-output calculation. Thus, the incorrect estimation of the channel SNR will affect the performance of the turbo codes.

The log-likelihood ratio of bit x conditioned on the corresponding received bit y, $L(x \mid y)$ at the detector is given by

$$\begin{aligned} L(x \mid y) &= log\frac{P(x=+1 \mid y)}{P(x=-1 \mid y)} \\ &= log\left(\frac{p(y \mid x=+1)}{p(y \mid x=-1)} \cdot \frac{P(x=+1)}{P(x=-1)}\right) \\ &= log\frac{e^{\left(-\frac{E_s}{N_o}(y-a)^2\right)}}{e^{\left(-\frac{E_s}{N_o}(y+a)^2\right)}} + log\frac{P(x=+1)}{P(x=-1)} \\ &= L_c \cdot y + L(x) \end{aligned} \tag{6.2}$$

where $L_c = 4 \cdot a \cdot \frac{E_s}{N_o}$ is called the reliability value of the channel, a is the fading attenuation. For a Gaussian channel a equals 1 and $L(x)$ is the *a priori* value. The channel SNR mismatch affects the value of L_c used in iterative decoding, for instance, the estimated $\frac{E_b}{N_o}$, i.e., $\widetilde{\frac{E_b}{N_o}} = \frac{1}{R_c} \cdot \widetilde{\frac{E_s}{N_o}}$, where R_c is the code rate, is underestimated by d dB. L_c at the receiver is given by

$$\begin{aligned} L_c &= 4 \cdot a \cdot R_c \cdot \widetilde{\frac{E_b}{N_o}} \\ &= 4 \cdot a \cdot R_c \cdot \left(\frac{E_b}{N_o} / 10^{\frac{d}{10}}\right) \end{aligned} \tag{6.3}$$

We investigate the effect of channel SNR mismatch in terms of $\frac{E_b}{N_o}$ since the performance evaluation is usually done in terms of $\frac{E_b}{N_o}$. However, in the case of channel SNR estimation, the $\frac{E_s}{N_o}$ is directly considered.

It is obvious that in the above formulae, the turbo decoding which uses iterative MAP decoding requires the knowledge of channel SNR or $\frac{E_s}{N_o}$. Where the $\frac{E_s}{N_o}$ is calculated as

$$\frac{E_s}{N_o} = \frac{channel\ signal\ variance}{noise\ variance} \tag{6.4}$$

Some estimation methods of the channel SNR are studied and presented in [60], [61], where the first one uses the polynomial approximation of channel SNR obtained from the mean and variance of the received bits and the second one obtains the channel SNR from the variation of extrinsic information at each iteration. In [62], the hard decision from turbo decoder and received sequences together are used to estimate the noise variance. In this thesis, we use the channel SNR estimation algorithm as in [62] to calculate noise variance, since it is simple and provides good approximation. The noise variance is given as follows:

$$\begin{aligned} \widehat{\sigma}^2_{j+1} &= E\left[(Y_j - \overline{y_j})^2\right] \\ &= E\left[Y_j^2 - 2\overline{y_j}Y_j + \overline{y_j}^2\right] \\ &= E\left[Y_j^2\right] - 2\overline{y_j}^2 + \overline{y_j}^2 \\ &= E\left[Y_j^2\right] - \overline{y_j}^2 \end{aligned} \tag{6.5}$$

where $E\left[Y_j^2\right]$ and $\overline{y_j}$ are given as:

$$E\left[Y_j^2\right] = \frac{1}{N}\sum_{k=0}^{N-1}(r_{k,j})^2 \tag{6.6}$$

$$\overline{y_j} = E[Y_j] = \frac{1}{N}\sum_{k=0}^{N-1} \widehat{d}_{k,j} \cdot r_{k,j} \tag{6.7}$$

where $r_{k,j}$ is the received bit at time k in block j, $\widehat{d}_{k,j}$ is the hard decision output of the turbo decoder after the last iteration. The averaged variance in data block $j+1$ is obtained from weighting the estimated variance in block $j+1$ itself and in previous block j. It is given by

$$\widetilde{\sigma}^2_{j+1} = \widehat{\sigma}^2_{j+1}\beta + (1-\beta)\widetilde{\sigma}^2_j \tag{6.8}$$

where $\beta = 0.95$.

6.3.1 Simulation Results

Figure 6.2 shows the BER versus channel SNR mismatch for RM $(8,4)^2$ - turbo code at different $\frac{E_b}{N_o}$. It is shown that the performance in terms of BER is subtly degraded at the channel SNR mismatch of -5 to -6 dB, otherwise there is no degradation observed. The BER versus channel SNR mismatch for RM $(16,11)^2$-turbo code at different $\frac{E_b}{N_o}$ is presented in Figure 6.3, where the degradation on the performance is observed when the channel SNR mismatch is less than -3 dB. Similarly, Figure 6.4 illustrates the BER versus the channel SNR mismatch for RM $(32,26)^2$ - turbo code at different $\frac{E_b}{N_o}$, where at -2 dB or less of channel SNR mismatch, the performance degrades rapidly. From these three figures, it can be seen that the higher the $\frac{E_b}{N_o}$, the more the tolerance to channel SNR mismatch, for example in Figure 6.3, the start points of the performance degradation are -4, -3, -2 dB or less at $\frac{E_b}{N_o}$ of 3.5, 3, 2.5 dB, respectively. It is shown that the RM-turbo codes are more sensitive to underestimation of the channel SNR than overestimation of the channel SNR. The reason is that, in the case of underestimation, the factor used for calculating the soft-output was smaller than it should have been so less information could be extracted and transfered between two decoders resulting in no improvement from

the iterative decoding. Also the longer the code length, the more significant the effect of underestimation of channel SNR mismatch is obtained. This was due to the fact that the longer the code, the larger interleaver size and the more powerful the decoding process.

