Joint Source-Channel Coding via Turbo Codes

by

Guang-Chong Zhu

A dissertation submitted to the

Department of Mathematics and Statistics

in conformity with the requirements

for the degree of

Doctor of Philosophy

Queen's University

Kingston, Ontario, Canada

January 2003

Abstract

Based on Shannon's Separation Principle, source and channel coding are often treated independently, resulting in what we call a *tandem* coding system. This separation of source and channel coding incurs no loss of optimality provided that the sequence length approaches infinity. In practical implementations, however, due to limitations on system delay and complexity, the infinite sequence length assumption is often not valid. Therefore, joint source-channel coding may be a more suitable approach in practice, as it can significantly outperform traditional tandem coding systems. During the past few decades, significant improvements have been achieved in the two separate areas of source and channel coding. One of the most exciting breakthroughs in channel coding is the invention of *Turbo codes*, whose error-correcting performance is unprecedentedly close to the Shannon limit. In this thesis, we investigate three joint source-channel coding issues in the context of Turbo codes.

In the first part of the thesis, a robust soft-decision channel-optimized vector quantization (COVQ) scheme for Turbo-coded additive white Gaussian noise (AWGN) and Rayleigh fading channels used in conjunction with binary phase shift keying (BPSK) modulation is proposed. The log-likelihood ratio (LLR) generated by the Turbo decoder is exploited via the use of a q-bit scalar soft-decision demodulator. The concatenation of the Turbo encoder, modulator, AWGN channel or Rayleigh fading channel, Turbo decoder, and q-bit soft-decision demodulator is modeled as an expanded discrete memoryless channel (DMC), or a discrete block-memoryless channel (DBMC). A COVQ scheme for these expanded discrete channel models is designed. Numerical results indicate substantial performance improvements over traditional tandem coding systems, and COVQ schemes designed for hard-decision demodulated Turbo-coded channels ($q = 1$). We also compare our system with a recent soft Turbo decoding COVQ scheme proposed by Ho, in which the entire unquantized soft-decision information provided by the LLR of the Turbo decoder is utilized. Our DMC-based system offers comparable performance, while the DBMC-based system outperforms Ho's scheme. Furthermore, the decoding complexity of both of our systems is lower but the storage requirement is higher.

Turbo codes were originally designed for the transmission of uniform independent and identically distributed (i.i.d.) sources, which contain zero redundancy. In the second part of the thesis, we first design systematic Turbo codes for the reliable communication of non-uniform i.i.d. sources over BPSK modulated AWGN and Rayleigh fading channels. One main objective is to achieve the best possible performance vis-a-vis the Shannon limit. The redundancy in the form of non-uniformity is exploited in the decoder. The encoder structure, on the other hand, is optimized to further enhance the performance. We prove that for recursive convolutional encoders, under certain (necessary and sufficient) conditions, the state and output marginal distributions are asymptotically uniform, irrespective of the degree of non-uniformity in the

source distribution. We also observe a significant mismatch between the biased systematic Turbo encoder stream and the channel inputs, which limits the best possible achievable performance of the system. To eliminate this mismatch, we hence investigate non-systematic Turbo codes for this problem, and achieve significant gains over systematic Turbo codes with a performance within 0.74 to 1.17 dB from the Shannon limit. Finally, we compare our joint source-channel coding system with two tandem schemes which employ a 4^{th}-order Huffman code and a Turbo code that either gives excellent waterfall bit error rate (BER) performance or a low error-floor performance. At the same overall transmission rate, our system offers robust and superior performance at low BER levels (below 10^{-4}), while its complexity is lower.

The third part of the thesis investigates the design of Turbo codes for Markov sources to exploit the source redundancy in the form of memory. In the first constituent decoder, the decoding algorithm is modified by incorporating the Markovian property of the source. Due to interleaving, the second constituent decoder is designed for memoryless sources. However, when the source is asymmetric, it can also take advantage of the source redundancy in the form of non-uniformity. Furthermore, the extrinsic information from the second decoder is modified via a weighted correction term. Simulation results demonstrate substantial gains (from 1.29 up to 3.57 dB) over the original Turbo codes' performance. As a result, the OPTA gaps are in the range of 0.94—1.45 dB. Finally, similar to the scenario of non-uniform i.i.d. sources, we compare the performance of our joint source-channel coding system designed for Markov sources with two tandem schemes over AWGN channels. All schemes have the same overall transmission rate. To achieve near-optimal data compression, a 8^{th}-order Huffman code is adopted. While the tandem schemes suffer from an inevitable high BER performance, our system yields a substantially better performance and enjoys a lower system complexity since the source encoding and decoding are omitted.

Acknowledgements

First of all, I would like to thank my supervisor, Dr. Fady Alajaji, for his excellent supervision and guidance. This thesis would not have been completed without his continuous support, sincere encouragement and valuable suggestions. I would also like to thank Dr. Jon Davis for his guidance and helpful discussions.

I also wish to thank Dr. Jan Bajcsy and Patrick Mitran for enjoyable and fruitful collaborations.

I want to say "Thank you" to all my friends and classmates; our friendship will always be cherished.

During my Ph.D. studies, I have received financial support from the Ontario Graduate Scholarship for Sciences and Technologies (OGSST), the Queen's Graduate Fellowship, as well as a teaching assistantship from the Department of Mathematics and Statistics and a research assistantship from the Natural Sciences and Engineering Research Council (NSERC) of Canada and the Premier's Research Excellence Award (PREA) of Ontario. Several graduate conference awards from the Queen's School of Graduate Studies and Research provided me the opportunity to attend several international conferences. I gratefully acknowledge all these sources of financial support.

Last but not the least, I am deeply thankful to all my family members. My special thanks go to my wife, Ning, for her love, encouragement and support.

Statement of Originality

All results presented in this thesis are original, unless otherwise stated. The results quoted from the literature are presented as statements with indicated references.

Dedicated to the memory of my father

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

The fundamental goal of a communication system is to efficiently and reliably transmit information data from a source to a destination over a physical channel where noise distortion may occur. For the sake of efficiency, the source output is usually compressed prior to transmission. This *data compression* procedure (also known as *source coding*), in turn, renders the sources more vulnerable to noise corruption. In what follows, in order to make the transmission reliable, *channel coding* is usually employed to combat the noise by adding controlled redundancy to the data.

Traditionally, source and channel coding are treated as two independent subjects. In 1948, Shannon proved that there is no loss of optimality by implementing the

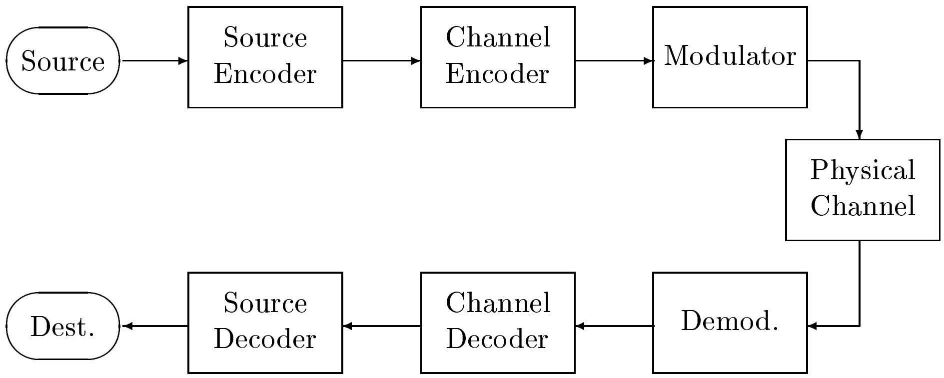

Figure 1.1: Block diagram of a digital communication system.

source and channel coding independently [69]. This is known as *Shannon's Separation Principle*. Therefore, the source encoder is designed to remove as much redundancy as possible to achieve bandwidth efficiency, and the goal of channel coding, on the other hand, is to make the transmitted signals robust to channel noise distortions such that they can be delivered to the destination nearly error-free. A typical digital communication system based on a separate design is depicted in Fig. 1.1. However, Shannon's Separation Principle is valid under the assumption that the information sequence length goes to infinity, resulting in infinite delay and system complexity. In practice, when delay and complexity are limited, joint source-channel coding schemes often outperform the traditional separately designed (or tandem) coding systems.

In his landmark paper "A mathematical theory of communication" [69], Shannon established the *Channel Coding Theorem*, which states that for any transmission rate below the channel capacity C, there exists a channel coding system that can achieve vanishing probability of error as long as the sequence length goes to infinity. On the other hand, when a desired transmission rate is given, the capacity C of channels with additive noise (such as AWGN and Rayleigh fading channels) is essentially a function of E_b/N_0, where E_b is the average energy per information bit, and $N_0/2$ is the additive noise variance; i.e., $C = C(E_b/N_0)$. The lowest possible value of E_b/N_0 at which the information can be reliably (with zero error probability) transmitted at the given rate is called the *Shannon limit.* The Shannon limit can also be determined for any desired value of the error probability by invoking Shannon's *Lossy Joint Source-Channel Coding Theorem* (also known as Shannon's *Lossy Information Transmission Theorem*). More detailed discussions on the Shannon limit will be elaborated in the thesis.

Shannon's proof of his Channel Coding Theorem is an existence proof; it does not give any suggestions on how to construct good channel coding schemes. Over the last five decades, motivated by Shannon's promise, researchers devoted great efforts to construct powerful and efficient coding structures that can approach the Shannon limit. In particular, the attention was focused on the transmission of Bernoulli (1/2) sources (or uniform independent identically distributed (i.i.d.) sources) over noisy channels. A motivation for this endeavor is given in [47], where it is quoted that 1 dB gain in channel coding is equivalent to 80 million dollars' saving in deep space communications.

For almost half a century, the gap between the Shannon limit and the best reported channel coding performance for the communication of a Bernoulli (1/2) source over an AWGN channel reported remained above 2 dB [47], until 1993 when Berrou, Glavieux, and Thitimajshima [14] made a breakthrough by constructing a scheme named "Turbo codes," whose performance is historically unmatched: at a channel coding rate of 1/2, and a BER level of 10^{-5}, the Turbo codes performance was only 0.5 dB away from the Shannon limit [15].

The extraordinary error-correcting capability of Turbo codes ignited great enthusiasm in the communications society. Since their debut in 1993, within less than 10 years, there has been a huge volume of publications regarding various aspects of Turbo codes. Turbo codes have indeed blossomed into one of the most exciting and active research areas in Telecommunications.

Among numerous research directions, in this thesis, we are particularly interested in the following joint source-channel coding issues in the context of Turbo codes. The first topic is to address the channel optimized vector quantization (COVQ) design problem for Turbo-coded AWGN and Rayleigh fading channels. The second part considers using Turbo codes based joint source-channel codes for the transmission of non-uniform i.i.d. sources over noisy channels. The third part examines joint source-channel Turbo codes for binary Markov sources.

1.2 Literature Review

The simplest case of quantization is scalar quantization, which deals with 1-dimensional signals. The first work on designing optimal scalar quantizers was by Max [58] and also independently by Lloyd in an unpublished report dated back to the 1960s. Their design algorithm is widely known as the Lloyd-Max algorithm. About 20 years later, Linde, Buzo and Gray extended the Lloyd-Max algorithm to vector quantizers, which is referred to as the LBG algorithm [52]. However, both the Lloyd-Max algorithm and the LBG algorithm do not take the channel noise distortion into consideration as they are designed for noiseless channels. The first work considering the effect of the channel noise distortion in a scalar quantization system was conducted by Kurtenbach and Wintz [51], resulting in the so-called channel optimized scalar quantizer (COSQ). Kumazawa *et al.* extended the COSQ to the high-dimensional vector space or channel-optimized vector quantizer (COVQ) [50]. The performance and complexity of the COVQ system is studied by Farvardin and Vaishampayan [29]. In [28], Farvardin also introduced the simulated annealing technique to the initial codebook design for the COVQ system. In [2, 62], Alajaji and Phamdo successfully implemented a COVQ system for AWGN and Rayleigh fading channels in conjunction with a q-bit soft-decision scalar quantizer at the channel output. Since Turbo codes can offer extraordinary error-correcting performance, we would like to incorporate Turbo codes in the design of the soft-decision COVQ system proposed in [2, 62].

Turbo codes were originally designed for uniform i.i.d. sources over AWGN channels. Later the work was extended to Rayleigh fading channels showing comparable performance [46]. Recently, McEliece pointed out that Turbo codes have also great potential on nonstandard channels (asymmetric, non-binary, and multiuser, etc.) [60]. However, most of the works in Turbo codes focus only on uniform i.i.d. sources. To the best of our knowledge, the issue of designing Turbo codes for non-uniform i.i.d. sources has not been fully studied. In [44], Hagenauer briefly mentioned that when Turbo codes are used for non-uniform i.i.d. sources, the extrinsic information needs to be modified such that the source redundancy in the form of non-uniformity can be exploited. A similar observation was mentioned in [33, 35]. However, there has been no study on the performance for such scenario, particularly vis-a-vis the Shannon limit.

On the other hand, in almost all the Turbo coding literature, the encoder structures of Turbo codes are exclusively *systematic*. The only series of works using non-systematic Turbo codes was conducted by Costello *Jr.* et al. [22, 21, 23, 57]. However, their attention is restricted to the uniform i.i.d. source case. Their goal is to find a code in the non-systematic Turbo codes family that can outperform the best systematic peers with the motivation that non-systematic Turbo codes have a larger code space. In [57], Massey *et al.* found an asymmetric non-systematic Turbo code that outperforms Berrou's (37,21) code by about 0.2 dB at the 10^{-5} BER level with a block length size of 4096. For larger block length, we expect that Massey's code should still outperform Berrou's code, although it is not clear if the 0.2 dB gain would be maintained.

Since the debut of Turbo codes, there has been a belief in the Turbo coding society that the two parity sequences generated by each constituent recursive convolutional encoder may (with high probability) have significantly different weight and thus different marginal distributions. For example, in [61], it is stated that when the first constituent encoder has low-weight parity output, a pseudo-random interleaver can ensure the second constituent encoder produces a high-weight parity sequence. Note that all constituent encoders considered in [61] are recursive convolutional encoders and the input sequence is assumed to be *uniform* and memoryless.

In our studies of Turbo codes for non-uniform i.i.d. sources, we empirically observed in our simulations that the marginal distribution of the parity output generated by most recursive convolutional encoders is almost uniform, even when an extremely biased non-uniform i.i.d. source is used as the input. Furthermore, this is observed for both constituent encoders' parity sequences; that is, they are both almost uniform regardless of the use of pseudo-random or other interleavers. This led us to the proof that under certain conditions, recursive convolutional encoders can produce asymptotically uniformly distributed parity sequences, irrespective to the degree of non-uniformity in the source distribution. A similar result was also separately obtained by Fair *et al.* in [27] in the context of line coding. Meanwhile, a recent work by Shamai and Verdú [67] attracted our attention. They proved that the empirical distribution of any good code (i.e., a code with rate approaching capacity and asymptotically vanishing probability of error) converges to the input distributions that achieve channel capacity. Furthermore, their conclusion is also valid for any k^{th}-order distribution, where k is an arbitrary positive integer. The capacity of a binary input AWGN or Rayleigh fading channel is achieved when its mutual information is maximized by a uniformly distributed input. The above observations led us to investigate the design of non-systematic Turbo codes for the transmission of non-uniform i.i.d. sources.

Despite the huge volume of publications regarding various aspects of Turbo codes, only limited amount of work has been devoted to the design of Turbo codes for sources with memory, such as Markov sources. Independently of our investigation on this issue, Garcia-Frias *et al.* first proposed using Turbo codes for Hidden Markov sources [33, 35]. Also in [30], Fazel considered joint source-channel decoding algorithms for the 2400 bps MELP speech coder. She considered using Turbo codes for the MELP speech coder output whose parameters are modeled by Markov chains. Another related work is by Cai *et al.* [19], in which a simple modification is proposed to the extrinsic information terms of both constituent decoders, and the decoding algorithm remains the same as for uniform i.i.d. sources.

Our work on Turbo codes for Markov sources is in the same spirit as the work of Garcia-Frias *et al.* and Fazel, where the modifications to the Bahl-Cocke-Jelinek-Raviv (BCJR) algorithm [9] are proposed for the first constituent decoder. Due to interleaving, the second constituent decoder remains the same as for non-uniform i.i.d. sources, in which the source redundancy in the form of non-uniformity can be exploited. We also adopt a similar correction term for the extrinsic information from the second decoder passed on to the first; however, only marginal improvement is observed due to this correction term.

1.3 Contributions

The contributions of this thesis are as follows:

- The design and evaluation of a COVQ system for Turbo-coded AWGN and Rayleigh fading channels with known channel state information. Substantial gains are achieved in comparison with the original COVQ system, a tandem coding scheme, as well as Ho's recent COVQ Turbo coding scheme.

- The investigation of using standard systematic Turbo codes for non-uniform i.i.d. sources. Besides exploiting the source redundancy in the form of non-uniformity in the decoder, the encoder structure is optimized for a given source distribution, resulting in significantly enhanced performance.

- The examination of the performance loss due to the systematic structure in systematic Turbo codes for the transmission of non-uniform i.i.d. sources. This leads us to prove the asymptotic uniformity of the recursive convolutional encoder's state and parity output under certain conditions. As a result, we propose a more suitable solution to the issue of transmitting non-uniform i.i.d. sources over noisy channels: *recursive non-systematic* Turbo codes. We achieve a close to Shannon limit performance, which is also superior to that of two typical tandem schemes at low bit error rates (below 10^{-4}).

- The investigation of Turbo codes for Markov sources. We propose a modified BCJR algorithm in the first decoder to incorporate the Markovian source property, as well as a weighted correction term in the second decoder's extrinsic information. Significant gains over the original Turbo codes are achieved. In addition, performance comparisons with tandem schemes are also made, where robust and superior performance are demonstrated over tandem schemes which suffer from a high BER performance.

1.4 Thesis Outline

The thesis is organized as follows.

In Chapter 2, we provide the basic concepts of Turbo codes. After first introducing recursive systematic convolutional encoders, we describe the general structures of Turbo code encoders and decoders. A brief description of the BCJR algorithm [9] used in the Turbo code decoder is then given, along with explanations on the iterative decoding procedure.

In Chapter 3, we investigate the issue of designing a *channel optimized vector quantization* (COVQ) system for Turbo-coded AWGN and Rayleigh fading channels used in conjunction with BPSK modulation. We begin by introducing the concepts of vector quantization and COVQ, and then depict the system under consideration. To incorporate Turbo codes in the COVQ design, we provide two discrete channel models to describe the concatenation of Turbo code encoder, BPSK-modulator, physical channel, Turbo code decoder, as well as a q-bit soft-decision quantizer. The q-bit soft-decision quantizer is designed so that the capacity of the overall discrete channel model is maximized. A COVQ is then designed for these models, and its performance is compared with that of the original COVQ design, a tandem scheme, as well as a recent scheme proposed by Ho [48].

In Chapter 4, we consider designing Turbo codes for non-uniform i.i.d. sources. The encoder used in this chapter has the standard *systematic* structure. Besides an easy modification in the Turbo code decoder's extrinsic information, the encoder structure is optimized for a given source distribution via a simple iterative search procedure. Necessary and sufficient conditions for the asymptotic uniformity of the encoder states are also shown. Simulation results are provided and compared to the Shannon limit.