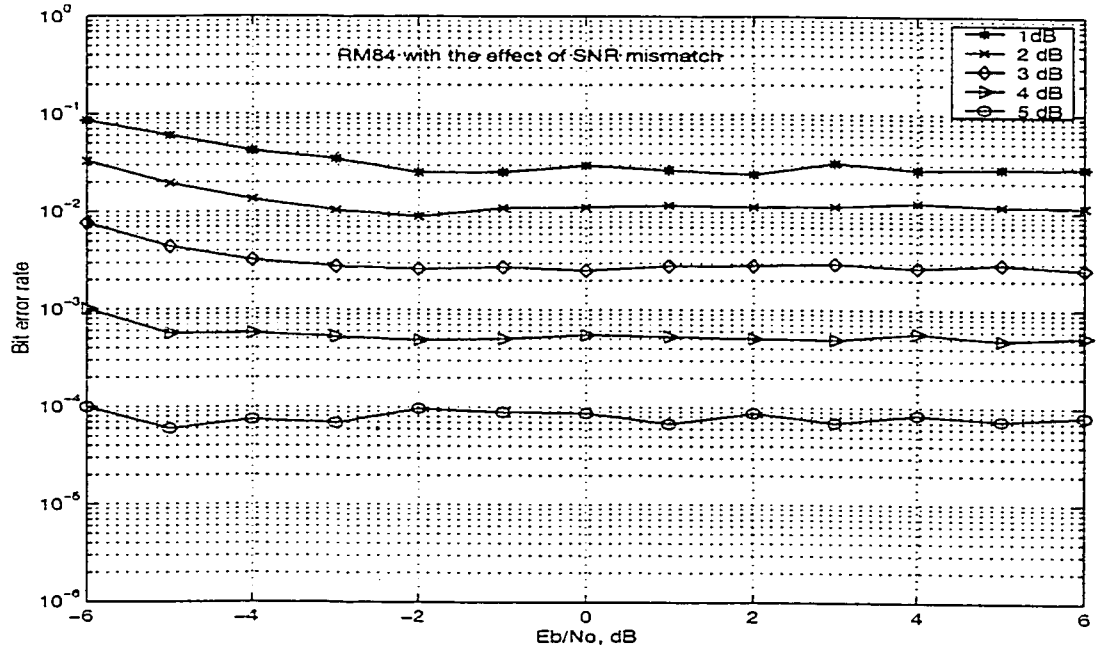

Figure 6.2: The effect of channel SNR mismatch on performance of a RM $(8,4)^2$ - turbo code

The BER versus $\frac{E_b}{N_o}$ of RM $(32,26)^2$ - turbo code with and without variance estimations on Gaussian channel is shown in Figure 6.5. It is shown that the estimation algorithm performs well. The BER versus $\frac{E_b}{N_o}$ of RM-turbo code with and without variance estimations on Rayleigh-fading channel is presented in Figure 6.6. The performance is better in the cases where the variance estimation is performed rather than the constant channel reliability is assumed, even though the perfect channel SNR is used; this is due to the time variant of fading channel. However, the improvement is modest in the case of RM $(8,4)^2$ and $(16,11)^2$ turbo codes because

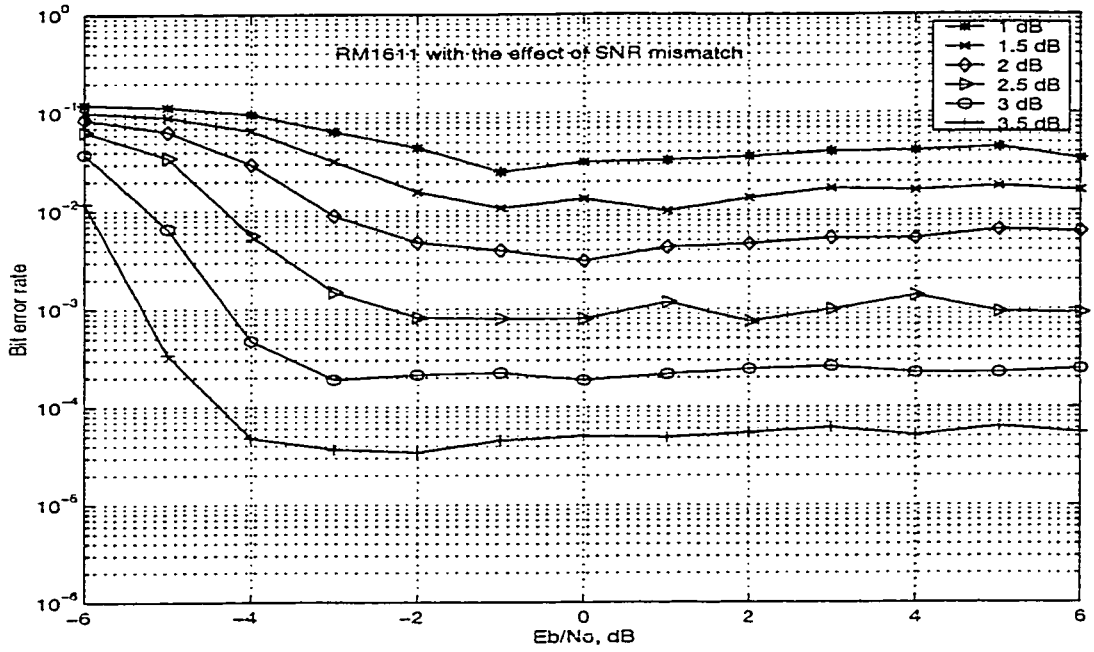

Figure 6.3: Effect of channel SNR mismatch on performance of a RM $(16, 11)^2$ - turbo code

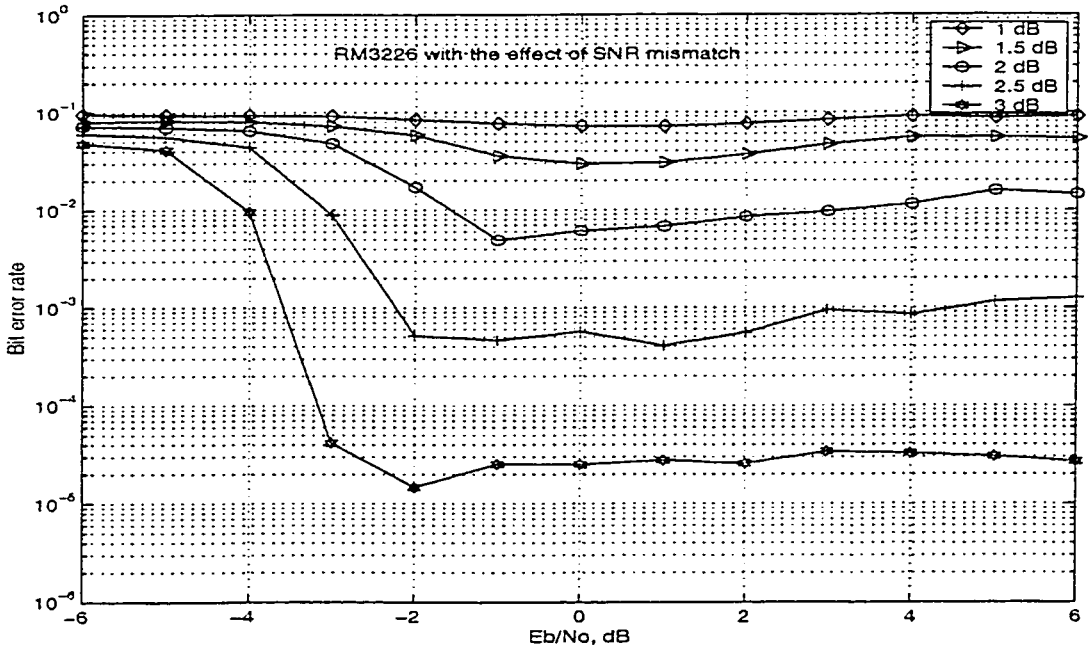

Figure 6.4: Effect of channel SNR mismatch on performance of a RM $(32, 26)^2$ - turbo code

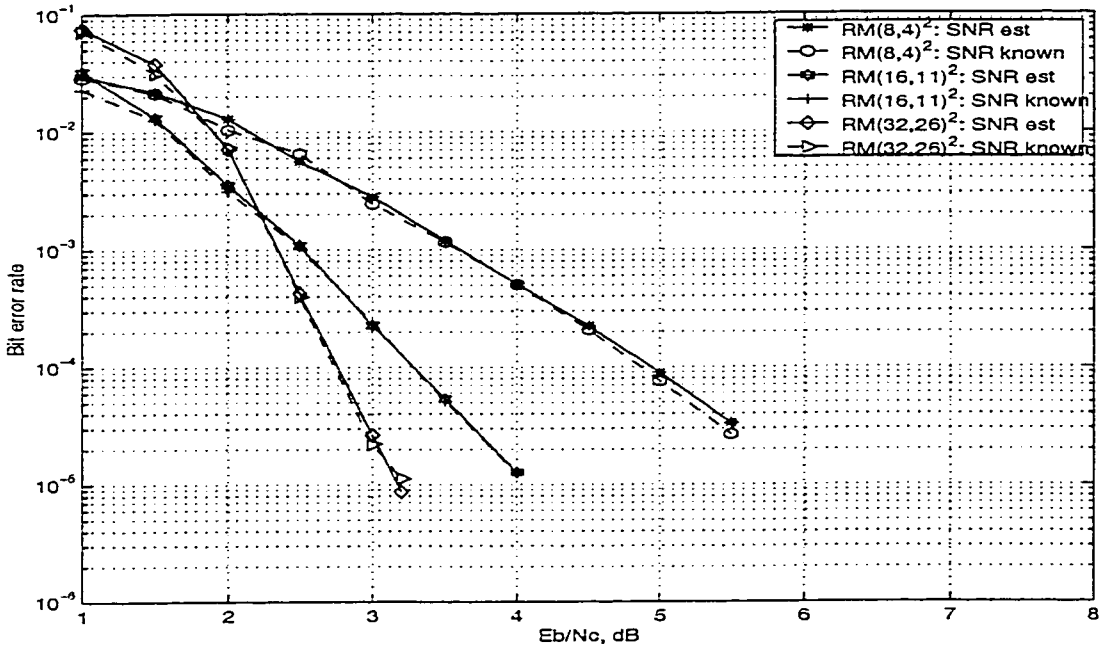

Figure 6.5: Performance of a RM $(32, 26)^2$ - turbo code with and without variance estimation on a Gaussian channel

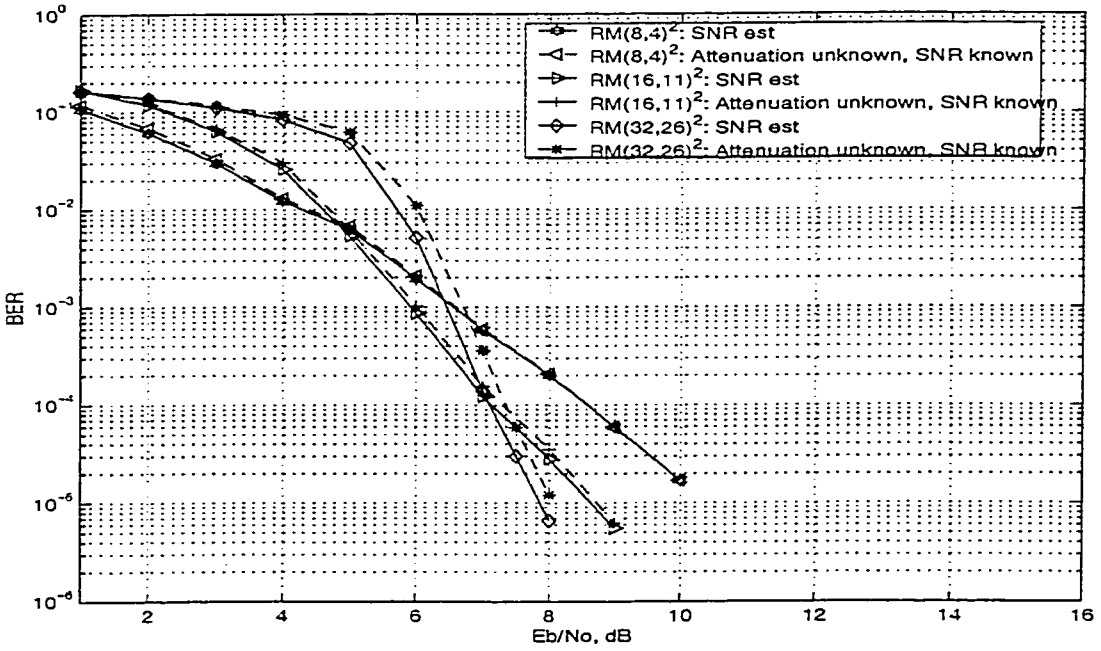

Figure 6.6: Performance of a RM $(32, 26)^2$ - turbo code with and without variance estimation on a Rayleigh-fading channel

they are less sensitive to channel SNR mismatch.