Although the gains achieved in Chapter 4 are considerably significant with respect to the original Berrou code, their performance gaps vis-a-vis the Shannon limit are still relatively large for heavily biased sources. In Chapter 5, an analysis on the encoder output reveals that the drawback lies in the *systematic* structure, which results in a mismatch between the biased distribution of the systematic bit stream and the uniform input distribution needed to achieve the channel capacity. We prove that when some constraints are satisfied, recursive non-systematic convolutional (RNSC) encoders can generate asymptotically uniformly distributed outputs even for extremely biased sources. Therefore, we propose using RNSC encoders as the constituent Turbo encoders. With a properly modified decoder and an optimized encoder for a given source distribution, substantial gains over systematic Turbo codes are demonstrated. The gaps to the Shannon limit for heavily biased sources are hence significantly reduced. Finally, we compare our joint source-channel coding system with two tandem schemes that employ a nearly optimal source code followed by Turbo codes.

In Chapter 6, we investigate the design of Turbo codes for Markov sources. We provide a modified BCJR algorithm in the first constituent decoder to incorporate the Markovian source property. A weighted correction term is also introduced to the extrinsic information generated by the second constituent decoder. Limitations on achievable performance with our system are examined. Significant gains over the original Turbo codes are shown in simulation results. Finally, performance comparisons with tandem schemes are also provided.

Finally, in Chapter 7, the thesis is summarized and future research directions are discussed.

Chapter 2

Turbo Codes Basics

In this chapter, we briefly provide the basics of Turbo codes as originally introduced by Berrou *et al.* [14, 15]. In particular, we first introduce the so-called recursive systematic convolutional encoders, which are used as the constituent encoders in Turbo codes. We then describe the structures of a Turbo code encoder and decoder, and describe the BCJR algorithm which is used in the iterative decoding process.

2.1 Recursive Systematic Convolutional Encoder

The constituent encoders of a Turbo code are *recursive systematic* convolutional encoders. A typical structure of such an encoder is depicted in Fig. 2.1. The number of shift registers, M, is called the *memory* of the encoder. Here we have $M = 4$. The tap coefficients of the feedback part are denoted as $\{f_i\}_{i=0}^{M}$, and the tap coefficients

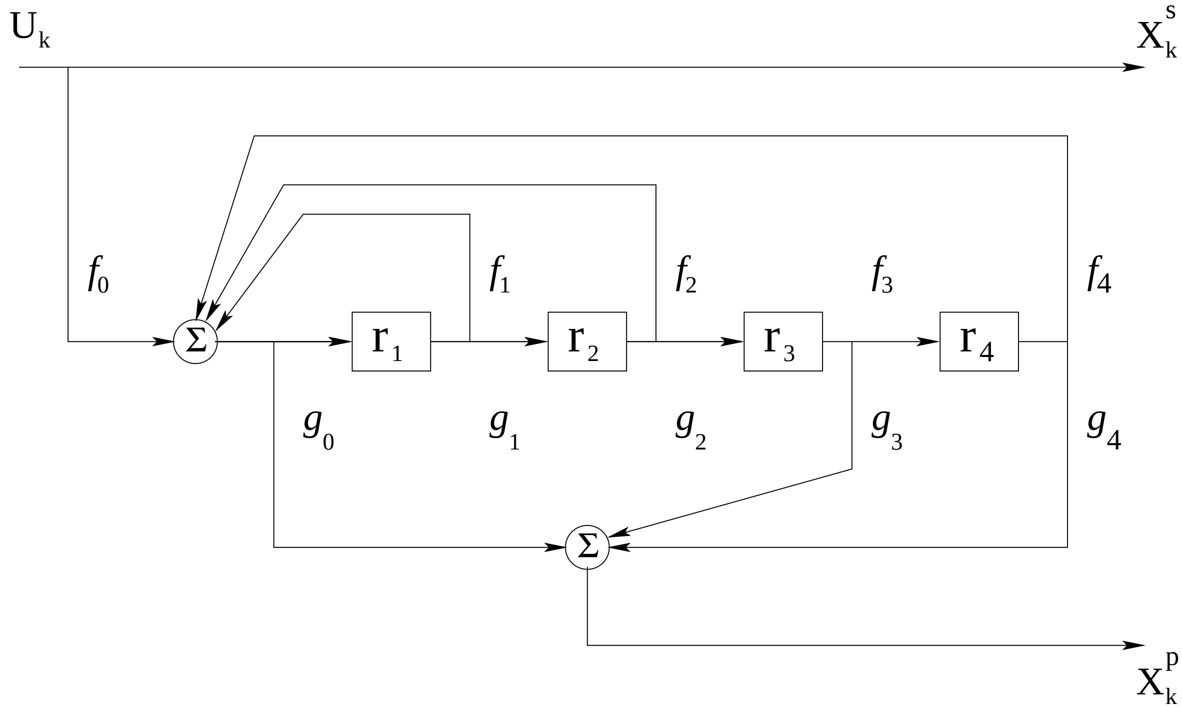

Figure 2.1: Recursive systematic convolutional encoder.

of the feed-forward part as $\{g_i\}_{i=0}^{M}$. Both $\{f_i\}_{i=0}^{M}$ and $\{g_i\}_{i=0}^{M}$ are binary, where a "1" means a circuit connection and a "0" otherwise. In many literatures, convolutional encoders are denoted in octal form. For example, as in Fig. 2.1, we have $\{f_0,\ f_1,\ f_2,\ f_3,\ f_4\}$=$\{1,\ 1,\ 1,\ 0,\ 1\}$ in binary, in octal form it is 35. Similarly we have the feed-forward tap coefficients $\{g_0,\ g_1,\ g_2,\ g_3,\ g_4\}$=$\{1,\ 0,\ 0,\ 1,\ 1\}$, and in octal form it is 23. Thus the encoder structure in Fig. 2.1 is often denoted as (35, 23).

As *systematic* implies, one of the outputs, X_k^s, is identical to the input U_k. The other output X_k^p is called the parity output. The relation between the input U_k and the parity output X_k^p is given by

$$\begin{aligned} V_k &= U_k + \sum_{i=1}^{M} f_i V_{k-i}, \\ X_k^p &= \sum_{i=0}^{M} g_i V_{k-i}, \end{aligned}$$

where the summations are modulo-2.

The *encoder state* is defined by the content of its shift registers. Before the information sequence $\{U_k\}_{k=1}^{L}$ is fed into the encoder, it is assumed that each register contains a "0". In this case, we say that the encoder is in the all-zero state. The encoder depicted in Fig. 2.1 has in total 16 possible states: 0000, 0001, 0010, ..., 1111. At time k, before U_k is fed into the encoder, we say that the encoder is in state S_{k-1}, and after the data-shifting is completed, the state is S_k. Both S_k and the parity output are determined by the input U_k and the state S_{k-1}. After the whole information sequence is encoded, to make the decoding process easier, the encoder is usually terminated at the all-zero state. For a convolutional encoder with memory M, this would require M extra tail bits. If the encoder is non-recursive, i.e., it has no feedback part, M consecutive zeros can flush the encoder to the all-zero state. For recursive convolutional encoders, each tail bit, either a "1" or a "0", is determined by the encoder state at that moment. The extra state-terminating tail bits may impose a bandwidth waste problem; however, when the sequence length is long, the effect is negligible.

2.2 Turbo Code Encoder

Fig. 2.2 illustrates the structure of a Turbo code encoder. A Turbo code encoder consists of two systematic recursive convolutional encoders in parallel concatenation and linked by an interleaver. In general, there can be more than two constituent encoders and they do not have to be identical.

The role of the interleaver used in the Turbo code encoder is to re-arrange the information sequence into a different order. In the Turbo codes literature, the most commonly used interleavers are mainly of three types: uniform interleavers, random interleavers, and pseudo-random interleavers. A uniform interleaver is a rectangular matrix, with the data written in row by row and read out column by column. A random interleaver, on the other hand, performs a permutation on the input sequence in a purely random fashion. After random interleaving, sometimes neighboring bits may still be permuted into neighboring bits; to eliminate this possibility, an S-random interleaver may be used. An S-random interleaver is a special type of random interleaver with the condition that all neighboring bits are permuted randomly into positions no closer than the spread S. The drawbacks with S-random interleavers are that their computer generation may require a long time, and sometimes when S is chosen to be too large, they may not be generated successfully. For a given input sequence, a pseudo-random interleaver permutes each bit according to a deterministic function, however, the permuted sequence appears to the decoder quite random-like.

Both random and pseudo-random interleavers offer superior performance than that of uniform interleavers. Therefore, in this thesis, we mainly use a pseudo-random interleaver described in [15] and adopt an S-random interleaver when necessary.

For a Turbo code encoder, the data flow goes as follows. The information sequence $\{U_k\}_{k=1}^{L}$ is fed directly into the first constituent encoder, which produces a parity sequence $\{X_k^{1p}\}_{k=1}^{L}$. At the same time, $\{U_k\}_{k=1}^{L}$ is permuted by the interleaver into a different ordered sequence $\{\tilde{U}_k\}_{k=1}^{L}$, which is then encoded by the second constituent encoder, generating a second parity sequence $\{X_k^{2p}\}_{k=1}^{L}$. The other output sequence $\{X_k^{s}\}_{k=1}^{L}$ is identical to the input $\{U_k\}_{k=1}^{L}$. Originally the second constituent systematic convolutional encoder has another output sequence $\{\tilde{X}_k^{s}\}_{k=1}^{L}$, which is identical to $\{\tilde{U}_k\}_{k=1}^{L}$; since it is nothing but a permuted version of $\{U_k\}_{k=1}^{L}$, there is no need to waste the bandwidth to transmit both systematic sequences. Thus corresponding to each input bit U_k, there are 3 output bits, and hence the rate of this encoder is 1/3 bits per information symbol. When the desired rate $R_c > 1/3$, a puncturing device is used to delete some of the parity bits. For example, for $R_c = 1/2$, we can puncture all the odd bits of $\{X_k^{1p}\}$, and all the even bits of $\{X_k^{2p}\}$.

As mentioned in the previous section, at the end of the information sequence, M extra tail bits are appended to terminate the state of the first constituent encoder. Due to interleaving, it is very unlikely that the second decoder can be terminated at the same time. For short sequence lengths and encoders with a small number of

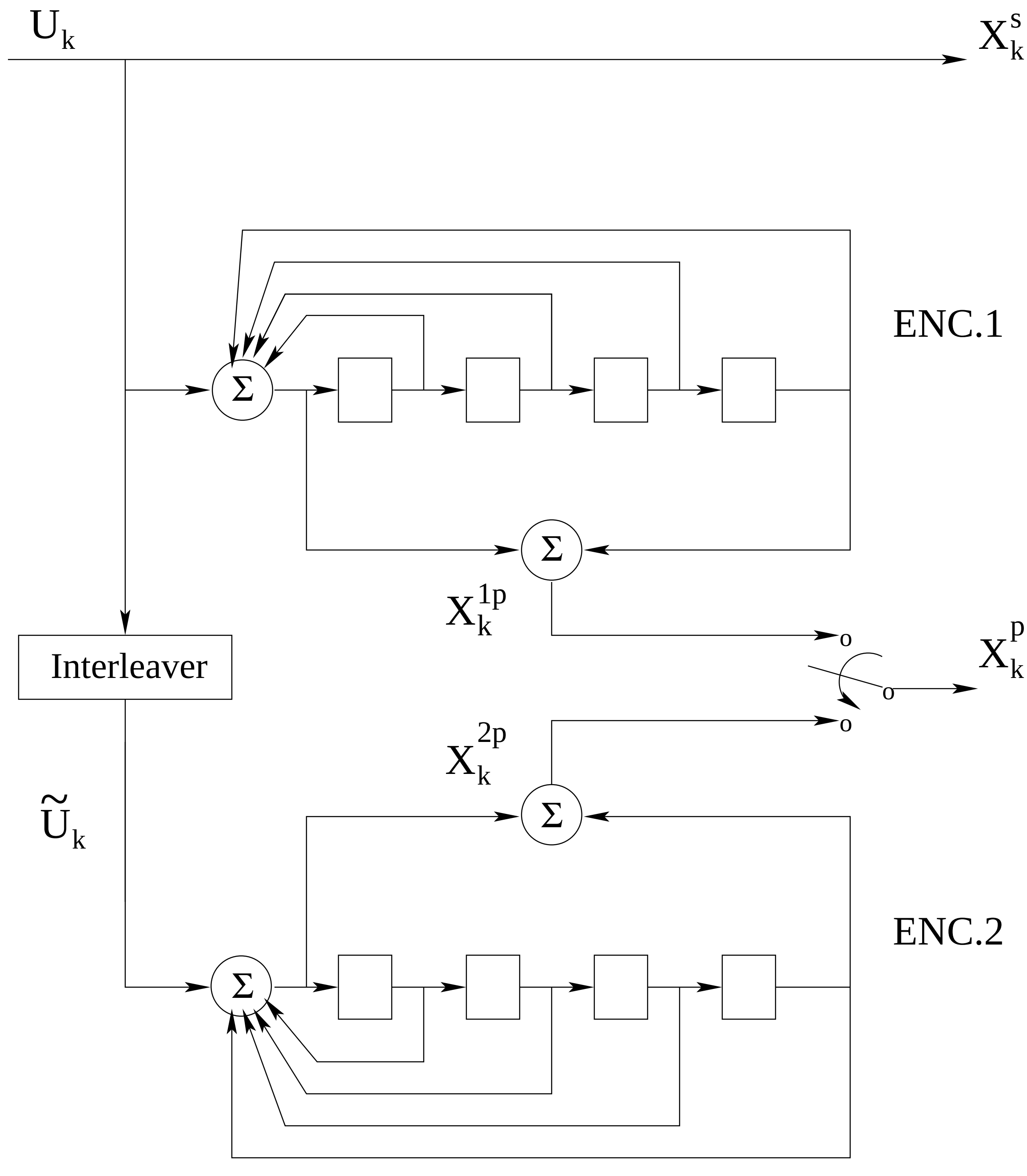

Figure 2.2: Turbo code encoder structure.

memory M, specially designed interleavers can guarantee both encoder states to be terminated to the all-zero state. For example, in [12], the sequence length is 420 bits and the memory is 2. However, such design may not be easily extended to long sequence lengths and encoders with large memory. Therefore, the second encoder state is usually not terminated.

2.3 Turbo Code Decoder and Iterative Decoding

Before introducing the structure of the Turbo code decoder, we first clarify the notation for the data before and after the transmission over the channel. As shown in Fig. 2.3, the channel inputs are the systematic bit X_k^s and the parity bit X_k^p, which are obtained from X_k^{1p} or X_k^{2p} via a puncturing device. After the noise distortion is introduced through transmission over the channel, we denote the systematic bit by Y_k^s, and the parity bit by Y_k^p, which is then separated into Y_k^{1p} and Y_k^{2p} by a switch. For the sake of simplicity in notation, we denote the pair (X_k^s, X_k^p) as X_k, and the pair (Y_k^s, Y_k^p) as Y_k in the analysis throughout the thesis. When puncturing is used to achieve high rates, some parity bits are not available. Furthermore, the sequence $\{Y_k\}_{k=1}^L$ is denoted as Y_1^L for brevity.

The structure of a Turbo code decoder is depicted in Fig. 2.4. Corresponding to the two parallel concatenated constituent encoders, a Turbo code decoder has two constituent decoders in serial concatenation. The two constituent decoders are

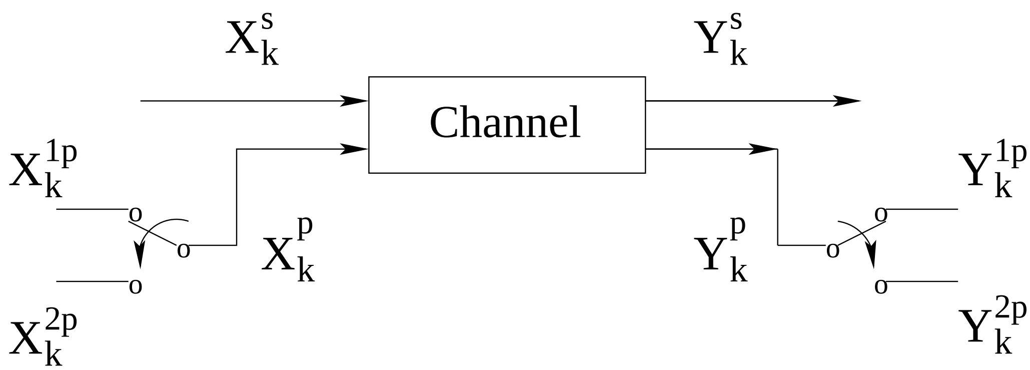

Figure 2.3: Data before and after transmission.

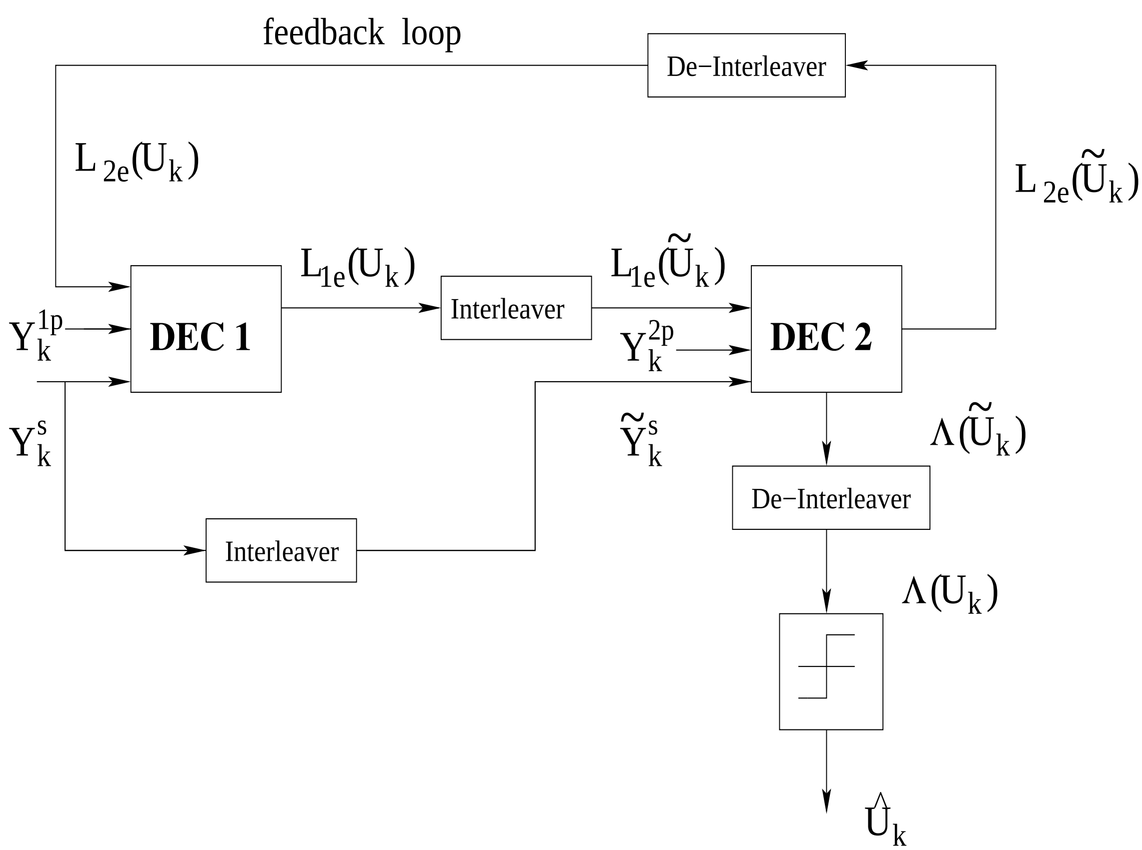

Figure 2.4: Turbo code decoder structure.

separated by an interleaver, which is identical to the one used in the encoder. In the feedback loop from the output of the second constituent decoder to the input of the first constituent decoder, a de-interleaver is used to permute the sequence back into the original order.