6.4 Carrier Phase Recovery

In this section, we consider the problem of carrier phase estimation when Quadrature Phase-Shift Keying (QPSK) modulation is used in the proposed turbo coding scheme. We split our investigation into two parts as follows:

6.4.1 Effect of Phase Offset on the Performance of RM Turbo Codes:

The effect of phase offset, $\triangle\phi$, on the received signal is given in equation 6.1, where $\alpha(t) = 1$ for a Gaussian channel. The symbol waveform can be represented in-phase and quadrature-phase terms and is given by

$$s(t) = s_I(t) + j\, s_Q(t) \tag{6.9}$$

Thus, the received signal corrupted by noise and phase offset is given by

$$\begin{aligned} r(t) &= (s_I(t) + j\, s_Q(t))\,(cos(\triangle\phi) + j\, sin(\triangle\phi)) + n(t) \qquad (6.10)\\ &= (s_I(t)cos(\triangle\phi) - s_Q(t)sin(\triangle\phi)) + j\ (s_Q(t)cos(\triangle\phi) + s_I(t)sin(\triangle\phi)) + n(t) \end{aligned}$$

6.4.2 Effect of Preamble Size on the Performance of RM Turbo Codes:

One way to recover the carrier phase is to send uncoded preamble bits through the channel along with the coded information. In the shortened turbo code, the set-zero information is not sent, however for synchronization purpose, some of the zeros are sent as preamble. The carrier phase can be computed by using the received preamble

symbols. The estimated carrier phase is

$$\widehat{\phi} = \arctan\left(\frac{\sum_{i=0}^{V} r_{Ii}}{\sum_{i=0}^{V} r_{Qi}}\right) \tag{6.11}$$

where r_{Ii} and r_{Qi} are in-phase and quadrature-phase components of the received signal. V is the number of QPSK symbols in the preamble.

6.4.3 Simulation Results

In Figure 6.7, we give the BER versus E_b/N_o of the shortened turbo code of case C (see section 5.3) with different variances of phase offset on an AWGN channel. The $\frac{E_b}{N_o}$ loss due to variances of phase offset of 0.001, 0.005, 0.01, 0.02, 0.04, 0.06, 0.1 $(rad)^2$ are 0, 0.1, 0.2, 0.4, 1.4, 2.4, 5 dB compared to no phase offset at BER of 10^{-4}. However, at variance of 0.22 $(rad)^2$, there is an error floor which means that very little improvement in terms of performance obtained when E_b/N_o is increased. In Figure 6.8, we present the BER curve of the effect of preamble size with $\Delta\phi \sim U\,[-\pi, \pi]$. The results show that with 50 preamble symbols (100 bits), the effect of phase offset is completely removed. A preamble length of 25 results in 0.25 dB degradation.

6.5 Conclusion

In this chapter, we presented the effect of channel impairments, including channel SNR mismatch and phase offset, on the performance of the RM turbo codes. Also the effect of preamble size used to recover the carrier phase was investigated. The results showed that the RM turbo code was more sensitive to the underestimation of the SNR than to the overestimation of SNR. The tolerance of SNR mismatch was -2 to 6 dB for RM $(32, 26)^2$ turbo code and more tolerant for the turbo codes with shorter code lengths. It was shown that a small phase offset (variance less than 0.02)

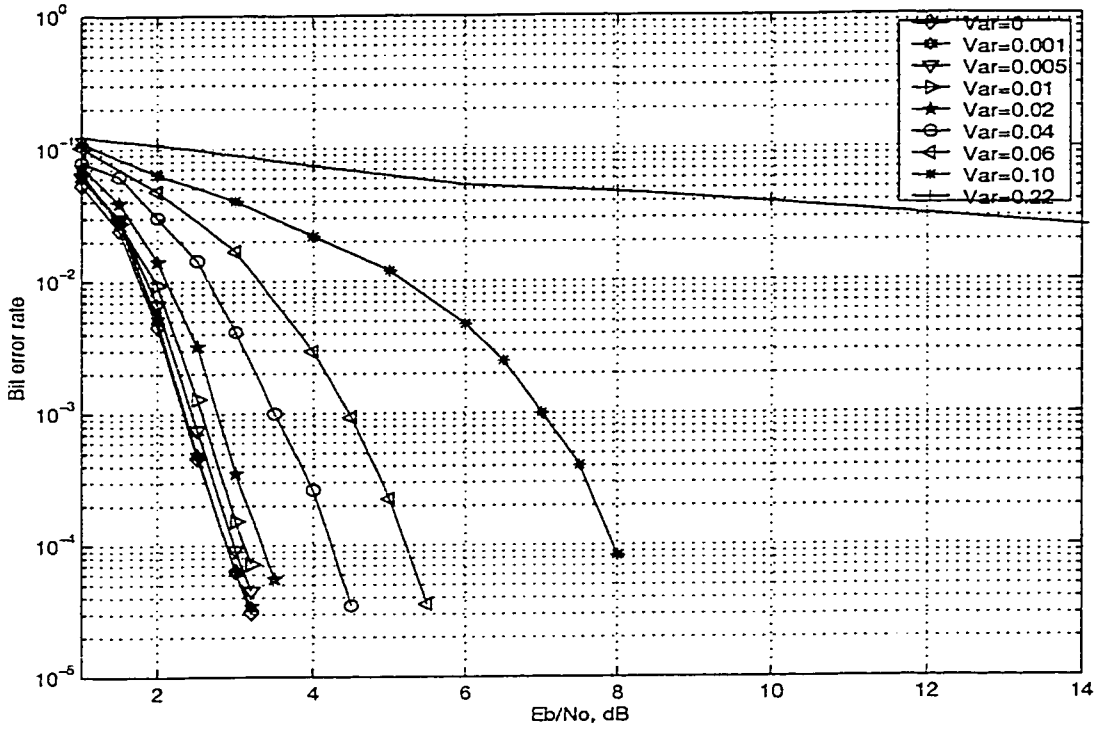

Figure 6.7: Effect of phase offset on the performance of shortened RM-turbo code case C.