To decode $\{U_k\}_{k=1}^L$, based on the observation of the whole received sequence Y_1^L, each constituent decoder evaluates the conditional *log-likelihood ratio (LLR)* for every bit U_k defined by

$$\Lambda(U_k) = \log \frac{Pr\{U_k = 1 | Y_1^L = y_1^L\}}{Pr\{U_k = 0 | Y_1^L = y_1^L\}}$$

via the so-called BCJR algorithm [9], which is a symbol-by-symbol maximum *a posteriori* (MAP) decoding algorithm. We next describe the Turbo decoding method given by Robertson [63].

In the BCJR algorithm, by considering the encoder state S_k associated with each information bit U_k, the conditional LLR is factorized into

$$\Lambda(U_k) = \log \frac{\sum_s \sum_{s'} \gamma(y_k, 1, s|s')\alpha_{k-1}(s')\beta_k(s)}{\sum_s \sum_{s'} \gamma(y_k, 0, s|s')\alpha_{k-1}(s')\beta_k(s)},$$

where

$$\begin{aligned} \alpha_k(s) &\triangleq Pr\{S_k = s | Y_1^k = y_1^k\}, \\ \beta_k(s) &\triangleq \frac{Pr\{Y_{k+1}^L = y_{k+1}^L | S_k = s\}}{Pr\{Y_{k+1}^L = y_{k+1}^L | Y_1^k = y_1^k\}}, \\ \gamma(y_k, i, s|s') &\triangleq Pr\{Y_k = y_k, U_k = i, S_k = s | S_{k-1} = s'\}. \end{aligned}$$

$\alpha_k(s)$ and $\beta_k(s)$ can be computed recursively using the following relations:

$$\begin{aligned}
\alpha_k(s) &= \frac{\sum_{s'} \sum_{i=0}^{1} \gamma(y_k, i, s|s')\alpha_{k-1}(s')}{\sum_s \sum_{s'} \sum_{i=0}^{1} \gamma(y_k, i, s|s')\alpha_{k-1}(s')}, \\
\beta_k(s) &= \frac{\sum_{s'} \sum_{i=0}^{1} \gamma(y_{k+1}, i, s'|s)\beta_{k+1}(s')}{\sum_s \sum_{s'} \sum_{i=0}^{1} \gamma(y_{k+1}, i, s'|s)\alpha_k(s)}.
\end{aligned}$$

We can see that $\alpha_k(s)$ is expressed in a forward recursion, while $\beta_k(s)$ is computed in a backward manner. Hence the BCJR algorithm is also known as the *forward-and-backward* algorithm. Like any recursive methods, we need to have some boundary conditions to solve $\alpha_k(s)$ and $\beta_k(s)$. For the first constituent encoder, since it starts and terminates at the all-zero state, we have

$$\alpha_0(0) = 1, \quad \text{and} \quad \alpha_0(s) = 0 \ \ \forall\ s \neq 0,$$

$$\beta_L(0) = 1, \quad \text{and} \quad \beta_L(s) = 0 \ \ \forall\ s \neq 0.$$

For the second constituent encoder, it starts at the all-zero state, but its trellis is left "open" at the end. Therefore, the boundary conditions for the second constituent encoder are

$$\alpha_0(0) = 1, \quad \text{and} \quad \alpha_0(s) = 0 \ \ \forall\ s \neq 0,$$

$$\beta_L(s) = \frac{1}{2^M}, \qquad \forall\ s = 0, 1, \cdots, 2^M - 1.$$

To compute $\alpha_k(s)$ and $\beta_k(s)$, at each step we need to compute $\gamma(\cdot,\cdot,\cdot|\cdot)$ first, which can be factorized into

$$\begin{aligned}\gamma(y_k, i, s|s') &= p(Y_k^s = y_k^s|U_k = i, S_k = s, S_{k-1} = s') \\ &\cdot p(Y_k^p = y_k^p|U_k = i, S_k = s, S_{k-1} = s') \\ &\cdot Pr\{U_k = i|S_k = s, S_{k-1} = s'\} \cdot Pr\{S_k = s|S_{k-1} = s'\}. \quad (2.1)\end{aligned}$$

For the systematic bit, conditioning on the states is not necessary; therefore

$$p(Y_k^s = y_k^s|U_k = i, S_k = s, S_{k-1} = s') = p(Y_k^s = y_k^s|U_k = i),$$

which can be described by the probability density function (pdf) of the channel noise. For the parity bit Y_k^p, either $(U_k = i,\ S_k = s)$ or $(U_k = i,\ S_{k-1} = s')$ is sufficient to yield $X_k^p = j$, where $j = 0,\ 1$, and thus this channel transition probability can also be computed. Denote the third factor in (2.1) as

$$q(i|s, s') \triangleq Pr\{U_k = i|S_k = s, S_{k-1} = s'\}.$$

Since given the state transition from S_{k-1} to S_k, the probability of $U_k = i$ can only be either 1 or 0. Therefore $q(i|s, s')$ is actually deterministic. Finally, the conditional probability $Pr\{S_k = s|S_{k-1} = s'\}$ is essentially the *a priori* probability of the information bit U_k which causes this state transition. That is

$$Pr\{S_k = s|S_{k-1} = s'\} = Pr\{U_k = i\}$$

with i satisfying

$$q(i|s, s') = 1.$$

Furthermore, by introducing the *a priori* LLR for U_k defined by

$$L_{ap}(U_k) = \log \frac{Pr\{U_k = 1\}}{Pr\{U_k = 0\}},$$

we can express the state transition probability as

$$\begin{aligned}
Pr\{S_k = s | S_{k-1} = s'\} &= \frac{e^{L_{ap}(U_k)}}{1 + e^{L_{ap}(U_k)}} \quad \text{when} \quad q(1|s, s') = 1, \\
Pr\{S_k = s | S_{k-1} = s'\} &= \frac{1}{1 + e^{L_{ap}(U_k)}} \quad \text{when} \quad q(1|s, s') = 0.
\end{aligned}$$

For memoryless sources and memoryless channels, the LLR can be decomposed into three separate terms:

$$\Lambda(U_k) = L_{ch}(U_k) + L_{ex}(U_k) + L_{ap}(U_k), \tag{2.2}$$

where

$$\begin{aligned}
L_{ch}(U_k) &= \log \frac{Pr\{Y_k^s = y_k^s | U_k = 1\}}{Pr\{Y_k^s = y_k^s | U_k = 0\}}, \\
L_{ex}(U_k) &= \log \frac{\sum_s \sum_{s'} \bar{\gamma}(y_k^p, 1, s|s') \alpha_{k-1}(s') \beta_k(s)}{\sum_s \sum_{s'} \bar{\gamma}(y_k^p, 0, s|s') \alpha_{k-1}(s') \beta_k(s)}, \\
L_{ap}(U_k) &= \log \frac{Pr\{U_k = 1\}}{Pr\{U_k = 0\}},
\end{aligned}$$

where

$$\bar{\gamma}(y_k^p, i, s|s') = Pr\{Y_k^p = y_k^p | U_k = i, S_k = s\} \cdot Pr\{U_k = i | S_k = s, S_{k-1} = s'\}.$$

The three terms in (2.2) are called the channel transition term, the extrinsic information term, and the *a priori* term, respectively.

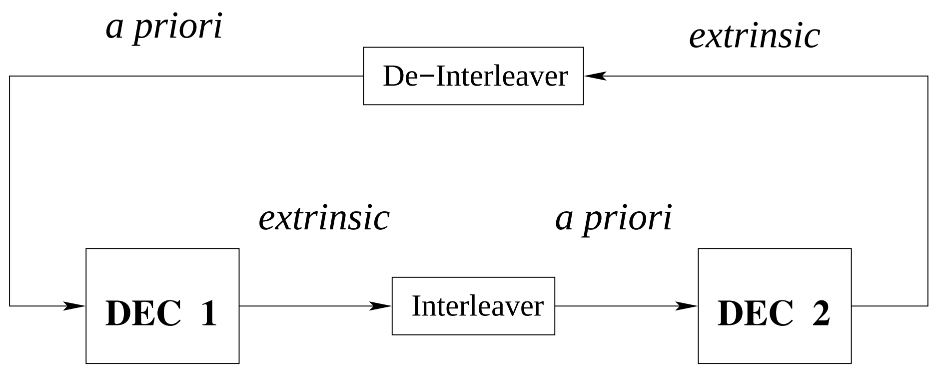

Figure 2.5: Information exchange between the two decoders.

There are also other alternative decoding algorithms such as the soft output Viterbi algorithm (SOVA) [43, 45] and the Log-Max algorithm [64] proposed to reduce the decoding complexity at the expense of some performance degradation. In this thesis, we only employ the BCJR algorithm due to its excellent performance.

As shown in Fig. 2.4, the Turbo decoding process is realized in an iterative fashion. At the first iteration, the first decoder takes two input sequences: $\{Y_k^s\}_{k=1}^L$ and $\{Y_k^{1p}\}_{k=1}^L$, and produces the LLR of each bit as the output. Among the LLR $\Lambda(U_k)$, only the extrinsic information term $L_{ex}(U_k)$ is passed on to the second decoder via the interleaver. The second decoder then has three input sequences: $\{\tilde{Y}_k^s\}_{k=1}^L$, $\{Y_k^{2p}\}_{k=1}^L$, and $\{L_{ex}(\tilde{U}_k)\}_{k=1}^L$, and it passes on the extrinsic information part of its LLR evaluation back to the first decoder via the de-interleaver. Therefore, from the second iteration on, both constituent decoders have three input sequences.

The extrinsic information produced by one decoder is used as the *a priori* information for the other decoder. This is illustrated in Fig. 2.5. The reasons that only the extrinsic information term is utilized in the iterative decoding process are as follows. First, the channel transition term $L_{ch}(U_k)$ is a direct estimate of the uncoded systematic bit; therefore it is not always reliable. If this term is passed on to the next decoder, it can be shown that the LLR computed by the next decoder becomes

$$\Lambda(U_k) = 2L_{ch}(U_k) + L_{ex}(U_k) + L_{ap}(U_k).$$

That is, the channel transition term is doubled. As a result, the extrinsic information term becomes less significant in correcting the error. After a few iterations, $L_{ch}(U_k)$ gets accumulated, and eventually becomes the dominant term, rendering the whole iterative decoding process meaningless. Second, if the *a priori* term is used in the iterative process, then a constituent decoder would take its own extrinsic information estimation from the previous iteration as part of the input. The new estimation would therefore be biased since the errors made in the previous iteration would likely reappear. Therefore, both $L_{ch}(U_k)$ and $L_{ap}(U_k)$ should be eliminated as part of the input to the other constituent decoder. Thus, both constituent decoders produce their own LLR estimation based upon the extrinsic information exchanged between them in an iterative manner. The reliability of their estimations is improved after each update, hence leading towards a greatly enhanced performance.

The most substantial coding gain is achieved by the first few iterations. As the number of iterations increases, the improvement becomes less significant. Several stopping criteria are proposed to terminate the iterative process when the improvement is negligible [71], which can practically save considerable amount of unnecessary computations. However, in most of the Turbo codes literature, a predetermined fixed number of iterations is usually adopted for fair comparisons. At the end of the iterative decoding, a hard decision is made based on the last LLR estimation:

$$\hat{U}_k = 1 \quad \text{if} \quad \Lambda(U_k) \geq 0,$$

$$\hat{U}_k = 0 \quad \text{if} \quad \Lambda(U_k) < 0.$$

Chapter 3

Soft-Decision COVQ Design Based on Turbo Codes

3.1 Introduction

The goal in source coding is to eliminate as much source redundancy as possible. Source coding can be classified into two categories: *lossless* and *lossy*. In lossless source coding, the source output is compressed in such a way that the decoded data is identical to the original one. Lossy source coding, on the other hand, achieves higher compression rate by sacrificing a tolerable amount of distortion between the recovered and the original data. Quantization performs a mapping from continuous signals or a large set of discrete signals into a pre-determined set (with relatively small size) of symbols. Therefore, quantization falls into the category of lossy source coding.

The vector quantization originated by Linde, Buzo and Gray [52] is one of the most notable techniques in fixed-length source coding, and has been successfully applied to speech [55] and image coding [37]. The channel-optimized vector quantization (COVQ) studied by Kumazawa *et al.* [50] and Farvardin *et al.* [29, 28] incorporates the channel statistics in the vector quantization design, thus achieves high end-to-end performance over noisy channels.

In this chapter, we design and implement a robust soft-decision COVQ scheme for Turbo-coded AWGN channels and Rayleigh fading channels with known side information. More specifically, we employ the methods recently introduced in [2, 62] to design a COVQ system that improves the end-to-end performance of a Turbo-coded system by exploiting the log-likelihood ratio (LLR) generated by the Turbo decoder. This is achieved via the use of a q-bit scalar soft-decision demodulator at the output of the Turbo decoder, and by designing a COVQ scheme for the resulting *expanded discrete channel* which consists of the concatenation of the Turbo encoded and decoded channel with the soft-decision demodulator. Alternative approaches for channel-optimized quantization using Turbo codes have been previously studied by Bakus and Khandani for scalar quantization [10, 11], and by Ho for vector quantization [48], where the *entire* (unquantized) soft-decision information provided by the

LLR of the Turbo decoder is utilized. Our scheme offers better performance than Ho's system; furthermore, the decoding complexity of our system is lower but the storage requirement might be higher.

3.2 Vector Quantization

A vector quantizer Q of dimension k and size N is a mapping from a vector in the k-dimensional Euclidean space, $\mathbb{R}^k$, into a finite set $\mathcal{C}$ containing N output or reproduction points, called *code vectors* or *codewords* [37]. That is,

$$Q : \mathbb{R}^k \to \mathcal{C},$$

where $\mathcal{C} = (\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_N)$ and $\mathbf{y}_i \in \mathbb{R}^k$ for each $i \in \mathcal{I} \stackrel{\triangle}{=} \{1,\ 2,\ \cdots,N\}$. The set $\mathcal{C}$ is called the *codebook*. The rate of the quantizer is defined as

$$R_s = \frac{1}{k} \log_2 N \quad \text{bits/source sample.}$$

For a given N-point vector quantizer, $\mathbb{R}^k$ is partitioned into N regions, which are defined as

$$S_i = \{\mathbf{x} \in \mathbb{R}^k : Q(\mathbf{x}) = \mathbf{y}_i\}, \qquad i = 1, 2, \cdots, N,$$

and they satisfy

$$\bigcup_i S_i = \mathbb{R}^k \quad \text{and} \quad S_i \bigcap S_j = \phi \quad \forall \;\; i \neq j.$$

$\mathcal{P} = \{S_1, S_2, \cdots, S_N\}$ is called the partition set.

A vector quantizer can be decomposed into two component operations, the vector encoder $\mathcal{E}$ and the vector decoder $\mathcal{D}$, which are described by the following mappings:

$$\mathcal{E} : \mathbb{R}^k \to \mathcal{I} \quad \text{and} \quad \mathcal{D} : \mathcal{I} \to C.$$

To assess the performance of a vector quantizer, a *distortion* measure needs to be defined. The most convenient and widely used distortion measure between an input vector $\mathbf{X}$ and a quantized vector $\hat{\mathbf{X}} = Q(\mathbf{X})$ is the *squared error* (or the squared Euclidean distance), which is defined as

$$\begin{aligned} d(\mathbf{X}, \hat{\mathbf{X}}) &= ||\mathbf{X} - \hat{\mathbf{X}}||^2 \\ &= \sum_{i=1}^{k} (X_i - \hat{X}_i)^2. \end{aligned}$$

Suppose the source generates random vectors according to its probability density function (pdf) $f(\mathbf{x})$, then the *average distortion per sample* of a given vector quantizer is defined as

$$\begin{aligned} D &= \frac{1}{k} E d(\mathbf{X}, \hat{\mathbf{X}}) = \frac{1}{k} \int_{\mathbb{R}^k} d(\mathbf{x}, Q(\mathbf{x})) f_{\mathbf{X}}(\mathbf{x}) d\mathbf{x} \\ &= \frac{1}{k} \sum_{i=1}^{N} \int_{S_i} d(\mathbf{x}, Q(\mathbf{x})) f_{\mathbf{X}}(\mathbf{x}) d\mathbf{x}. \end{aligned} \tag{3.1}$$

The goal of designing a vector quantizer with given dimension k and size N is to find the optimal partition set $\mathcal{P}^*$ and the optimal codebook $\mathcal{C}^*$ such that the average distortion per sample defined in (3.1) is minimized. To achieve this goal, two necessary conditions are considered. The first one is called the *nearest neighbor condition*, which states that for a given codebook $\mathcal{C}$, the partition set $\mathcal{P}$ that minimizes the average distortion must satisfy

$$S_i = \{\mathbf{x} : d(\mathbf{x}, \mathbf{y}_i) \leq d(\mathbf{x}, \mathbf{y}_j), \quad \forall\, j \neq i\}. \tag{3.2}$$

The second condition is known as the *centroid condition*: for a given partition set $\mathcal{P}$, the optimal codebook is described by

$$\mathbf{y}_i = E\left[\mathbf{X} | \mathbf{X} \in S_i\right] = \frac{\int_{S_i} \mathbf{x} f(\mathbf{x}) d\mathbf{x}}{\int_{S_i} f(\mathbf{x}) d\mathbf{x}}. \tag{3.3}$$

In general, it may not be easy to find the *optimal* vector quantizer that minimizes the average distortion. However, a locally optimal solution of the above problem was proposed in [52] by Linde, Buzo and Gray by using an iterative algorithm (LBG algorithm). The basic idea of the LBG algorithm is, for a given number of quantization level N, first choose an initial codebook $\mathcal{C}^{(0)}$, use the nearest neighbor condition (3.2) to find the optimal partition $\mathcal{P}^{(1)}$; then for this new partition $\mathcal{P}^{(1)}$, find the optimal codebook $\mathcal{C}^{(1)}$ by using the centroid condition (3.3). The average distortion decreases as this iterative procedure continues. When the difference between the average distortions of two successive steps is less than a pre-determined small threshold ϵ, the iteration is terminated, yielding a locally optimal vector quantizer, which is often referred to as the LBG vector quantizer (LBG-VQ).

3.3 Channel Optimized Vector Quantization

Vector quantization is an efficient and powerful lossy data compression scheme. However, it does not take the channel noise distortion into consideration. As a result, when the index of a vector-quantized signal is transmitted over a noisy channel, at the receiver end, a wrong index may be used for decoding due to the channel noise corruption. Channel optimized vector quantization (COVQ) is an extension of the vector quantization by incorporating the channel statistics into the design. A COVQ system consists of three operations: the encoder mapping, the channel index assignment mapping and the decoder mapping. Consider a source that emits a real-valued, stationary and ergodic sequence with zero mean and unit variance. The COVQ encoder is a k-dimensional N-level vector quantizer, described by the encoder mapping $\mathcal{E} : \mathbb{R}^k \to \mathcal{I}$ according to

$$\mathcal{E}(\mathbf{X}) = i, \quad \text{if} \quad X \in S_i, \quad \forall\, i \in \mathcal{I}.$$

The channel index assignment is a one-to-one mapping $b : \mathcal{I} \to \mathcal{I}$ such that for every COVQ encoder output i, an index i' is chosen by

$$i' = b(i) \in \mathcal{I},$$

and then i' is transmitted through the noisy discrete memoryless channel (DMC), whose transitional probability is $P(j|i')$, $\forall\ i', j \in \mathcal{I}$. Finally, the COVQ decoder performs the mapping $\mathcal{D} : \mathcal{I} \to \mathcal{C}$ according to

$$\mathcal{D}(j) = \mathbf{y}_j.$$

Correspondingly, the *average distortion per sample* is re-defined as

$$D((\mathcal{P}, \mathcal{C}); b) \triangleq \frac{1}{k} \sum_{i=1}^{N} \sum_{j=1}^{N} P(j|b(i)) \int_{S_i} f(\mathbf{x}) d(\mathbf{x}, \mathbf{y}_j) d\mathbf{x}. \tag{3.4}$$

The problem of minimizing the average distortion is identical to the VQ design problem with a modified distortion measure, and the two necessary conditions are modified accordingly as follows.

1) For a fixed codebook $\mathcal{C}$ and a fixed index mapping b, the optimal partition is given by

$$S_i = \left\{ \mathbf{x} : \sum_{j=1}^{N} P(j|b(i)) d(\mathbf{x}, \mathbf{y}_j) \leq \sum_{j=1}^{N} P(j|b(l)) d(\mathbf{x}, \mathbf{y}_j),\ \forall l \in \mathcal{I} \right\}; \tag{3.5}$$

2) For a fixed partition $\mathcal{P}$ and a fixed index mapping b, the optimal codebook $\mathcal{C}$ is described by

$$c_j = \frac{\sum_{i=1}^{N} P(j|b(i)) \int_{S_i} \mathbf{x} f(\mathbf{x}) d\mathbf{x}}{\sum_{i=1}^{N} P(j|b(i)) \int_{S_i} f(\mathbf{x}) d\mathbf{x}}. \tag{3.6}$$

3.4 System Structure

We now consider the following system (see Figure 3.1). The COVQ encoder takes a k-dimensional real vector $\mathbf{X}$ as its input, operates at a rate of R_s bits per source sample, and generates kR_s bits as the output $\mathbf{U} \in \{0,1\}^{kR_s}$. This output is then fed into a Turbo code encoder, which is of rate R_c information bits/code bit. From the Turbo encoder, the output V is binary phase-shift keying (BPSK) modulated as $\mathbf{W} \in \{-1,+1\}^{kR_s/R_c}$ (assuming kR_s/R_c is an integer). The sequence $\{\mathbf{W}_i\}$ is then transmitted through an additive white Gaussian noise (AWGN) channel or a Rayleigh fading channel described by

$$Z_l = A_l W_l + N_l, \qquad l = 1, 2, 3, \cdots,$$

where $W_l \in \{-1,+1\}$ is the BPSK signal of unit energy and $\{N_l\}$ is an i.i.d. Gaussian noise with zero mean and variance $N_0/2$. For the AWGN channel, $A_l = 1$, $l = 1, 2, 3, \cdots$, while for the Rayleigh fading channel, the fading envelope $\{A_l\}$ (also known as the channel state information or channel side information) is assumed to be i.i.d. and Rayleigh distributed. We assume that $\{A_l\}$ is known at the decoder, and that A_l, W_l, and N_l are independent of each other.

At the receiver end, Turbo decoding is used to provide the LLR given by

$$\Lambda_l = \log \frac{Pr\{U_l = 1|\mathbf{Z}\}}{Pr\{U_l = 0|\mathbf{Z}\}}, \qquad l = 1, 2, 3, \cdots,$$

which is then demodulated via a q-bit uniform scalar quantizer $\alpha(\cdot)$ with quantization step Δ. The quantizer is described by

$$\alpha(\Lambda) = j \quad \text{if} \quad \Lambda \in (T_{j-1}, T_j),$$

where $j = 0, 1, \cdots, 2^q - 1$.

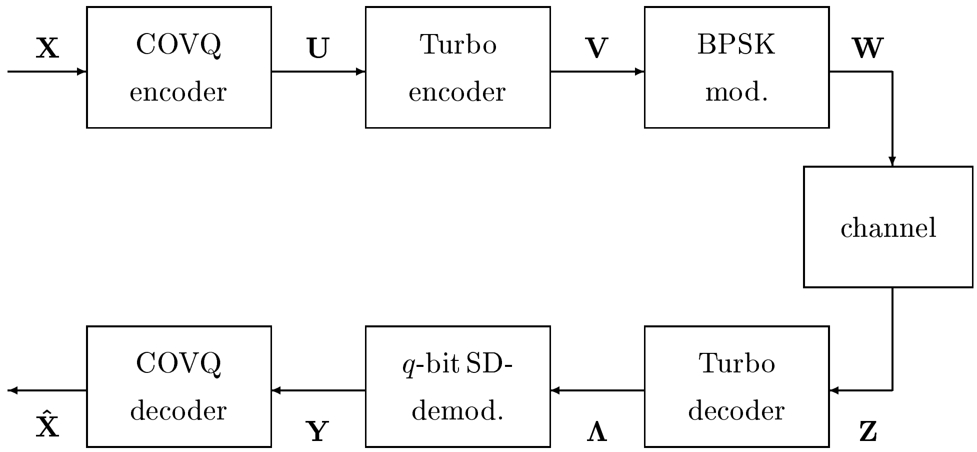

Figure 3.1: Block diagram of the system.

The thresholds $\{T_j\}$ are uniformly spaced with quantization step Δ; they satisfy

$$T_j = \begin{cases} -\infty & \text{if } \; j = -1, \\ (j+1-2^{q-1})\Delta, & \text{if } \; j = 0, 1, \cdots, 2^q - 2, \\ +\infty & \text{if } \; j = 2^q - 1. \end{cases}$$

Finally, these qkR_s bits are passed to the COVQ decoder, from which $\hat{\mathbf{X}}$, an estimation of $\mathbf{X}$, is produced.

Note that in the q-bit scalar quantization part, when $q = 1$, the whole system is equivalent to a hard-decision COVQ system; when q increases, it approaches the system that exploits the *entire* soft-decision information provided by the LLR of the Turbo code decoder.

As justified by Shannon's separation principle, the source coding part and the Turbo coding part can be designed independently, resulting in the so-called tandem scheme. For example, the system consists of a LBG-VQ designed for noiseless channels and followed by a regular Turbo code. Obviously, such tandem schemes would practically be unable to achieve optimality, since the LBG-VQ assumes that all channel errors can be corrected by the Turbo code, which is not the case in reality. Here we propose a joint source-channel coding scheme by designing the COVQ for a discrete Turbo-coded channel model.

3.5 DMC Model for Turbo-Coded AWGN Channels

We model the concatenation of the Turbo encoder, BPSK modulator, AWGN channel, Turbo decoder, and q-bit soft-decision demodulator as an equivalent expanded discrete memoryless channel (DMC) model. This expanded DMC is of 2^{kR_s}-input and 2^{qkR_s}-output. This channel can be regarded as a binary-input, 2^q-output DMC used kR_s times. In this case, the channel input alphabet is $\mathcal{U}=\{0,1\}$ and the channel output alphabet is given by $\mathcal{Y}=\{0,1,\cdots,2^q-1\}$. To design a COVQ scheme for this Turbo-coded DMC model, we need to determine its transitional probability matrix

$$\Pi = [\pi_{ij}], \quad i \in \mathcal{U},\ j \in \mathcal{Y}.$$

where

$$\pi_{ij} \triangleq P(Y=j|U=i).$$

We propose to find Π using a Gaussian approximation method. As observed by

Berrou *et al.* [14], [15] and Colavolpe *et al.* [20], within a certain region of channel signal-to-noise ratio (CSNR), and for large information block length L, the LLR generated by the Turbo decoder is approximately Gaussian with mean $+M$ or $-M$, and variance σ_Λ^2. Therefore, the transition probability distribution for this equivalent channel can be approximated by

$$p(\Lambda_l|U_l = i) = \frac{1}{\sqrt{2\pi\sigma_\Lambda^2}} e^{-\frac{(\Lambda_l-(2i-1)M)^2}{2\sigma_\Lambda^2}},$$

$$i = 0, 1, \quad l = 1, 2, \cdots, kR_s.$$

The values of M and σ_Λ^2 depend on the structure of the Turbo encoder, the channel statistics, as well as the decoding algorithm used in the Turbo decoder. While analytical expressions for M and σ_Λ^2 are intractable, we obtain a reliable estimation of their values by data training. For this channel model, we are now able to give the transition probability matrix Π as

$$\begin{aligned}
\pi_{ij} &= Pr\{Y = j|U = i\} = Pr\{\Lambda \in (T_{j-1}, T_j)|U = i\} \\
&= \frac{1}{\sqrt{2\pi\sigma_\Lambda^2}} \int_{T_{j-1}}^{T_j} e^{-\frac{(\lambda-(2i-1)M)^2}{2\sigma_\Lambda^2}} d\lambda \\
&= \frac{1}{2}\mathrm{erfc}\left(\frac{T_{j-1} - (2i-1)M}{\sqrt{2\sigma_\Lambda^2}}\right) - \frac{1}{2}\mathrm{erfc}\left(\frac{T_j - (2i-1)M}{\sqrt{2\sigma_\Lambda^2}}\right),
\end{aligned}$$

where

$$\mathrm{erfc}(x) = \frac{2}{\sqrt{\pi}} \int_x^\infty e^{-t^2} dt$$

is the complementary error function.

Hence, the channel transition probability matrix of our DMC model can be computed in terms of the quantization step Δ, the channel SNR, and the complementary error function. It can be observed that this DMC is "weakly" symmetric in the sense that its transition probability matrix $\mathbf{\Pi}$ can be partitioned (along its columns) into symmetric arrays — where a symmetric array is defined as an array whose rows are permutations of each other, and so are the columns [32]; therefore, its capacity is achieved by a uniform input distribution. For each channel SNR, the quantization step size Δ of the q-bit demodulator is chosen such that the capacity of this DMC is maximized.

3.6 DBMC Model for Turbo-Coded AWGN and Rayleigh Channels

Besides the simple DMC model described in the previous section, we can also obtain the transition probability matrix $\mathbf{\Pi}$ via data training. Note that in this expanded channel model, because of Turbo encoding and decoding, the memoryless assumption is in fact not very accurate, particularly for low CSNRs. Indeed, it should be regarded as a channel with memory. However, to keep our COVQ design tractable, we regard it as a discrete block-memoryless channel (DBMC) with 2^{kR_s} inputs and 2^{qkR_s} outputs, and ignore the memory between the blocks (inter-block memory). We can estimate the $2^{kR_s} \times 2^{qkR_s}$ transition probability matrix $\mathbf{\Pi}$ by using a long training sequence,

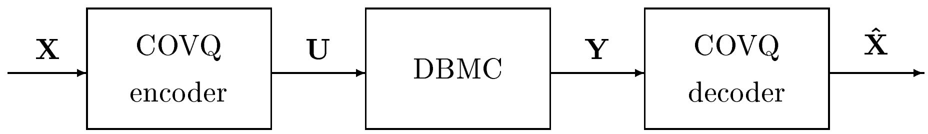

Figure 3.2: Expanded DBMC model of our system.

and then use Blahut's algorithm [17] to calculate the capacity for this channel. The quantization step size Δ is chosen to maximize the channel capacity for each channel SNR. This is a more accurate method since it captures the block memory in the expanded channel. Also, as the block length, kR_s, gets large, the model becomes more accurate.

Since this method does not need the Gaussian assumption of the LLR distribution from the Turbo decoder output, we can also employ it for other channel models such as the Rayleigh fading channel.

3.7 COVQ Design

We next design a COVQ for the 2^{kR_s}-input 2^{qkR_s}-output DBMC (including the special case of the binary-input 2^q-output DMC). We now have the following setup.

The encoder mapping is described by a partition $\mathcal{P} = \{S_{\mathbf{u}} \subset \mathbb{R}^k : \mathbf{u} \in \mathcal{U}^{kR_s}\}$ according to $\mathcal{E}(\mathbf{x}) = \mathbf{x}$ if $\mathbf{x} \in S_{\mathbf{u}}, \mathbf{u} \in \mathcal{U}^{kR_s}$, where $\mathbf{x} = (x_1, x_2, \cdots, x_k)$ is a block of k successive source samples. The DBMC is described by its block channel transition matrix $P(\mathbf{y}|\mathbf{u})$, where $\mathbf{u} \in \mathcal{U}^{kR_s}$ and $\mathbf{y} \in \mathcal{Y}^{qkR_s}$. Finally, the decoder mapping $\mathcal{D}$ is given by a codebook $\mathcal{C} = \{\mathbf{c}_{\mathbf{y}} \in \mathbb{R}^k : \mathbf{y} \in \mathcal{Y}^{qkR_s}\}$ according to $\mathcal{D}(\mathbf{y}) = \mathbf{c}_{\mathbf{y}}, \mathbf{y} \in \mathcal{Y}^{qkR_s}$.

As both models involve estimations using long training sequences, the average distortion per sample in (3.4) and the centroid condition (3.6) need to be modified accordingly. Suppose the training sequence has length T, then the average distortion per sample is defined by

$$D((\mathcal{P}, \mathcal{C}); b) \triangleq \frac{1}{T} \sum_{t=1}^{T} \sum_{j=1}^{N} \frac{1}{k} P(j|b(i_t)) d(\mathbf{x}_t, \mathbf{y}_j),$$

and the centroid condition is expressed as

$$\mathbf{y}_j = \frac{\sum_{i=1}^{N} P(j|b(i)) \sum_{l:\mathbf{x}_l \in S_i} \mathbf{x}_l / T}{\sum_{i=1}^{N} P(j|b(i)) |S_i| / T},$$

where $|S_i|$ denotes the number of training sequences in region S_i. The nearest neighbor condition remains the same as in (3.5). By using long training sequences, we also circumvent the cumbersome evaluations of high-dimensional numerical integrals in (3.4) and (3.6).

The codebook can be pre-computed off-line. Therefore, the COVQ decoding is implemented simply by a table-lookup with no extra computation. However, the memory for storing the codebook is high for large values of qkR_s.

3.8 Numerical Results and Discussion

In this section, we present numerical results in terms of the signal-to-distortion ratio (SDR) versus the channel signal-to-noise ratio (CSNR), which is defined by

$$CSNR = \frac{E_s}{N_0/2} = \frac{1}{N_0/2},$$

where E_s is the average symbol energy, and $E_s = E_b R_c / R_s$, where E_b is the average energy per information bit.

We first compare the performance of the original COVQ to that of a COVQ designed for our Turbo-coded systems. In the original COVQ scheme, the q-bit scalar quantizer is directly applied to the real-valued channel output. In all cases, 80,000 training source vectors are used. The system that does not employ Turbo codes, consists of a COVQ with rate R_s=2 bits/source symbol and dimension $k = 2$; thus its overall transmission rate is 0.5 source symbol/channel symbol. The Turbo code in the COVQ system using channel coding is a rate-1/2, 16-state code with generator (37, 21), and block length L=65536 bits. A pseudo-random interleaver is used [15, 48], and the number of decoding iterations is 10. The dimension of the COVQ source input is also $k = 2$, and the quantization rate is $R_s = 1$; therefore, with the rate $R_c = 1/2$ Turbo code, the overall transmission rate is also $R_c/R_s = 0.5$ source symbol/channel symbol.

In Table 3.1, we present performance results for the quantization of a memoryless

Gaussian source over a BPSK-modulated AWGN channel. The first two columns provide the SDR performance for COVQ without Turbo codes, the middle two columns give the performance for COVQ based on Turbo codes and the DMC model (using the Gaussian LLR assumption), and the last two columns give the performance for the COVQ-Turbo system based on the DBMC model (direct channel estimation). For very low CSNRs (≤ 0.5 dB), the original COVQ offers better performance than that of our COVQ-Turbo systems. However, in this low CSNR region, the bit error rates of both the Turbo-coded and uncoded sequences are very high (above 10^{-2} level); therefore, the performance in this low CSNR region is not of much practical interest. Around 0.6 dB, the Turbo code gives a water-fall BER performance, resulting in a quick improvement in the overall SDR performance of the whole system. From 0.7 dB and above, the performances of both our DMC and DBMC based systems quickly reach that of the COVQ designed for the noiseless channel. This is due to the excellent BER performance provided by the Turbo code (below 10^{-5} level). For high CSNRs, the Gaussian approximation is no longer very accurate; this is reflected in a small gap between the SDR performances of the DMC and the DBMC model. Note that while the SDR performance of our systems remains constant for CSNR>0.7 dB, the performance curve of the original COVQ system keeps increasing, although at a very slow slope. Since the original COVQ system can potentially reach an SDR performance of 9.52 dB (for noiseless channels) as opposed to the 4.43 dB achievable by our joint source-channel coding systems, at some point, the original COVQ will again outperform our systems. In fact, this occurs at about 1.8 dB in CSNR for the case considered here. The performance limitation for very high CSNRs imposed to our system is due to the fact that we sacrifice half of the overall rate for error-correction. In reality, if the performance at very high CSNRs (for example, above 1.8 dB for the discussed case) is also of interest, we can re-allocate the rate between the source coding and channel coding. This can be achieved by using a high-rate Turbo code and assigning more of the available bits to the COVQ; in this case, our joint source-channel coding scheme will still offer a better performance than that provided by pure COVQ.

Similar numerical results for a Gauss-Markov source with $\rho = 0.9$ are presented in Table 3.2. Also starting at around 0.6 dB, where the Turbo code's water-fall BER performance occurs, a slight increment of CSNR results in a drastic improvement of our systems' performances. For a long range of CSNRs above 0.6 dB, the performance of our DBMC system offers a significant gain (up to 3.8 dB for q=1 and 2.5 dB for q=4) over the original COVQ design.

In Tables 3.3 and 3.4, we present performance results simulated for quantizing a memoryless Gaussian source or a Gauss-Markov source with ρ=0.9 over a Rayleigh fading channel with known side information. For the sake of comparison, we also give results of a COVQ for a Rayleigh fading channel without using Turbo codes. We observe that the SDR performance improves quickly from 2.4 dB to 2.6 dB, which corresponds to the water-fall region of the BER performance of Turbo codes for Rayleigh fading channels (with known side information) [46]. At CSNR=2.6 dB, for a Gaussian source, our system yields a 1.66 dB gain for q=1, and a 0.97 dB gain for q=4. For a Gauss-Markov source with ρ=0.9, at the same CSNR value, the gains are 4.2 dB for q=1, and 2.98 dB for q=4, respectively.

In the above tables, we also observe improvements due to the increase of the soft-decision decoding resolution q. For example, in Table 3.4, without using Turbo codes, the SDR performance gain from $q = 1$ to $q = 4$ is around 1.2 dB for all CSNRs. With Turbo codes, the gain due to the increase of q is about 1.3 dB at CSNR=2.0 dB; but it becomes less obvious as CSNR increases. Beyond CSNR=2.6 dB, there is no gain due to the increase of q. This is because for such high CSNRs, Turbo codes offer very powerful error-correcting ability, making the Turbo-coded Rayleigh fading channel behave like a noiseless channel.