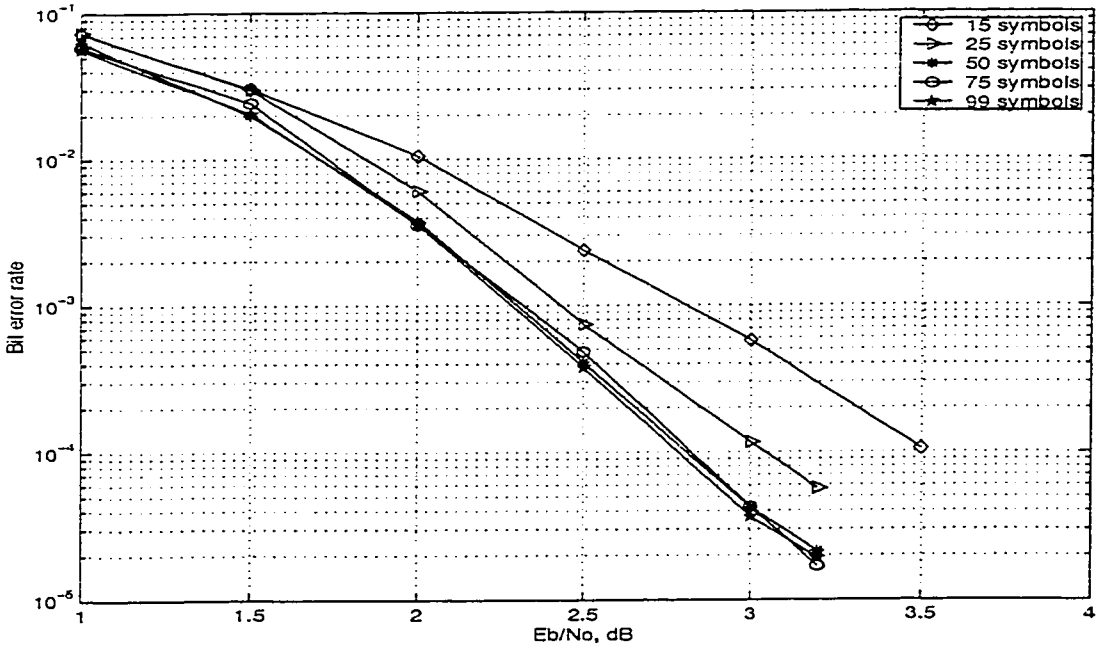

Figure 6.8: Effect of preamble sizes on the performance of shortened RM-turbo code case C.

is bearable however, beyond this, carrier phase offset should be compensated. We showed that a preamble of 50 symbols for QPSK modulation scheme was enough to recover the carrier phase completely.

Chapter 7

Conclusions and Suggestions for Future Research.

7.1 . Conclusions

In this thesis, a new turbo coding scheme with Reed Muller (RM) codes as component codes is proposed. Simulation results showing the performance of these turbo codes on Gaussian and Rayleigh-fading channels using QPSK modulation scheme were presented. These RM-turbo codes were the two-dimensional parallel concatenated codes using RM codes as their component codes with built-in block interleaver. The codes could be thought of as product codes without parity on parity. The trellis-based MAP decoding scheme was used consisting of two MAP decoders where each of them accepted soft-input and provided soft-output. The extrinsic or extra information obtained from code constraint was traded between the two decoders under iterative decoding process.

We also presented a modification to the above class of turbo codes which is the shortening of the codes to match a given cell-size in the application of satellite and wireless ATM for DVB-RCS. The design of shortened patterns was presented. When

shortening the codes, one may reduce the number of parity bits on the shortened section or use the extra parity bits to provide the Unequal Error Protection property for the shortened codes. This fits to the nature of ATM network, i.e., the fact that the ATM cell-header contains connection information which is more important than its payload.

We evaluated the proposed coding scheme from a more practical point of view by considering channel impairments as the imperfect estimation of the system parameters such as channel SNR and the carrier phase. Furthermore, the effect of preamble size used to recover the carrier phase on the performance of the shortened RM-turbo codes was presented where the preamble used in this case was the shortened bits which were sent along with coded bits.

From simulation results of the above investigations presented in Chapter 4, 5 and 6, the following conclusions were reached.

- The RM-turbo codes presented in this thesis outperform the results presented in [31], which used Hamming codes as the component codes by at least 0.5 dB at BER of 10^{-5}. The reason was that the RM codes used here could be considered as the extended version of Hamming codes leading to higher minimum distant than Hamming codes so they are stronger codes.

- The shortened RM-turbo codes with Equal Error Protection (EEP) performed, on the average, slightly better than the UEP codes. However, the UEP codes had nice property which is suitable for ATM applications. We believe that our UEP and EEP codes introduced in this work are just the preliminary results with possibility of being improved further and can be a candidate for Satellite ATM in DVB-RCS applications.

- The RM $(32, 26)^2$-turbo code and its shortened versions provided coding gain up to 0.5 dB over the existing coding scheme used for DVB-RCS i.e., the RS

(73,57) and convolutional code rate of $\frac{1}{2}$ concatenated code at BER of 10^{-6} with about 1.6 times higher code rate than that of the concatenated code which is equivalent to an additional coding gain of about 2 dB.

- Our result in the case for ATM cell was better than the performance of turbo coding schemes which was reported in [43] using extended Hamming codes as component codes of product codes about 0.3 dB with a higher code rate that can be thought of overall 0.5 dB gain. However, it was worse than that of the DVB-RCS standard proposed recently in [26] which used double-binary Circular Recursive Systematic Convolutional (CRSC) component codes with non-uniform interleaver and local disorder of data couples about 0.8 dB.

- RM-turbo codes were less tolerable of the underestimation of the channel SNR than the overestimation. Moreover, the longer the code length, the less tolerant channel SNR mismatch was. In longer code, the RM $(32, 26)^2$-turbo code could bear the channel SNR mismatch in the range from -2 to 6 dB, whereas the shorter RM-turbo codes i.e. the RM $(16, 11)^2$ and RM $(8, 4)^2$ -turbo codes could withstand the SNR mismatch in the range from -3 to 6 dB and -4 to 6 dB, respectively. The channel SNR estimation is more essential in a Rayleigh-fading channel than in an AWGN channel. The reason due to the time varying attenuation, even exact knowledge of the channel SNR does not provide full knowledge of the fading channel.