Finally, we compare our joint source-channel coding system with two other schemes. One is the traditional tandem scheme, (which consists of a noiseless LBG-VQ followed by a regular Turbo code independently designed of each other); the other scheme is proposed by Ho [48]. As shown in Fig. 3.3, the DMC based scheme offers comparable performance to Ho's system; however, the DBMC based system offers better performance that Ho's system (with an SDR gain up to 1.5 dB), because it captures the channel memory. Furthermore, both of our schemes have lower complexity than that of Ho's. In comparison with the tandem scheme, at CSNR=0 dB, our DBMC scheme with $q = 4$ can achieve a gain of about 3 dB in SDR; at CSNR=0.6 dB, the gain is about 3.5 dB. The performance is improved when q increases. For low CSNRs, the gain from $q = 1$ to $q = 4$ is as big as 1 dB. However, the most significant gain is achieved at $q = 2$. For high CSNRs, the gain due to increasing q is less obvious, since the performance is already very close to the performance of the system under no channel errors.

As a concluding remark for this chapter, it is important to point out that, although COVQ is an efficient and powerful data compression scheme, there are usually still some *residual* redundancy in the COVQ encoder output either in the form of non-uniformity or memory. How to exploit this residual redundancy presents another interesting problem, which will be treated subsequently in the following chapters.

CSNR (dB)	Without TC		With TC (DMC Model)		With TC (DBMC Model)	
	$q=1$	$q=4$	$q=1$	$q=4$	$q=1$	$q=4$
0.2	2.79	3.50	1.57	1.91	2.13	2.48
0.3	2.84	3.56	1.66	2.00	2.29	2.63
0.4	2.89	3.62	1.81	2.16	2.50	2.83
0.5	2.94	3.67	2.13	2.47	3.07	3.33
0.6	3.00	3.73	3.31	3.53	4.34	4.02
0.7	3.05	3.79	4.34	4.36	4.42	4.43
0.8	3.10	3.85	4.39	4.41	4.42	4.43
0.9	3.16	3.91	4.41	4.41	4.43	4.43
1.0	3.21	3.97	4.41	4.41	4.43	4.43
1.1	3.27	4.03	4.41	4.41	4.43	4.43
1.2	3.33	4.09	4.41	4.41	4.43	4.43
∞	9.52	9.52	4.43	4.43	4.43	4.43

Table 3.1: SDR (in dB) performances of COVQ based on Turbo codes for the AWGN channel, (COVQ rate R_s=1 bit/source symbol, Turbo code rate R_c=1/2 bit/channel symbol), compared with that of COVQ without Turbo codes, (COVQ rate R_s=2 bits/source symbol). Both use dimension k=2 and a Gaussian Source (ρ=0).

CSNR (dB)	Without TC		With TC (DMC Model)		With TC (DBMC Model)	
	$q=1$	$q=4$	$q=1$	$q=4$	$q=1$	$q=4$
0.2	3.80	4.92	1.75	2.96	2.55	3.76
0.3	3.86	5.01	2.46	3.09	2.81	3.98
0.4	3.91	5.08	2.66	3.31	3.16	4.25
0.5	3.97	5.16	3.10	3.75	4.21	4.92
0.6	4.03	5.25	4.95	5.19	7.58	7.63
0.7	4.09	5.33	7.57	7.59	7.88	7.89
0.8	4.16	5.41	7.79	7.80	7.90	7.91
0.9	4.22	5.49	7.83	7.83	7.91	7.91
1.0	4.28	5.57	7.85	7.85	7.91	7.91
1.1	4.34	5.66	7.86	7.86	7.91	7.91
1.2	4.41	5.74	7.87	7.87	7.91	7.91
∞	13.52	13.52	7.91	7.91	7.91	7.91

Table 3.2: SDR (in dB) performances of COVQ based on Turbo codes for the AWGN channel, (COVQ rate R_s=1 bit/source symbol, Turbo code rate R_c=1/2 bit/channel symbol), compared with that of COVQ without Turbo codes, (COVQ rate R_s=2 bits/source symbol). Both use dimension k=2 and a Gauss-Markov Source (ρ=0.9).

CSNR (dB)	Without TC			With TC (DBMC Model)		
	$q=1$	$q=2$	$q=4$	$q=1$	$q=2$	$q=4$
2.0	2.54	3.08	3.25	2.01	2.31	2.40
2.1	2.57	3.11	3.29	2.19	2.47	2.56
2.2	2.61	3.16	3.33	2.33	2.60	2.69
2.3	2.65	3.20	3.37	2.69	2.93	3.01
2.4	2.68	3.24	3.41	3.60	3.73	3.78
2.5	2.72	3.28	3.45	4.37	4.38	4.39
2.6	2.76	3.32	3.50	4.42	4.43	4.43
2.7	2.79	3.36	3.54	4.43	4.43	4.43
2.8	2.83	3.40	3.58	4.43	4.43	4.43
2.9	2.87	3.44	3.52	4.43	4.43	4.43
3.0	2.91	3.48	3.66	4.43	4.43	4.43
3.1	2.95	3.52	3.71	4.43	4.43	4.43
3.2	2.98	3.57	3.75	4.43	4.43	4.43
∞	9.52	9.52	9.52	4.43	4.43	4.43

Table 3.3: SDR (in dB) performances of COVQ based on Turbo codes for the Rayleigh fading channel with known side information, (COVQ rate R_s=1 bit/source symbol, Turbo code rate R_c=1/2 bit/channel symbol), compared with that of COVQ without Turbo codes, (COVQ rate R_s=2 bits/source symbol). Both use dimension k=2 and a Gaussian Source ($\rho = 0$).

CSNR (dB)	Without TC			With TC (DBMC Model)		
	$q=1$	$q=2$	$q=4$	$q=1$	$q=2$	$q=4$
2.0	3.41	4.28	4.57	2.38	3.52	3.69
2.1	3.46	4.33	4.63	2.61	3.70	3.87
2.2	3.52	4.39	4.69	2.86	3.92	4.08
2.3	3.56	4.45	4.75	3.48	4.36	4.52
2.4	3.61	4.51	4.81	5.37	5.72	5.93
2.5	3.65	4.57	4.87	7.71	7.73	7.74
2.6	3.71	4.62	4.93	7.91	7.91	7.91
2.7	3.77	4.68	4.99	7.91	7.91	7.91
2.8	3.82	4.74	5.05	7.91	7.91	7.91
2.9	3.88	4.80	5.11	7.91	7.91	7.91
3.0	3.92	4.86	5.17	7.91	7.91	7.91
3.1	3.97	4.91	5.23	7.91	7.91	7.91
3.2	4.03	4.97	5.29	7.91	7.91	7.91
∞	13.52	13.52	13.52	7.91	7.91	7.91

Table 3.4: SDR (in dB) performances of COVQ based on Turbo codes for the Rayleigh fading channel with known side information, (COVQ rate R_s=1 bit/source symbol, Turbo code rate R_c=1/2 bit/channel symbol), compared with that of COVQ without Turbo codes, (COVQ rate R_s=2 bits/source symbol). Both use dimension k=2 and a Gauss-Markov source with $\rho = 0.9$.

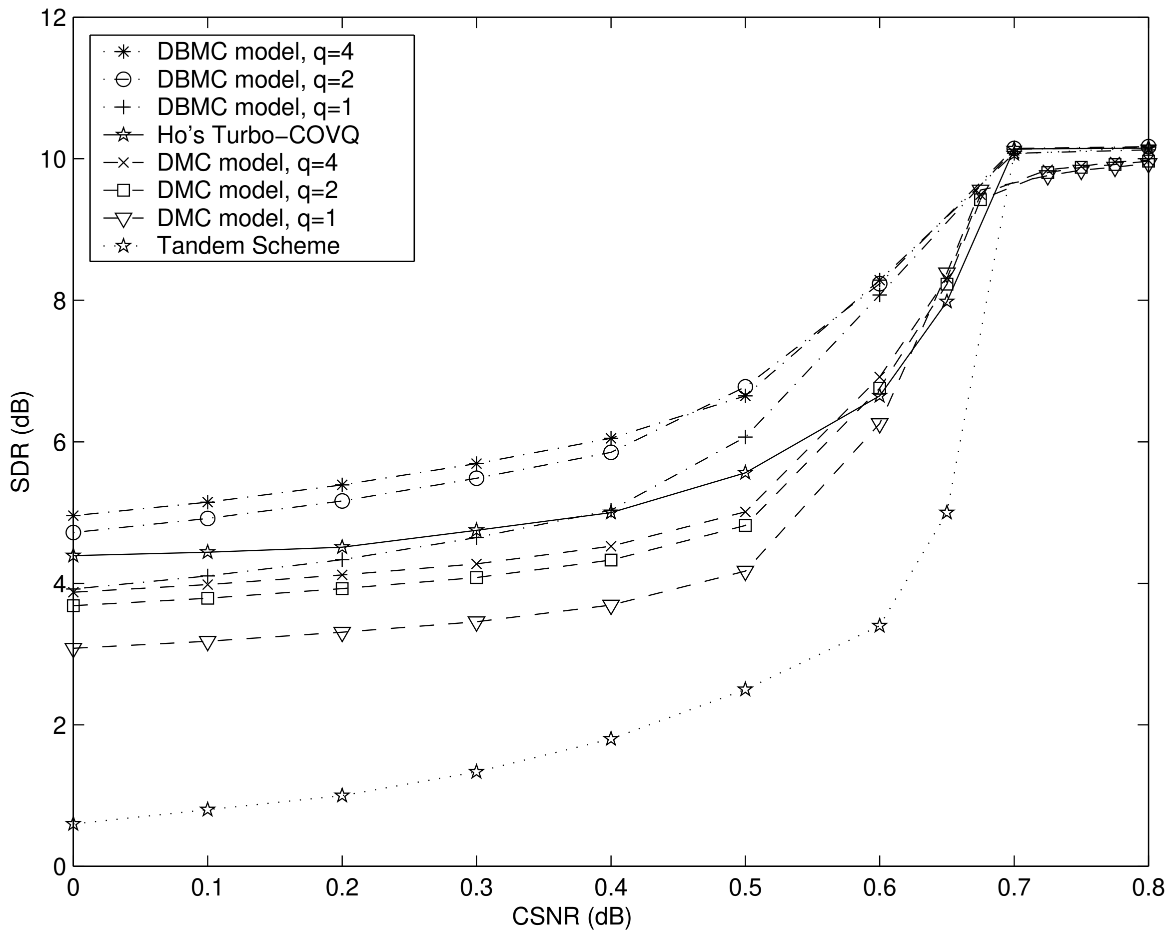

Figure 3.3: SDR performance of Turbo-coded AWGN channel, k=4, R_s=1, R_c=1/2. Gauss-Markov source with $\rho = 0.9$.

Chapter 4

Systematic Turbo Codes for Non-Uniform I.I.D. Sources

4.1 Introduction

In almost all the theory and practice of error-control coding, the source that is encoded for transmission over the channel is usually assumed to be a binary memoryless Bernoulli (1/2) source, i.e., it generates uniform i.i.d. bit streams $\{U_k\}_{k=1}^{\infty}$, where

$$Pr\{U_k = 0\} = Pr\{U_k = 1\} = 1/2.$$

In reality, however, natural sources (e.g., image and speech sources) often exhibit substantial amounts of redundancy in the form of memory and/or non-uniformity [3]; in this case, a source encoder would be used. An ideal source encoder would be able to eliminate all the source redundancy and hence produce a uniform i.i.d. sequence of bits, which would then be used as the input of the channel encoder. However, most existing source encoders are suboptimal (particularly fixed-length encoders that are commonly used for transmission over noisy channels). As a result, the input to the channel encoder contains a certain amount of *residual* redundancy. For example, the 4.8 kbits/s US Federal Standard 1016 CELP speech vocoder produces line spectral parameters that contain 41.5% of residual redundancy due to non-uniformity and memory [5]. For uncompressed sources, it has been observed that many binary images (e.g., facsimile and medical images) may contain as much as 80% of redundancy in the form of non-uniformity (e.g., [18], [49]), this corresponds to the *a priori* probability $Pr\{U_k = 0\} = 0.97$. Therefore, transmission of sources with a considerable amount of residual or natural redundancy is an important issue. Several studies (e.g., [4]–[6], [31, 41, 65, 75, 77], etc.) have shown that appropriate use of the source redundancy can significantly improve the system performance.

In this chapter, we investigate the issue of designing Turbo codes for non-uniform i.i.d. sources sent over AWGN and Rayleigh fading channels. Unlike the methods in the previous chapter, in which the channel statistics is incorporated in the source code design, here we propose a different joint source-channel coding approach — exploiting the source redundancy in the design of Turbo codes. Our objective is to strive to come as close to the Shannon limit as possible.

A non-uniform i.i.d. source is described by the non-equiprobable probability distribution of the bit stream. The source emits a sequence of bits $\{U_k\}_{k=1}^{\infty}$ with probability

$$Pr\{U_k = 0\} = p_0, \quad k = 1, 2, 3, \cdots .$$

To take advantage of the source redundancy in the form of non-uniformity, the extrinsic information in the Turbo decoder is modified; as a result, the bit error rate (BER) performance of the Turbo coded system is significantly enhanced. We also observe that while the original Berrou Turbo code which uses the $(37, 21)$ convolutional code in each constituent encoder offers extraordinary performance (excellent "waterfall" region at very low SNR's) for uniform i.i.d. sources, it provides relatively poor performance with a considerably high error floor when the source is non-uniform. Furthermore, when the source distribution is more biased, performance degradation becomes more obvious. An analysis of the encoder's structure reveals to us that it is important to find more suitable constituent encoders for non-uniform i.i.d. sources. Through a systematic search we performed by simulations, we show that some constituent encoders substantially outperform Berrou's $(37, 21)$ encoders for a given non-uniform i.i.d. source. Significant coding gains are further achieved by combining this optimized encoder structure with the appropriately modified decoder that exploits the source non-uniformity.

4.2 Modifications of the Decoder Extrinsic Information

In the BCJR algorithm used by the Turbo decoder, we observe that the log-likelihood ratio (LLR) produced by the Turbo decoder can be decomposed into three terms:

$$\Lambda(U_k) = L_{ch}(U_k) + L_{ex}(U_k) + L_{ap}(U_k),$$

where $L_{ch}(U_k)$, $L_{ex}(U_k)$, and $L_{ap}(U_k)$ are the channel transition term, the extrinsic information term, and the *a priori* term, respectively, [15], [63]. The extrinsic information produced by one constituent decoder is used as the *a priori* estimation for the other constituent decoder. At the first iteration, for the first constituent decoder, if the source is uniform i.i.d., which is the case investigated in [15], we have

$$L_{ap}(U_k) = \log \frac{P(U_k = 1)}{P(U_k = 0)} = 0,$$

since $P(U_k = 1) = P(U_k = 0) = 1/2$.

When the source is non-uniform i.i.d., $\log((1 - p_0)/p_0) \neq 0$ is used as the initial *a priori* input for the first decoder in the first iteration. As a result, in the output $\Lambda(U_k)$ produced by the first decoder,

$$L_{ap}(U_k) = \log \left(\frac{1 - p_0}{p_0} \right).$$

By passing this term together with the extrinsic information from the first decoder to the second decoder, a BER performance gain is observed[1]. It can be shown via the BCJR algorithm's derivation that $\log((1-p_0)/p_0)$ will appear in the output $\Lambda(U_k)$ as an extra term. In our design, we then use $L_{ex} + \log((1-p_0)/p_0)$ as the new extrinsic information for both decoders at each iteration. With this simple procedure, the performance is greatly improved.

4.3 Constituent Encoder Structure

In the original design of Turbo codes by Berrou *et al*, extensive search was performed to find the best constituent encoder structure. The one that offers the best waterfall BER performance was found to be a 16-state recursive systematic convolutional (RSC) encoder with tap coefficients (37, 21) [15]. The structure of Berrou's (37, 21) constituent encoder is depicted in Fig. 4.1. Note that in [15] the source is uniform i.i.d. When the source is non-uniform i.i.d., we found that a Turbo code using Berrou's encoder performs poorly over a wide range of E_b/N_0 values, where E_b is the average energy per source bit, and $N_0/2$ is the additive noise variance. As the source gets more biased, performance degradation becomes even more obvious. When p_0=0.9, our simulations show that the whole performance curve drops down very slowly and

[1]This simple modification of appropriately using the source information in the Turbo decoder for non-uniform sources was also briefly mentioned in [44] (see Remark 4.d on page 433) but not explicitly studied and evaluated. One of our goals in this paper is to explicitly assess the gains achieved by this method and examine how close we can come to the Shannon limit.

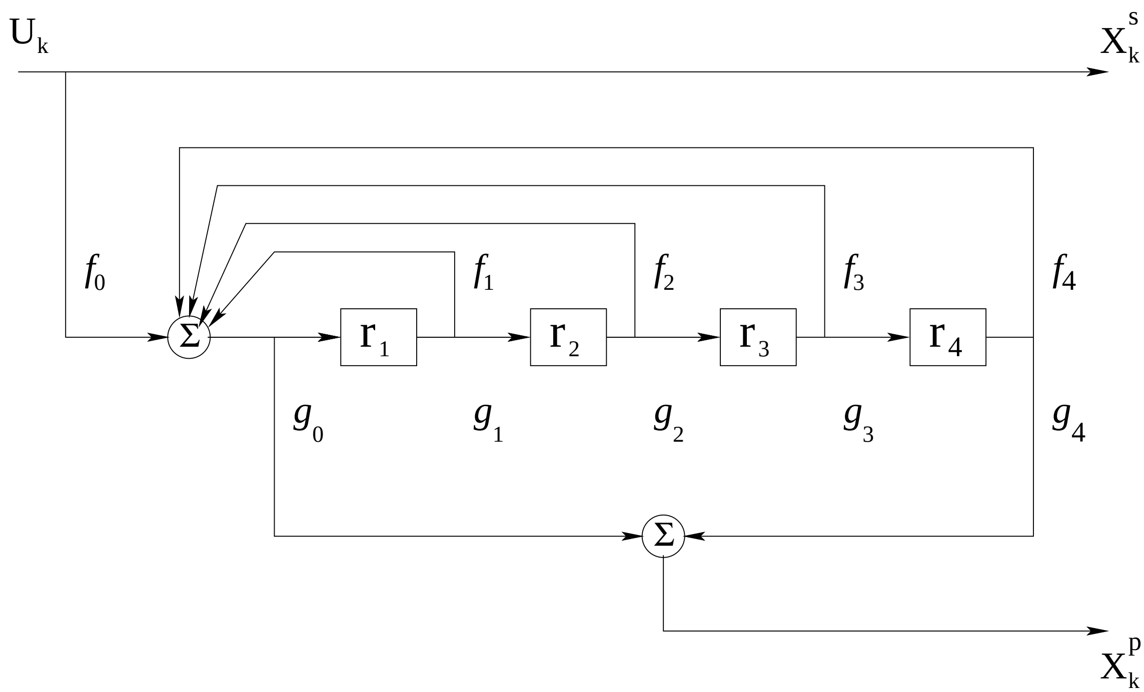

Figure 4.1: Berrou's (37, 21) constituent encoder.

there is virtually no sharp water-fall region.

To reveal why Berrou's (37, 21) code performs so badly when the source is non-uniform i.i.d., we consider the state transition of the encoder as follows.