- The shortened RM-turbo codes proposed for satellite ATM application using QPSK modulation scheme were sensitive to carrier phase offset where the small phase offset (variance less than 0.02) was bearable but not beyond that without any carrier recovery. In addition, the preamble size of 50 symbols (100 bits) was needed to recover the carrier phase completely for this coding scheme. A preamble size of 25 symbols resulted in about 0.25 dB degradation at BER of 10^{-4}.

Bibliography

[1] C. E. Shannon, "A Mathematical Theory of Communication," *Bell System Technical Journal*, Vol. 27, pp. 379-423 and 623-656, July 1948.

[2] G. D. Forney, *Concatenated Codes*, Cambridge, MA: MIT Press, 1966.

[3] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo-codes (1)," in *Proc. IEEE Int. Conf. on Comm.*, pp. 1064-1070, Geneva, Switzerland, May 1993.

[4] D. Chase, " A Class of Algorithm for Decoding Block Codes with Channel Measurement Information," *IEEE Trans. on Inform. Theory*, Vol. IT-18, pp. 170-182. Jan. 1972

[5] L. R. Bahl, J. Cocke, F. Jelinek and J. Raviv, "Optimal Decoding of Linear Codes for Minimizing Symbol Error Rate," *IEEE Trans. on Inform. Theory*, Vol. IT-20, pp. 284-287, March 1974.

[6] B. Vucetic and J. Yuan, *Turbo Codes: Principles and Applications*, Kluwer Academic Publishers, Massachusetts, USA, 2000.

[7] W. J. Blackert, E. K. Hall and S. G. Wilson, "Turbo Code Termination and Interleaver Conditions," *Electronics Letters*, Vol. 31, No. 24, November 1995.

[8] A. S. Barbulescu and S. S. Piertrobon, "Terminating the Trellis of Turbo-Codes in the Same State," *IEEE Electronics Letters*, Vol. 31, No. 1, pp. 22-23, Jan. 1995.

[9] J. Hokfelt, O. Edfors, and T. Maseng, "Interleaver Design for Turbo Codes Based on the Performance of Iterative Decoding," in *Proc. IEEE Inter. Conf. on Comm.*, Vancouver, BC, Canada, pp. 93-97, June 1999.

[10] D. Divsalar and F. Pollara, "Turbo Codes for PCS Applications," in *Proc. IEEE Int. Conf. on Comm.*, pp. 54-59, May 1995.

[11] J. Yuan, B. Vucetic and W. Feng, "Combined Turbo Codes and Interleaver Design," *in IEEE Trans. on Comm.*, Vol. 47, No. 4, pp. 484-487, April 1999.

[12] S. Benedetto and G. Montorsi, "Unveiling Turbo Codes: Some Results on Parallel Concatenated Coding Schemes," *IEEE Trans. on Inform. Theory*, Vol. 42, No. 2, March 1996.

[13] D. Divsalar, S. Dolinar, R. J. McEliece, and F. Pollara, "Transfer Function Bounds on the Performance of Turbo Codes," *TDA Progress Report 42-121*, JPL, Aug. 1995.

[14] E. K. Hall and S. G. Wilson, "Design and Performance Analysis of Turbo Codes on Rayleigh Fading Channels," in *Proc. 5th Mini-conf. on Comm. GLOBECOM*, London, UK, pp. 16-20, Nov. 1996.

[15] J. Hagenauer, "Source-Controlled Channel Decoding," *IEEE Trans. on Comm.*, Vol. 43, pp. 2449-57, Sep. 1995.

[16] M. Reinhardt and T. Frey, "Turbo-Equalization for Symbol-Spread Block Transmission System," *IEEE Electronics Letters*, Vol. 32, pp. 2321-2323, Dec. 1996.

[17] D. Yellin, A. Vardy, and O. Amrani, "Joint Equalization and Coding for Intersymbol Interference Channels," *IEEE Trans. on Inform. Theory*, Vol. 43, pp. 409-25, March 1997.

[18] G. Bauch, " Concatenation of Space-Time Block Codes and "Turbo"-TCM," in *Proc. IEEE Int. Conf. on Comm.*, pp. 1202-1206, June 1999.

[19] G. Bauch, J. Hagenauer and N. Seshadri, "Turbo-TCM and Transmit Antenna Diversity in Multipath Fading Channel," in *Proc. Int. Symp. on Turbo Codes and Related Topics*, Brest, France, pp. 189-192, Sept. 2000.

[20] S. Goff, A. Glavieux, and C. Berrou, "Turbo-Codes and High Spectral Efficiency Modulation," in *Proc. IEEE Int. Conf. on Comm.*, pp. 645-9, May 1994.

[21] S. Benedetto, D. Divsalar, G. Montorsi, and F. Pollara, "Parallel Concatenated Trellis Coded Modulation," in *Proc. IEEE Int. Conf. on Comm.*, pp. 974-978, May1996.

[22] S. Benedetto and G. Montorsi, "Versatile Bandwidth-Efficient Parallel and Serial Turbo-Trellis-Coded Modulation," in *Proc. Int. Symp. on Turbo Codes and Related Topics*, Brest, France, pp. 201-208, Sept. 2000.

[23] P. Jung, M. Nasshan, and J. Blanz, "Application of Turbo-Codes to a CDMA Mobile Radio System using Joint Detection and Antenna Diversity," in *Proc. IEEE Veh. Tech. Conf.*, Stockholm, pp. 770-774, 1994.

[24] K. V. Ravi and T. Soh Khum, "Performance of Turbo TCM in Wideband CDMA applications," in *Proc. Int. Symp. on Turbo Codes and Related Topics*, Brest, France, pp.399-402, Sept. 2000.

[25] R. Cusani and D. Crea, "Recursive Interference Cancellation and Turbo Decoding for TDD-CDMA Link," in *Proc. Int. Symp. on Turbo Codes and Related Topics*, Brest, France, pp.523-526, Sept. 2000.