First, consider the extreme case — the input sequence contains only one "1" at the beginning of the sequence and all the other bits are zeros (which is the case of an extremely biased or nearly deterministic source). Suppose the $(37, 21)$ encoder shown in Fig. 4.1 starts at the all-zero state 0000, where each digit represents the content of each shift register. The encoder would remain in this state until the "1" arrives, which would cause a transition to state 1000. The following state transitions would be $1100 \to 0110 \to 0011 \to 0001 \to 1000$, and from then on the state transition would be confined in a short cycle consisting of the above 5 states while all the other states remain dormant. We herein cite the following definition from [16].

Definition: Let a recursive convolutional encoder initially be in any state other than the all-zero state. After introducing a certain number of consecutive 0s at the input, the encoder will return to its initial state. This number of 0s required to drive the encoder to its initial state is called the *period* of the encoder, and is denoted by L.

Therefore, the total number of excitable (or reachable) states in our extreme case scenario, where the encoder (initialized at the all-zero state) is driven by an input sequence containing a single 1 and a very large number of 0s, is equal to $L+1$. Thus, Berrou's (37, 21) code has only 6 out of 16 excitable states, which indicates that its encoder memory is not fully exploited. Similarly, we can find that for a encoder with feedback 35, its period is $L = 7$; so its number of excitable states is 8. However, for an encoder with feedback 31, its period is L=15, and hence all of its 16 states can be excited. The following table provides the periods for some 16-state recursive convolutional encoders.

Next, consider a heavily biased source with p_0 very close to but still strictly less than 1. Then with high probability, the state transition is confined within a cycle as listed above over a relatively long time period. A second "1" might bring the state

feedback	21	37	25	33	27	35	23	31
L	4	5	6	6	7	7	15	15

Table 4.1: Periods of 16-state recursive convolutional encoders.

transition outside the cycle; however, again with high probability, the state transition would be confined in a new cycle consisting of a few other possible states. Therefore, for such sources, feedback 31 and 23 are apparently better structures than others as indicated in Table 4.1. When p_0 decreases, the advantage of using a feedback with larger L becomes less obvious. Therefore, for a source with a given probability distribution p_0, we need to find the encoder structure that gives the best possible performance.

The number of possible encoders grows exponentially with the encoder memory. Consider memory M=4, and denote the coefficients of the feedback and feed-forward polynomials of a RSC encoder in binary form as $\{f_0, f_1, f_2, f_3, f_4\}$ and $\{g_0, g_1, g_2, g_3, g_4\}$, respectively, where $f_i, g_j = 0$ or 1, $i, j = 0, 1, \cdots, 4$. Since f_0 is the tap for the encoder input, we always have $f_0 = 1$. Then altogether there are $2^4 \times (2^5 - 1) = 496$ possible combinations; therefore, an exhaustive search is impractical. However, among these 496 possible choices, some encoders obviously would give poor performance, and therefore should be eliminated in our search for the best encoders.

It is known that for convolutional codes, larger memory generally results in better performance [76]; although for Turbo codes, to find one with $M > 4$ that gives better performance than that of Berrou's (37,21) code requires extensive simulations [56]. In our simulations, we only focused on 16-state encoders. To fully exploit the memory, the last tap in the feedback f_4 is set to 1. The following theorem gives a more rigorous explanation for this choice. First, we give the following lemma.

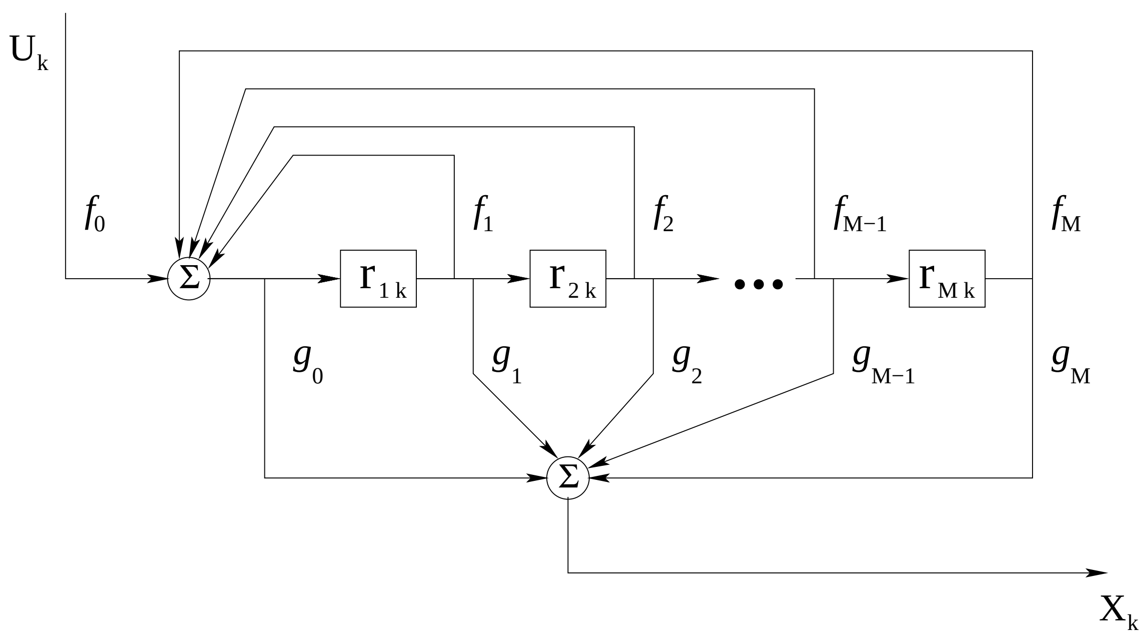

Figure 4.2: General structure of a recursive convolutional code.

Lemma 4.1 For a recursive convolutional encoder as depicted in Fig. 4.2, at time $k+1$, each state s_{k+1} can be reached from only two (state, input) pairs $(s_k^{(0)}, u_{k+1}^{(0)})$ and $(s_k^{(1)}, u_{k+1}^{(1)})$, where the two states at time k satisfy $s_k^{(0)} \neq s_k^{(1)}$. The necessary and sufficient condition that $u_{k+1}^{(0)} \neq u_{k+1}^{(1)}$ is $f_M = 1$.

Proof: The encoder state is determined by the content of each memory element of the shift register. Let the state at time k be $s_k = (r_{1k}, r_{2k}, \cdots, r_{Mk})$. By the shift register's structure, we have

$$
\begin{aligned}
r_{1,k+1} &= u_{k+1} \oplus \sum_{j=1}^{M} f_j \cdot r_{jk}, &(4.1)\\
r_{j,k+1} &= r_{j-1,k}, \quad j = 2, 3, \cdots, M, &(4.2)
\end{aligned}
$$

where $\oplus$ and the summation are modulo-2. Therefore, s_{k+1} can only be reached from two states which differ at r_{Mk}. Now rewrite (4.1) as follows,

$$u_{k+1} = r_{1,k+1} \oplus f_M \cdot r_{Mk} \oplus \sum_{j=1}^{M-1} f_j \cdot r_{jk}.$$

For both states $s_k^{(0)}$ and $s_k^{(1)}$, the summand produces the same result. When $f_M = 0$, the input u_{k+1} is independent of the encoder state being $s_k^{(0)}$ or $s_k^{(1)}$. When $f_M = 1$, different value of r_{Mk} results in different value of u_{k+1}. $\square$

Theorem 4.1 Consider a recursive convolutional encoder with feedback $\{f_1, f_2, \cdots, f_M\}$, and let a non-uniform i.i.d. source with distribution $p_0 \in (0,1)$ be its input. Then the encoder state distribution is asymptotically uniform (as the source sequence length tends to infinity) regardless of the value of p_0 if and only if $f_M = 1$.

Proof: The shift register is a fully connected (irreducible) time-invariant Markov chain when $p_0 \in (0,1)$; so the state distribution will asymptotically converge to the steady state distribution. Then it is equivalent to show that the uniform distribution is or is not the steady state distribution. From Lemma 4.1, there are only two (state, input) pairs $(s_k^{(0)}, u_{k+1}^{(0)})$ and $(s_k^{(1)}, u_{k+1}^{(1)})$ that may transit to state s_{k+1}. Without loss of generality, denote the possible encoder state as $0, 1, \cdots, 2^M - 1$. Assuming that at time k the state distribution is uniform, i.e.,

$$Pr(S_k = s) = 2^{-M}, \quad s = 0, 1, \cdots, 2^M - 1,$$

then at time $k+1$, if $f_M = 1$, we have

$$\begin{aligned} Pr(S_{k+1} = s_{k+1}) &= Pr(S_k = s_k^{(0)}, U_{k+1} = u_{k+1}^{(0)}) + Pr(S_k = s_k^{(1)}, U_{k+1} = u_{k+1}^{(1)}) \\ &= Pr(U_{k+1} = u_{k+1}^{(0)})2^{-M} + Pr(U_{k+1} = u_{k+1}^{(1)})2^{-M} \\ &= 2^{-M}, \end{aligned}$$

where the second equality is due to the independence of S_k and U_{k+1}, and the last equality holds because $u_{k+1}^{(0)} \neq u_{k+1}^{(1)}$ by Lemma 4.1. Therefore, $f_M = 1$ yields that the uniform distribution is the steady distribution of the encoder states. On the other hand, if $f_M = 0$, by Lemma 4.1, $u_{k+1}^{(0)} = u_{k+1}^{(1)}$, then

$$Pr(S_{k+1} = s_{k+1}) = 2Pr(U_{k+1} = u_{k+1}) \cdot 2^{-M}.$$

Therefore, when the source is biased, the uniform distribution is not the steady distribution of the encoder states when $f_M = 0$. Hence for any $p_0 \in (0, 1)$, the encoder state distribution is asymptotically uniform if and only if $f_M = 1$. As a final remark, we note that p_0=0.5 is a special trivial exception to the "only if" part. $\square$

Now by setting $f_4 = 1$, we have narrowed our possible choices down to $2^3 \times (2^5 - 1)$=248. To further eliminate redundant choices, we note the following.

For the same feedback coefficients, the choice of the feed-forward coefficients $\{g_0, g_1, g_2, g_3, g_4\}$ to be $\{1, 0, 0, 0, 0\}$ or $\{0,\ 1,\ 0,\ 0,\ 0\}$, or $\{0, 0, 0, 0, 1\}$ gives essentially the same performance. The same behavior is also observed if the feed-forward coefficients are $\{1, 0, 1, 0, 0\}$ or $\{0, 1, 0, 1, 0\}$, or $\{0, 0, 1, 0, 1\}$. Therefore, we can choose $g_0 = 1$, hence the total number of possible encoders becomes $2^3 \times 2^4$=128. Again, to avoid an exhaustive search, we obtain sub-optimal encoders with f_0=g_0=1 and f_4=1 through the following iterative steps:

1) Fix the feed-forward polynomial, (e.g., $\{g_0, g_1, g_2, g_3, g_4\}$=$\{1, 0, 0, 0, 0\}$), find (by simulation) the best feedback polynomial among the remaining possible choices;

2) Fix the feedback polynomial as the one found in step 1), find the best feed-forward polynomial among all remaining possible choices;

3) Fix the feed-forward polynomial as found in step 2), go back to step 1), if the feedback polynomial coincides with the one obtained in step 1), stop (otherwise, proceed to the next step).

Via this procedure, for a given non-uniform probability distribution, several encoders were found to outperform Berrou's $(37, 21)$ encoder significantly; they also have considerably lower error floors. Among them, the $(35, 23)$ encoder gives the best performance for $p_0 = 0.7$ and 0.8; when $p_0 = 0.9$, the best encoder is $(31, 23)$.

However, when the source is uniform i.i.d, Berrou's $(37, 21)$ encoder gives a better performance in the water-fall region than the above encoders, though its error floor is higher. For uniform sources, improving the error floor region of Turbo codes is usually achieved at the expense of the waterfall region; however, for the non-uniform case, this tendency seems to decrease as p_0 increases.

4.4 Numerical Results and Discussion

In this section, we present simulation results of Turbo codes for non-uniform i.i.d. sources over BPSK-modulated AWGN and Rayleigh fading channels. All simulated Turbo codes use the same pseudo-random interleaver introduced in [15]. The sequence length is $L = 512 \times 512 = 262144$ and 200 blocks are used; this would guarantee a reliable BER estimation at the 10^{-5} level with 524 errors. The number of iterations used in the decoder is 20. Simulations for our selected codes and the Berrou code are performed for rates $R_c = 1/3$ and $R_c = 1/2$, for p_0=0.5, 0.7, 0.8 and 0.9. To show the performance gains due to the use of the modified extrinsic information, simulations are also performed by using the unchanged extrinsic information; i.e., the decoder has no knowledge of the probability distribution and assumes the source is uniform[2].

[2] In this case, regardless of the source generated at the encoder (p_0 = 0.9, 0.8, 0.7 or 0.5), the performance of the system with unchanged extrinsic information at the decoder varies very slightly with p_0; so for this system, we can assume that $p_0 = 0.5$ at both the encoder and the decoder.

Figure 4.3 shows the performance comparison of Berrou's rate-1/3 $(37, 21)$ Turbo code and our selected Turbo codes for transmitting uniform and non-uniform i.i.d. sources (with p_0=0.5, 0.7, 0.8 and 0.9) over AWGN channels by using the modified extrinsic information in the decoder. The $(35, 23)$ code offers the best performance from p_0=0.7 up to 0.8; when $p_0 = 0.9$, the $(31, 23)$ seems to be the best choice. At the 10^{-5} BER level, the gains over the Berrou $(37, 21)$ code due to optimization of the encoder for p_0=0.7, 0.8 and 0.9 are 0.09, 0.51 and 1.18 dB, respectively. Furthermore, the gains due to using the modified extrinsic information for p_0=0.7 and 0.8 (with the $(35, 23)$ code) are 0.43 dB and 1.08 dB, respectively; for p_0=0.9 (with the $(31, 23)$ code) it is 2.46 dB. Similar results can be observed for rate-1/2 codes in Figure 4.4. For example, when p_0=0.9, our selected $(31, 23)$ code gives a gain of 1.08 dB over the Berrou $(37, 21)$ code at the 10^{-5} BER level, while the gain achieved due to exploiting the source redundancy in the form of non-uniformity is 2.20 dB.

In Figure 4.5, we provide the performance comparison for the Rayleigh fading channel with known channel state information. When p_0=0.7, the gain due to encoder optimization is visible only below the 10^{-5} BER level, because the performance offered by Berrou's code has a lower error floor than that in the AWGN channel case. When p_0=0.8, our $(35, 23)$ Turbo code offers a 0.33 dB gain at the 10^{-5} BER level over its $(37, 21)$ peer; when p_0=0.9, the gain is further increased up to 1.08 dB by using the $(31, 23)$ encoder. Furthermore, the gains due to exploiting the source redundancy in the form of non-uniformity become bigger than those obtained in the AWGN channel case: when p_0=0.7 and 0.8, the gains at a BER of 10^{-5} produced by the $(35, 23)$ Turbo code are 0.54 dB and 1.36 dB, respectively; when p_0=0.9, the $(31, 23)$ Turbo code realizes a significant 3.01 dB gain. Similar performances for rate-1/2 are shown in Figure 4.6. For example, at the same BER level, and for $p_0 = 0.9$, by using our selected $(31, 23)$ code, the gain achieved by encoder optimization is 1.03 dB, while the gain achieved due to using the modified extrinsic information is also 3.01 dB.

4.5 Shannon Limit

In his landmark papers [69], [70], Shannon established the *Lossy Information Transmission Theorem*, also known as the *Joint Source-Channel Coding Theorem with Fidelity Criterion* [59]. From this theorem, we know that for a given memoryless source and a given memoryless channel with capacity C, for sufficiently large source block lengths, the source can be transmitted via a source-channel code over the channel at a transmission rate of $r = R_c/R_s$ source symbols/channel symbols (where R_c and R_s are the channel coding and source coding rates, respectively) and reproduced at the receiver end within an end-to-end distortion given by D if the following condition is satisfied [59]:

$$r \cdot R(D) < C, \tag{4.3}$$

where $R(D)$ is the rate-distortion function. For a discrete binary non-uniform i.i.d. source with distribution p_0, we have that $D = P_e$ (BER) under the Hamming distortion measure [24]; then $R(D)$ becomes

$$R(P_e) = \begin{cases} h_b(p_0) - h_b(P_e), & 0 \le P_e \le \min\{p_0, 1-p_0\} \\ 0, & P_e > \min\{p_0, 1-p_0\} \end{cases}$$

where $h_b(\cdot)$ is the binary entropy function:

$$h_b(x) = -x \log_2 x - (1-x)\log_2(1-x).$$

The capacity of discrete-input memoryless channels is defined as the maximum of the mutual information between the channel input and output:

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

where the maximization is taken with respect to all input distributions $p(x)$.