[26] C. Douillard, M. Jezequel, C. Berrou, N. Brengarth, J. Tousch and N. Pham, "The Turbo Code Standard for DVB-RCS," in *Proc. Int. Symp. on Turbo Codes and Related Topics*, Brest, France, pp.535-538, Sept. 2000.

[27] J. Fang, F. Buda and E. Lemois, "Turbo Product Code: A Well Suitable Solution to Wireless Packet Transmission for Very Low Error Rates," in *Proc. Int. Symp. on Turbo Codes and Related Topics*, Brest, France, pp. 101-111, Sept. 2000.

[28] J. Lodge, P. Hoeher and J. Hagenauer, "The Decoding of Multidimensional Codes using Separable MAP Filters," in *Proc.* 16^{th} *Biennial on Communications*, Queen's University, pp. 343-346, May 1992.

[29] J. Lodge, R. Young, P. Hoeher and J. Hagenauer, "Separable "Filters" for the Decoding of Product and Concatenated Codes," in *Proc. IEEE Int. Conf. on Comm.*, Geneva, Switzerland, pp. 1740-1745, May 1993.

[30] P. Robertson, E. Villebrun, and P. Hoeher, "A Comparison of Optimal and Sub-Optimal MAP Decoding Algorithms Operating in the Log Domain," in *Proc. IEEE Int. Conf. on Comm.*, Seattle, WA, pp. 1009-1013, June 1995.

[31] J. Hagenauer, "Iterative Decoding of Binary Block and Convolutional Codes," *in IEEE Trans. Inform. Theory*, Vol. 42, No. 2, pp. 429-445, March 1996.

[32] Y. Liu, H. Tang, M. Fossorier and S. Lin, "Iterative Decoding of Concatenated Reed-Solomon Codes," 37^{th} *Annual Allerton. Conf.*, Sept. 1999.

[33] Y. Liu, S. Lin, and M. Fossorier, "MAP Algorithm for Decoding Linear Block Codes Based on Sectionalized Trellis Diagrams," *IEEE Trans. on Inform. Theory*, Vol. 48, No. 4, April 2000.

[34] R. Pyndiah, A. Glavieux, A. Picart, and S. Jacq, "Near Optimum Decoding of Product Codes," in *Proc. IEEE GLOBECOM*, San Francisco, USA, pp. 339-343, Nov. 1994.

[35] R. Pyndiah, "Near-Optimum Decoding of Product Codes: Block Turbo Codes," *in IEEE Trans. on Comm.*, Vol. 46, No. 8, pp. 1003-1010, August 1998.

[36] O. Aitsab, R. Pyndiah, "Performance of Reed Solomon Block Turbo Codes," in *Proc IEEE GLOBECOM*, London, UK, pp. 121-125, Nov. 1996.

[37] R. Pyndiah, Pierre Combelles and P. Adde, "A Very Low Complexity Block Turbo Decoder for Product Codes," in *Proc. IEEE GLOBECOM*, London, pp. 101-105, Nov. 1996.

[38] P. Adde and R. Pyndiah, "Recent Simplifications and Improvements in Block Turbo Codes," in *Proc. Int. Symp. on Turbo Codes and Related Topics*, Brest, France, pp. 133-136, Sept. 2000.

[39] S. A. Hirst, B. Honary and G. Markarian, "Fast Chase Algorithm with Application in Turbo Decoding," in *Proc. Int. Symp. on Turbo Codes and Related Topics*, Brest, France pp. 259-262, Sept. 2000.

[40] AHA. "PS4501: Astro 36 Mbits/s Turbo Product Code Encoder/Decoder".

[41] S. Kerouedan and P. Adde, "Implementation of a Block Turbo Decoder on a Single Chip," in *Proc. Int. Symp. on Turbo Codes and Related Topics*, Brest, France, pp.243-246, Sept. 2000.

[42] A. Goalic and N. Chapalain, "Real Time Turbo Decoding of BCH Product Code on the DSP Texas TMS320C6201," in *Proc. Int. Symp. on Turbo Codes and Related Topics*, Brest, France, pp. 331-334, Sept. 2000.

[43] M. Vanderaar, R. T. Gedney and E. Hewitt, "Comparative Performance of Turbo Product Codes and Reed-Solomon/Convolutional Concatenated Codes

for ATM Cell Transmission," *Fifth Ka Band Utilization Conf.*, Toarmina, Italy, October 1999.

[44] S. Lin and D. J. Costello, *Error Control Coding Fundamentals and Applications*, Prentice-Hall, Englewood Cliffs, NJ, USA, 1983.

[45] S. B. Wicker, *Error Control Systems for Digital Communication and Storage*, Prentice-Hall, Englewood Cliffs, NJ, USA, 1995.

[46] J. K. Wolf, "Efficient Maximum-Likelihood Decoding of Linear Block Codes," *IEEE Trans. on Inform. Theory*, Vol. IT-24, pp. 76-80, Jan. 1978.

[47] G.D. Forney, Jr., "Coset codes II: Binary Lattices and Related Codes," *IEEE Trans. on Inform. Theory*, Vol. 34, No. 5, pp. 1152-1187, Sept. 1988.

[48] D. J. Muder, "Minimal Trellises for Block Codes," *IEEE Trans. on Inform. Theory*, Vol. 34, No. 5, pp. 1049-1053, Sept 1988.

[49] Y. Berger and Y Be'ery, "The Twisted Squaring Construction Trellis Complexity and Generalized Weights of BCH and QR codes," *IEEE Trans. on Inform. Theory*, Vol. 42, No. 6, pp.1817-1827, Nov. 1996.

[50] F. R. Kschischang and V. Sorokine, "On the Trellis Structure of Block Codes", *IEEE Trans. on Inform. Theory*, Vol. 41, No. 6, pp. 1924-1937, Nov. 1995.

[51] R. J. McEliece, "On the BCJR Trellis for Linear Block Codes," *IEEE Trans. on Inform. Theory*, Vol. 42, No. 4, pp. 1072-1092, July 1996.

[52] G. Horn and F. R. Kschischang, "On the Intractability of Permuting a Block Code to Minimize Trellis Complexity," *IEEE Trans. on Inform. Theory*, Vol. 42, No. 6, pp. 2042-2048, Nov. 1996.