For binary-input continuous-output AWGN channels,

$$\begin{aligned} I(X;Y) &= \sum_x \int_{-\infty}^{\infty} p(x,y) \log_2 \left[\frac{p(x,y)}{P(x)p(y)}\right] dy \\ &= \sum_x \int_{-\infty}^{\infty} p(y|x)P(x) \log_2 \left[\frac{p(y|x)}{p(y)}\right] dy \\ &= \sum_x \int_{-\infty}^{\infty} p(y|x)P(x) \log_2 \left[\frac{p(y|x)}{\sum_{x'} p(y|x')P(x')}\right] dy. \end{aligned} \tag{4.4}$$

The channel transitional probability density function of a binary-input continuous-output BPSK modulated AWGN channel is

$$p(y|x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{[y-(2x-1)]^2}{2\sigma^2}}.$$

Therefore, for a BPSK-modulated uniform channel input,

$$\begin{aligned}
I(X;Y) &= -\frac{1}{2}\int_{-\infty}^{\infty}\frac{1}{\sqrt{2\pi\sigma^2}}e^{-\frac{(y+1)^2}{2\sigma^2}}\log_2\left[\frac{1}{2}(1+e^{\frac{2y}{\sigma^2}})\right]dy \\
&\quad -\frac{1}{2}\int_{-\infty}^{\infty}\frac{1}{\sqrt{2\pi\sigma^2}}e^{-\frac{(y-1)^2}{2\sigma^2}}\log_2\left[\frac{1}{2}(1+e^{\frac{-2y}{\sigma^2}})\right]dy \\
&\triangleq C_{u0}+C_{u1}. \qquad (4.5)
\end{aligned}$$

By symmetry, $C_{u0}=C_{u1}$. Therefore,

$$\begin{aligned}
C_{AWGN} &= -\int_{-\infty}^{\infty}\frac{1}{\sqrt{2\pi\sigma^2}}e^{-\frac{(y+1)^2}{2\sigma^2}}\log_2\left[\frac{1}{2}(1+e^{\frac{2y}{\sigma^2}})\right]dy \\
&= 1-\int_{-\infty}^{\infty}\frac{1}{\sqrt{2\pi\sigma^2}}e^{-\frac{(y+1)^2}{2\sigma^2}}\log_2(1+e^{\frac{2y}{\sigma^2}})dy. \qquad (4.6)
\end{aligned}$$

Similarly, consider binary-input continuous-output BPSK modulated Rayleigh fading channels with known channel state information. With the probability density function of the fading amplitude given by

$$p_A(a)=2ae^{-a^2}, \quad a\geq 0,$$

and the channel transitional probability density function given by

$$p(y|x,a)=\frac{1}{\sqrt{2\pi\sigma^2}}e^{-\frac{[y-a(2x-1)]^2}{2\sigma^2}},$$

the mutual information becomes:

$$\begin{aligned}
I(X;Y) &= E_{(X,Y,A)}\log_2\left[\frac{p(X,Y|A)}{p(X|A)p(Y|A)}\right] \\
&= \sum_x\int_0^{\infty}\int_{-\infty}^{\infty}p(y|x,a)P(x)p_A(a)\log_2\left[\frac{p(y|x,a)}{\sum_{x'}p(y|x',a)P(x')}\right]dy\ da \\
&= -\frac{1}{2}\int_0^{\infty}\int_{-\infty}^{\infty}\frac{1}{\sqrt{2\pi\sigma^2}}e^{-\frac{(y+a)^2}{2\sigma^2}}\cdot 2ae^{-a^2}\cdot\log_2\left[\frac{1}{2}(1+e^{\frac{2ay}{\sigma^2}})\right]dy\ da \\
&\quad -\frac{1}{2}\int_0^{\infty}\int_{-\infty}^{\infty}\frac{1}{\sqrt{2\pi\sigma^2}}e^{-\frac{(y-a)^2}{2\sigma^2}}\cdot 2ae^{-a^2}\cdot\log_2\left[\frac{1}{2}(1+e^{\frac{-2ay}{\sigma^2}})\right]dy\ da. \qquad (4.7)
\end{aligned}$$

Again using symmetry the above two integrals are equal. Therefore we have

$$\begin{aligned} C_{Rayleigh} &= -\int_{0}^{\infty}\int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(y+a)^2}{2\sigma^2}} \cdot 2ae^{-a^2} \cdot \log_2\left[\frac{1}{2}(1+e^{\frac{2ay}{\sigma^2}})\right] dy\ da \\ &= 1 - \int_{0}^{\infty}\int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(y+a)^2}{2\sigma^2}} \cdot 2ae^{-a^2} \cdot \log_2(1+e^{\frac{2ay}{\sigma^2}}) dy\ da. \end{aligned} \tag{4.8}$$

From equations (4.6) and (4.8), we can see that for binary-input BPSK-modulated AWGN and Rayleigh fading channels, the channel capacity is a function of E_b/N_0; i.e., $C = C(E_b/N_0)$. Therefore, the optimum value of E_b/N_0 to guarantee a bit error rate of P_e can be solved using (4.3) assuming equality. This optimum value of E_b/N_0 is called the *Shannon limit*, or the *optimal performance theoretically achievable (OPTA)*. The Shannon limit cannot be explicitly solved for a BPSK-modulated input due to the lack of a closed form expression; so it is computed via numerical integration.

For the above simulations, we computed the OPTA at the 10^{-5} BER level for rates 1/2 and 1/3, and for $p_0 = 0.7$, 0.8 and 0.9. The OPTA values and OPTA gaps (distances between the performance of the designed systems and the corresponding OPTA values) are provided in Tables 4.2 and 4.3, respectively. We observe that the performances of our selected Turbo codes are significantly closer to the OPTA limit than those offered by their $(37, 21)$ peer. When p_0 increases, the gains achieved by using our selected encoders over the $(37, 21)$ encoder become more significant.

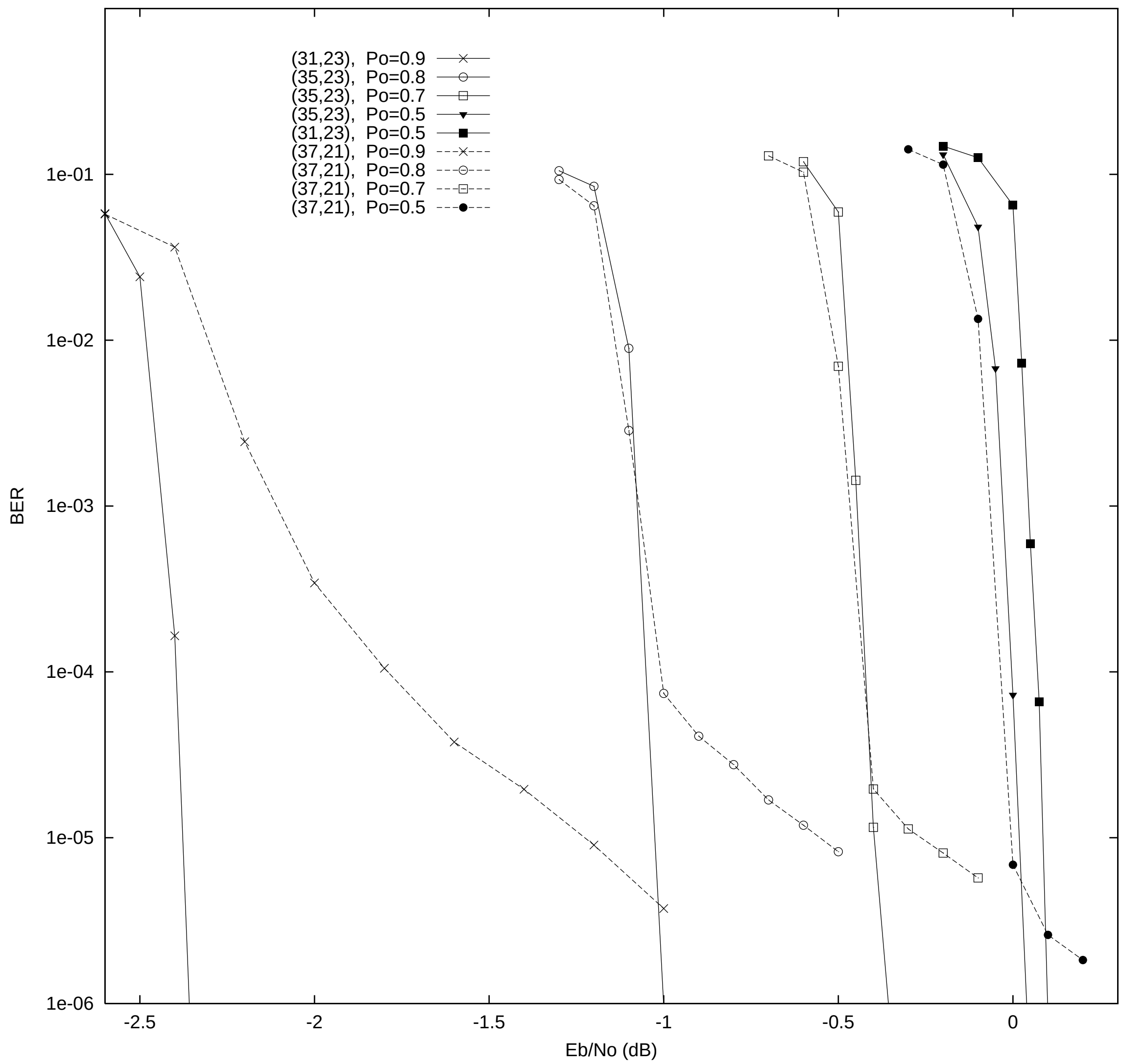

Figure 4.3: Turbo codes for non-uniform i.i.d. sources, R_c=1/3, L=262144, AWGN channel.

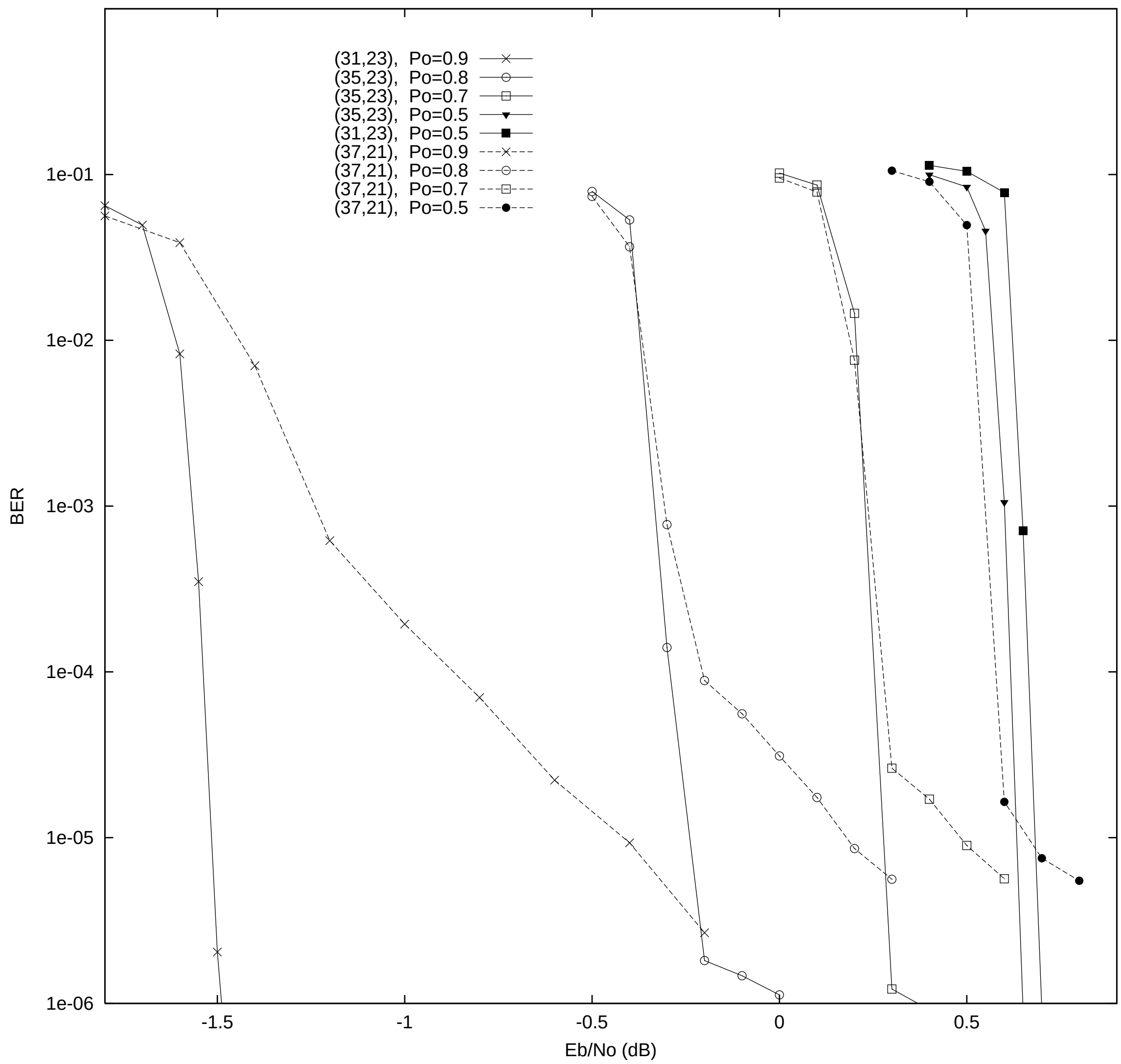

Figure 4.4: Turbo codes for non-uniform i.i.d. sources, R_c=1/2, L=262144, AWGN channel.

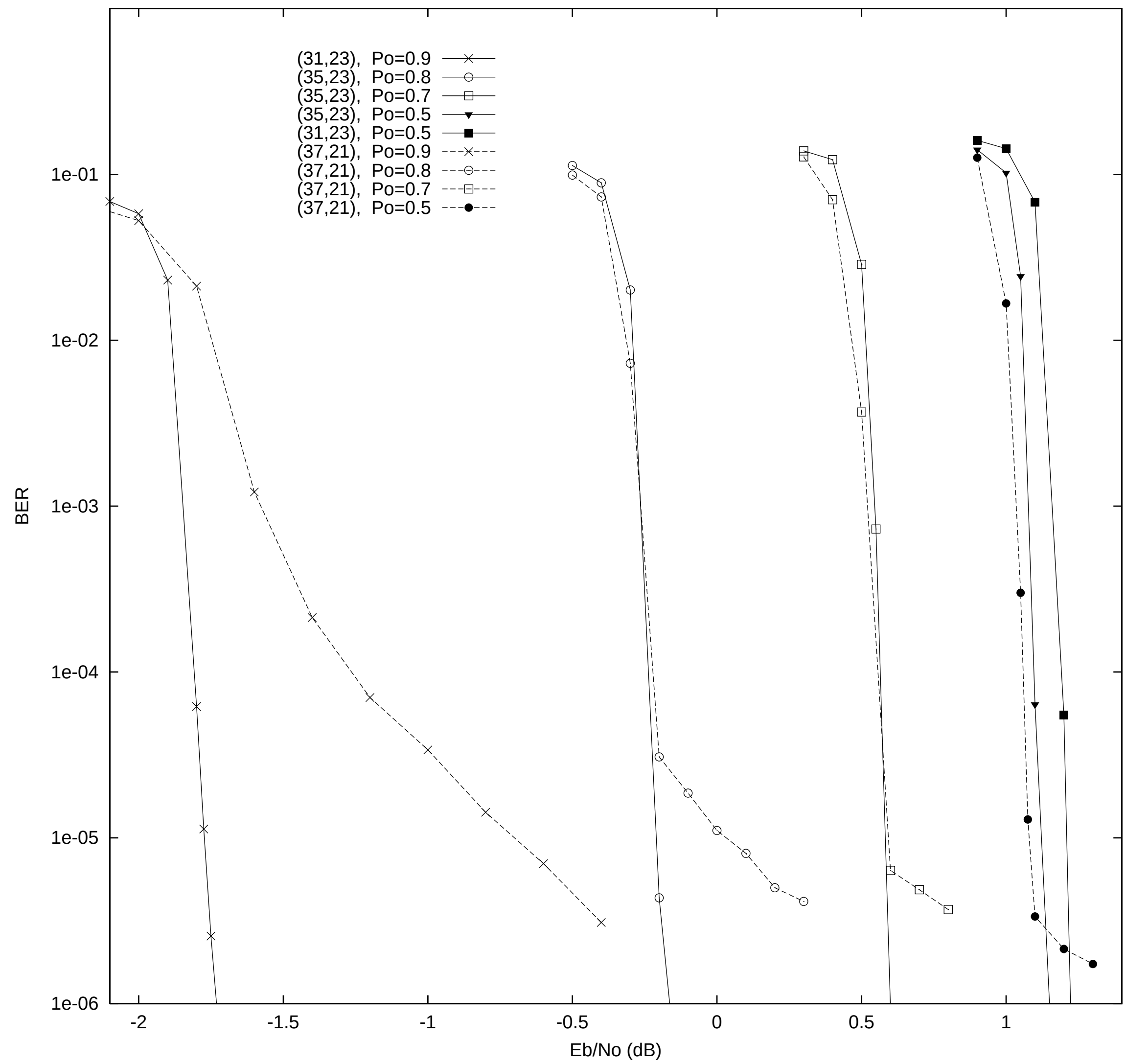

Figure 4.5: Turbo codes for non-uniform i.i.d. sources, R_c=1/3, L=262144, Rayleigh fading channel.

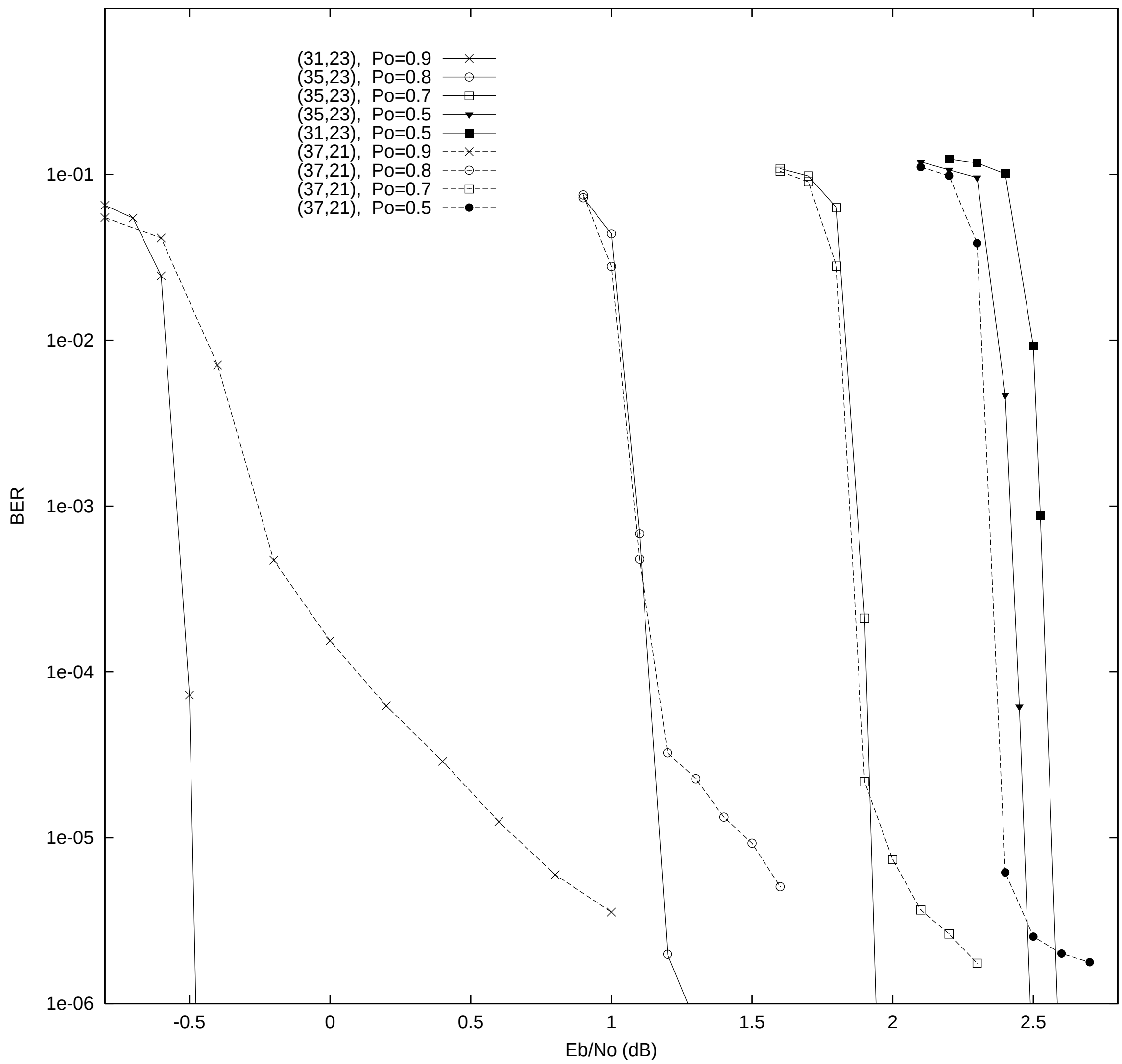

Figure 4.6: Turbo codes for non-uniform i.i.d. sources, R_c=1/2, L=262144, Rayleigh fading channel.

Source distribution	AWGN		Rayleigh	
	$R_c = 1/2$	$R_c = 1/3$	$R_c = 1/2$	$R_c = 1/3$
$p_0 = 0.7$	-0.62	-1.19	0.76	-0.34
$p_0 = 0.8$	-1.81	-2.24	-0.73	-1.56
$p_0 = 0.9$	-4.14	-4.40	-3.47	-3.96

Table 4.2: OPTA values at BER=10^{-5} level (in dB).