[53] T. Kasami, T. Takata, T. Fujiwara, and S. Lin, "On the State Complexity of Trellis Diagrams for Reed-Muller Codes and their Supercodes," *Proc. 14th*

Symp. on Inform. Theory and Its Applications, Ibusuki, Japan, pp. 101-104, Dec. 1991.

[54] J. L. Massey, "Foundation and methods of channel encoding," *Proc. Int. Conf. Information Theory and Systems*, vol. 65, NTG-Fachberichte, Berlin, pp. 148-157, 1978.

[55] V. S. Pless and W. C. Huffman, *Editors, Handbook of Coding Theory*, Elsevier Science B.V., Volume II, Amsterdam, Netherlands, pp. 1989-2117, 1998.

[56] A. M. Michelson and A. H. Levesque, *Error-Control Techniques for Digital Communication*, John Wiley & Sons, 1985.

[57] M. Vanderaar, Efficient Channel Coding (ECC) Inc., Personal correspondence.

[58] W. Stallings, *Data and Computer Communications*, Fifth edition, Prentice-Hall, New Jersey, 1997.

[59] ETSI standard, *Standard Digital Video Broadcasting (DVB) Interaction Channel for Satellite Distribution*, ETSI EN301 790, V1.22, Dec. 2000.

[60] T. A. Summers and S. G. Wilson, "SNR Mismatch and Online Estimation in Turbo Decoding," *IEEE Trans. on Comm.* , Vol. 46, No.4, April 1998.

[61] W. Oh and K. Cheun, "Adaptive Channel SNR Estimation Algorithm for Turbo Decoder," *IEEE Communication Letters*, Vol. 4, No. 8, August 2000.

[62] M. S. C. Ho and S. S. Pietrobon, "A Variance Mismatch Study for Serial Concatenated Turbo Codes," in *Proc. Int. Symp. on Turbo Codes and Related Topics*, Brest, France, pp. 483-485, Sept. 2000.

[63] U. Vilaipornsawai and M. R. Soleymani, "Turbo Codes for Satellite and Wireless ATM," accepted from *IEEE Int. Conf. on Information Technology: Coding and Computing (ITCC)*, Las Vegas, USA, 2001.

Appendix A

Derivation of the A Priori Probabilities and the Conditioned Probabilities

The derivation presented in Appendix A is the extended details from [31]. The log-likelihood ratio of the a priori probability of a binary random variable is given by

$$L(u) = log\frac{P(u=+1)}{P(u=-1)} \tag{A-1}$$

From Equation (A-1) where $P(u=+1) = 1 - P(u=-1)$, then

$$L(u) = log\frac{P(u=+1)}{1-P(u=+1)} \tag{A-2}$$

$$e^{L(u)} = \frac{P(u=+1)}{1-P(u=+1)} \tag{A-3}$$

$$\begin{aligned} P(u=+1) &= \frac{e^{L(u)}}{1+e^{L(u)}} \\ &= \frac{1}{1+e^{-L(u)}} \end{aligned} \tag{A-4}$$

similarly,

$$\begin{aligned} P(u=-1) &= 1-\frac{1}{1+e^{-L(u)}} \\ &= \frac{e^{-L(u)}}{1+e^{-L(u)}} \end{aligned} \tag{A-5}$$

so

$$P(u=\pm 1)=\frac{e^{\pm L(u)}}{1+e^{\pm L(u)}}=\left(\frac{e^{-L(u)/2}}{1+e^{-L(u)}}\right)\cdot e^{L(u)\cdot u/2}=A\cdot e^{L(u)\cdot u/2} \tag{A-6}$$

The conditional log-likelihood ratio of a binary random variable is given below,

$$L(y \mid u)=log\frac{P(y \mid u=+1)}{P(y \mid u=-1)} \tag{A-7}$$

where

$$P(y)=P(y \mid u=+1)\cdot P(u=+1)+P(y \mid u=-1)\cdot P(u=-1) \tag{A-8}$$

$$P(y \mid u=-1)=\frac{P(y)}{P(u=-1)}-\frac{P(y \mid u=+1)\cdot P(u=+1)}{P(u=-1)} \tag{A-9}$$

$$L(y \mid u)=log\frac{P(y \mid u=+1)}{\frac{P(y)}{P(u=-1)}-\frac{P(y|u=+1)\cdot P(u=+1)}{P(u=-1)}} \tag{A-10}$$

$$e^{L_c\cdot y}=\frac{P(y \mid u=+1)}{\frac{P(y)}{P(u=-1)}-\frac{P(y|u=+1)\cdot P(u=+1)}{P(u=-1)}} \tag{A-11}$$

$$P(y \mid u=+1)=\frac{\frac{P(y)}{P(u=-1)}e^{L_c\cdot y}}{1+\frac{P(u=+1)}{P(u=-1)}e^{L_c\cdot y}} \tag{A-12}$$

substitute $P(u=+1)$ and $P(u=-1)$ from Equation (A-6)

$$\begin{aligned} P(y \mid u=+1) &= \frac{\frac{P(y)\cdot(1+e^{-L(u)})}{e^{-L(u)}} e^{L_c \cdot y}}{1+e^{(L(u)+L_c \cdot y)}} \\ &= \frac{\frac{P(y)\cdot(1+e^{-L(u)})}{e^{-L(u)}} \cdot e^{-(L(u)+L_c \cdot y)} e^{L_c \cdot y}}{1+e^{-(L(u)+L_c \cdot y)}} \\ &= \frac{P(y)\cdot(1+e^{-L(u)})}{1+e^{-(L(u)+L_c \cdot y)}} \end{aligned} \tag{A-13}$$

Similarly,

$$P(y \mid u=-1) = \frac{P(y)\cdot(1+e^{-L(u)})\cdot e^{-L_c \cdot y}}{1+e^{-(L(u)+L_c \cdot y)}}$$

$$P(y \mid u=\pm 1) = \left(\frac{P(y)\cdot(1+e^{-L(u)})\cdot e^{-L_c \cdot y/2}}{1+e^{-(L(u)+L_c \cdot y)}} \right) \cdot e^{L_c \cdot y \cdot u/2} = B \cdot e^{L_c \cdot y \cdot u/2}$$