Turbo Coding System	AWGN		Rayleigh	
	$R_c = 1/2$	$R_c = 1/3$	$R_c = 1/2$	$R_c = 1/3$
(37,21), $p_0 = 0.7$	1.07	0.89	1.17	0.91
(35,23), $p_0 = 0.7$	0.87	0.80	1.16	0.91
(37,21), $p_0 = 0.8$	2.02	1.70	2.18	1.61
(35,23), $p_0 = 0.8$	1.56	1.19	1.88	1.28
(37,21), $p_0 = 0.9$	3.69	3.20	4.12	3.26
(31,23), $p_0 = 0.9$	2.61	2.02	2.99	2.18

Table 4.3: OPTA gaps in E_b/N_0 at BER=10^{-5} level (in dB).

Chapter 5

Non-Systematic Turbo Codes for Non-Uniform I.I.D. Sources

5.1 Introduction

In almost all the Turbo coding literature, the Turbo code encoders used are *systematic*. Non-systematic Turbo codes were recently proposed by Costello *et al.* [22], and followed by a series of works in [21, 23, 57]. The motivation for using non-systematic Turbo codes as an alternative to their systematic peers is due to their larger code space; therefore, there is a good potential that better codes *might* be found. However, the sources considered in these works are uniform i.i.d.

The Turbo encoders used in Chapter 4 are recursive *systematic* convolutional

(RSC) encoders. Despite the significant coding gains achieved, the performance can be further improved since the gaps to the Shannon limit are still relatively wide for heavily biased sources. For example, when rate R_c=1/2, p_0=0.8 and 0.9, for AWGN channels the OPTA gaps are 1.56 dB and 2.61 dB, respectively; for Rayleigh fading channels, the OPTA gaps are as big as 1.88 dB and 2.99 dB, respectively. An analysis on the encoder output reveals that the drawback lies in the *systematic* structure, which results in a mismatch between the biased distribution of the systematic bit stream and the uniform input distribution needed to achieve the channel capacity. As we will show in this chapter, when some constraints are satisfied, recursive non-systematic convolutional (RNSC) encoders can generate asymptotically uniformly distributed output even for extremely biased sources. It is known that [24, 69] the capacity of a binary input AWGN or Rayleigh fading channel is achieved by a uniform i.i.d. channel input; therefore, we propose using RNSC encoders as the constituent Turbo encoders. Simulation results demonstrate substantial gains over systematic Turbo codes. The OPTA gaps for heavily biased sources are hence significantly reduced.

5.2 OPTA Loss in Systematic Turbo Codes

In this section, we justify the need to examine non-systematic Turbo codes for the transmission of non-uniform sources by evaluating the OPTA loss incurred in systematic Turbo codes due to the distribution mismatch between the systematic bit stream and the channel input.

For symmetric discrete memoryless channels, the capacity is achieved (the mutual information is maximized) by a uniform channel input. When the channel input is known to be non-uniform, the capacity cannot be fully exploited if systematic Turbo codes are used due to the distribution mismatch between the systematic bit stream and the capacity achieving distribution. As a result, the *achievable* OPTA in terms of E_b/N_0 solved by equation (4.3) would be smaller than the real OPTA, resulting in an OPTA loss. In this section, we compute analytically the achievable OPTAs for transmitting non-uniform i.i.d. sources using the systematic Turbo codes as studied in Chapter 4, and thus find the OPTA losses due to the systematic structure.

For a rate-1/3 systematic encoder as depicted in Fig. 2.2, since the channel inputs consist of one systematic sequence (which is non-uniform i.i.d.) and two parity sequences (which are uniform), the *achievable* capacity is

$$C_{sys} = \frac{1}{3}\left[2I(X;Y)|_{P(X=0)=1/2} + I(X;Y)|_{P(X=0)=p_0}\right], \tag{5.1}$$

where for binary-input continuous-output BPSK-modulated AWGN channels,

$$\begin{aligned} I(X;Y)|_{P(X=0)=p_0} &= \sum_x \int_{-\infty}^{\infty} p(y|x)P(x)\log_2\left[\frac{p(y|x)}{\sum_{x'} p(y|x')P(x')}\right] dy. \\ &= -p_0 \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(y+1)^2}{2\sigma^2}} \log_2(p_0 + p_1 e^{\frac{2y}{\sigma^2}})dy \\ &\quad -p_1 \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(y-1)^2}{2\sigma^2}} \log_2(p_0 e^{\frac{-2y}{\sigma^2}} + p_1)dy. \end{aligned} \tag{5.2}$$

where $p_1 = 1 - p_0$.

When the source is uniform i.i.d., as in Section 4.5, we have

$$I(X;Y)|_{P(X=0)=1/2} = 1 - \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(y+1)^2}{2\sigma^2}} \log_2(1 + e^{\frac{2y}{\sigma^2}}) dy. \tag{5.3}$$

Similarly, for binary-input continuous-output BPSK modulated Rayleigh fading channels with known channel state information, the above mutual informations become:

$$\begin{aligned}
I(X;Y)|_{P(X=0)=p_0} &= E_{(X,Y,A)} \log_2 \left[\frac{p(X,Y|A)}{p(X|A)p(Y|A)} \right] \\
&= \sum_x \int_0^{\infty} \int_{-\infty}^{\infty} p(y|x,a)P(x)p_A(a) \log_2 \left[\frac{p(y|x,a)}{\sum_{x'} p(y|x',a)P(x')} \right] dy\ da \\
&= -p_0 \int_0^{\infty} \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(y+a)^2}{2\sigma^2}} \cdot 2ae^{-a^2} \cdot \log_2 \left[p_0 + p_1 e^{\frac{2ay}{\sigma^2}} \right] dy\ da \\
&\quad -p_1 \int_0^{\infty} \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(y-a)^2}{2\sigma^2}} \cdot 2ae^{-a^2} \cdot \log_2 \left[p_0 e^{\frac{-2ay}{\sigma^2}} + p_1 \right] dy\ da.
\end{aligned} \tag{5.4}$$

For the uniform i.i.d. case, we have

$$I(X;Y)|_{P(X=0)=1/2} = 1 - \int_0^{\infty} \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(y+a)^2}{2\sigma^2}} \cdot 2ae^{-a^2} \cdot \log_2(1 + e^{\frac{2ay}{\sigma^2}}) dy\ da. \tag{5.5}$$

For a rate-1/2 systematic encoder, as obtained conventionally by puncturing half of the parity bits of a rate-1/3 systematic encoder, the achievable capacity is

$$C_{sys} = \frac{1}{2}[I(X;Y)|_{P(X=0)=1/2} + I(X;Y)|_{P(X=0)=p_0}]. \tag{5.6}$$

For a fixed rate r and a given BER P_e, the achievable OPTA (in terms of E_b/N_0) of a systematic Turbo code can then be solved (as shown in Section 4.5) by

$$r \cdot R(D) < C_{sys} \tag{5.7}$$

assuming equality. Due to the lack of a closed-form expression, the above inequality is solved numerically; it is computed by the bisection method in conjunction with numerical integrals or numerical double integrals for (5.2) to (5.5). The achievable OPTAs and the corresponding OPTA losses for transmitting non-uniform i.i.d. sources over AWGN and Rayleigh fading channels using systematic Turbo codes are provided in Tables 5.1 to 5.2.

We observe that as p_0 increases, the OPTA loss due to the systematic structure becomes more significant. We also observe that the OPTA losses of R_c=1/2 are generally greater than those of R_c=1/3; this is because for systematic Turbo codes, higher rates are conventionally achieved by puncturing parity bits, which makes the proportion of systematic part in the encoder outputs even greater. As we will see in the next section, under certain conditions, the parity sequence of a recursive convolutional encoder is asymptotically uniform. Therefore, the above observations provide the motivation to investigate non-systematic Turbo codes for the transmission of non-uniform i.i.d. sources.

R_c	p_0	OPTA	systematic OPTA	OPTA Loss
1/3	0.7	-1.19	-0.93	0.26
	0.8	-2.24	-1.64	0.60
	0.9	-4.40	-3.28	1.12
1/2	0.7	-0.62	-0.19	0.43
	0.8	-1.81	-0.80	1.01
	0.9	-4.14	-2.25	1.89

Table 5.1: OPTA losses in systematic code, AWGN channels.

R_c	p_0	OPTA	systematic OPTA	OPTA Loss
1/3	0.7	-0.34	-0.04	0.30
	0.8	-1.56	-0.89	0.67
	0.9	-3.96	-2.75	1.21
1/2	0.7	0.76	1.31	0.55
	0.8	-0.73	0.49	1.22
	0.9	-3.47	-1.34	2.13

Table 5.2: OPTA losses in systematic code, Rayleigh fading channels.

5.3 Asymptotic Uniformity of Recursive Convolutional Encoder Output for Non-Uniform I.I.D. Sources

In this section, we prove that for a non-uniform i.i.d. source, when certain condition is satisfied, a recursive convolutional encoder produces an asymptotically uniform output sequence, regardless of the value of $p_0 \in (0, 1)$. The condition is both necessary and sufficient. Based upon empirical observations, we also give a conjecture on the condition when the encoder output has higher order asymptotically uniform distributions. This asymptotic uniformity property can be employed as a design criterion for the construction of good Turbo encoders. We first establish the following lemma (which was also independently shown in [27] [Lemma, p. 1692]).

Lemma 5.1 For a given binary non-uniform i.i.d. sequence $\{A_k\}_{k=0}^{\infty}$ with probability distribution $P(A_k = 0) = p_0$ and $P(A_k = 1) = p_1 = 1 - p_0$, $k = 0, 1, 2, \cdots$, construct $\{B_k\}_{k=0}^{\infty}$, where

$$\begin{aligned} B_0 &= A_0, \\ B_k &= \sum_{i=0}^{k} A_i, \quad k = 1, 2, 3, \cdots, \end{aligned}$$

where the summation is modulo-2. Then

$$\lim_{k\to\infty} P(B_k = 0) = \frac{1}{2}.$$

Proof: Let us denote $q_0^{(k)} \triangleq P(B_k = 0)$ and $q_1^{(k)} \triangleq P(B_k = 1)$, then for $k \geq 1$, we have

$$
\begin{aligned}
q_0^{(k)} &= P(B_k = 0) \;=\; P(A_1 + A_2 + \cdots + A_k = 0) \\
&= P(A_1 + A_2 + \cdots + A_{k-1} = 0) \cdot P(A_k = 0) \\
&\quad + P(A_1 + A_2 + \cdots + A_{k-1} = 1) \cdot P(A_k = 1) \\
&= q_0^{(k-1)} p_0 + q_1^{(k-1)} p_1. \qquad (5.8)
\end{aligned}
$$

Similarly we have

$$q_1^{(k)} = q_0^{(k-1)} p_1 + q_1^{(k-1)} p_0. \qquad (5.9)$$

Rewrite (5.8) and (5.9) in the equivalent vector form as

$$\left[q_0^{(k)}, q_1^{(k)}\right] = \left[q_0^{(k-1)}, q_1^{(k-1)}\right] \begin{bmatrix} p_0 & p_1 \\ p_1 & p_0 \end{bmatrix},$$

by recursion, we can get

$$\left[q_0^{(k)}, q_1^{(k)}\right] = \left[q_0^{(0)}, q_1^{(0)}\right] \begin{bmatrix} p_0 & p_1 \\ p_1 & p_0 \end{bmatrix}^k,$$

for $k \geq 0$.

Now denote the matrix

$$\Pi \triangleq \begin{bmatrix} p_0 & p_1 \\ p_1 & p_0 \end{bmatrix}.$$

To compute Π^k, we first find its eigenvalues $\lambda_1 = 1$ and $\lambda_2 = 2p_0 - 1$, and the corresponding eigenvectors

$$\mathbf{v}_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad \mathbf{v}_2 = \begin{bmatrix} 1 \\ -1 \end{bmatrix},$$

denote

$$P \triangleq \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \quad D \triangleq \begin{bmatrix} 1 & 0 \\ 0 & 2p_0 - 1 \end{bmatrix},$$

then since $P^{-1}\Pi P = D$, we have

$$\Pi^k = PD^kP^{-1} = \frac{1}{2}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}\begin{bmatrix} 1 & 0 \\ 0 & (2p_0 - 1)^k \end{bmatrix}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}.$$

Therefore,

$$\begin{aligned}
\lim_{k\to\infty}\left[q_0^{(k)}, q_1^{(k)}\right] &= \lim_{k\to\infty}\left[q_0^{(0)}, q_1^{(0)}\right]\Pi^k, \\
&= \lim_{k\to\infty}\frac{1}{2}\left[1 + (2p_0 - 1)^{k+1}, 1 - (2p_0 - 1)^{k+1}\right] \\
&= \left[\frac{1}{2}, \frac{1}{2}\right].
\end{aligned}$$

$\square$

Remark: The above proof is to explicitly show that $\left[q_0^{(k)}, q_1^{(k)}\right]$ approaches the uniform distribution *exponentially* as the sequence length goes to infinity. For example, if $\left|q_i^{(k)} - 1/2\right| \leq 10^{-5}$ $(i = 0, 1)$ is desired, we only need to have $k \geq 52$. Realizing that $\{B_k\}_{k=0}^{\infty}$ is indeed a Markov chain, we may also prove Lemma 5.1 by showing that the uniform distribution is the stationary distribution of $\{B_k\}_{k=0}^{\infty}$.

Using the above lemma, we can now establish the following result.

Theorem 5.1 For a recursive convolutional encoder with feedback polynomial $F(D)$ and feed-forward polynomial $G(D)$, the necessary and sufficient condition that the encoder output is asymptotically uniform for a non-uniform i.i.d. source with distribution $p_0 \in (0,1)$ (regardless of the value of p_0) is that $G(D)$ is not divisible by $F(D)$ in GF(2).

Proof: a) *Necessary Part*:

Suppose the source generates $\{U_k\}_{k=0}^{\infty}$, and the encoder produces an output sequence of $\{X_k\}_{k=0}^{\infty}$. Denote the encoder input and output in polynomial forms as $U(D) = \sum_{k=0}^{\infty} u_k D^k$ and $X(D) = \sum_{k=0}^{\infty} x_k D^k$, respectively. Then for a recursive convolutional encoder with feedback polynomial $F(D)$ and feed-forward polynomial $G(D)$, we have

$$X(D) = U(D) \cdot \frac{G(D)}{F(D)} \stackrel{\triangle}{=} U(D) \cdot E(D).$$

When $G(D)$ is divisible by $F(D)$ in GF(2), $E(D)$ is a polynomial of D with finite degree. Then any bit of the encoder output X_k is essentially a modulo-2 summation of a finite number of bits from $\{U_k\}_{k=0}^{\infty}$. However, a finite modulo-2 summation of non-uniform i.i.d. variables would still be non-uniform; therefore, $\{X_k\}_{k=0}^{\infty}$ is still non-uniform.

b) *Sufficient Part*:

When $G(D)$ is not divisible by $F(D)$ in GF(2), $E(D)$ is an infinite polynomial of D,

$$E(D) = e_0 + e_1 D + e_2 D^2 + \cdots + e_k D^k + \cdots ,$$

where $e_k, k = 0, 1, 2, \cdots$ is binary. Since $E(D) = G(D)/F(D)$ is a rational polynomial fraction, when $G(D)$ is not divisible by $F(D)$ in GF(2), $\{e_l\}$ is a periodic series. When $k \to \infty$, the number of 1s in $\{e_k\}$ also goes to infinity. Therefore, X_k is the modulo-2 summation of an infinite number of bits from $\{U_k\}_{k=0}^{\infty}$. From Lemma 5.1 we obtain that

$$\lim_{k \to \infty} P(X_k = 0) = 1/2.$$

$\square$

In [67], Shamai and Verdú proved that for any fixed positive integer k, the k^{th}-order empirical distribution of any good code (i.e., a code approaching capacity with asymptotically vanishing probability of error) converges to the input distributions that achieve channel capacity. The capacity of a binary input AWGN or Rayleigh fading channel is achieved when its mutual information is maximized by a uniformly distributed input. Therefore, having higher order uniform distributions (in addition to the first order distribution) in encoder outputs is also desirable. Observations through extensive simulations lead us to the following conjecture.

Conjecture: For a recursive convolutional encoder with feedback polynomial $F(D)$ and feed-forward polynomial $G(D)$, suppose $G(D)/F(D)$ is in its minimal form and that $F(D)$ has degree M, then the m^{th}-order distribution of the encoder output is asymptotically uniform for any $p_0 \in (0,\, 1)$ and $\forall\; m = 1, 2, \cdots, M$.

A proof for the above conjecture with arbitrary $G(D)$ and $F(D)$ is not obvious. However, the conjecture has been verified by empirical estimation of all possible combinations of recursive convolutional encoders with memory 4.

5.4 Non-Systematic Turbo Codes

As shown in Section 5.2, when non-uniform i.i.d. sources get heavily biased, a large OPTA loss is incurred due to the systematic structure. Justified by Shamai and Verdú's result, Theorem 5.1 and the conjecture in Section 5.3 indicate that non-systematic Turbo codes seem to be a suitable solution in this scenario and great potential gains can be expected.

5.4.1 Design of Good Encoder Structures

Fig. 5.1 shows our proposed non-systematic Turbo encoders. In a) the first constituent encoder has two parity outputs X_k^{1h} and X_k^{1g} while the second has only one parity output X_k^{2g}; so the overall rate is 1/3. In b) both constituent encoders have two parity outputs and the overall rate is 1/4. Structure b) can achieve the same overall rate of 1/3 by puncturing. Structure a) is actually a special case of structure b) obtained

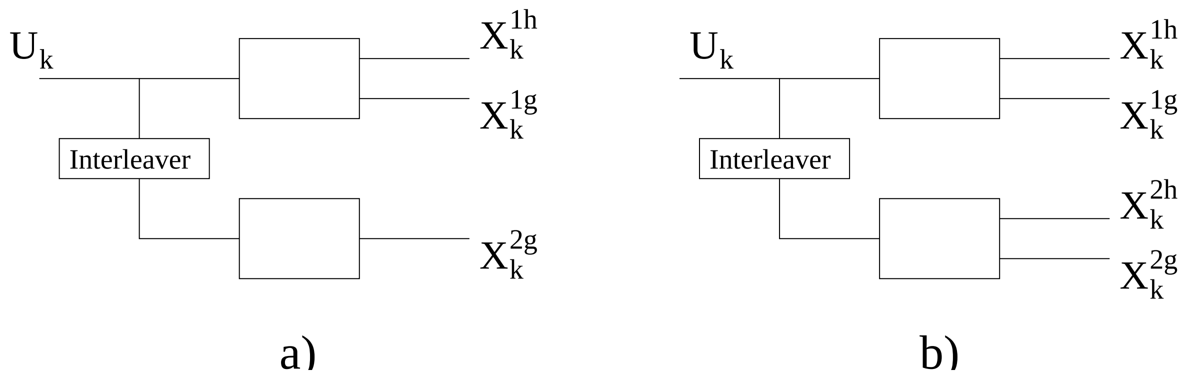

Figure 5.1: Non-Systematic Turbo encoder structures.

by completely puncturing X_k^{2h}; therefore, a generally designed decoder for structure b) can also be used for structure a). A detailed diagram of structure b) is shown in Fig. 5.2.

Ideally, good design criteria are analytically based on the minimization of the error probability. However, to the best of our knowledge, all available error bounds for Turbo codes are obtained by averaging over a code ensemble or by assuming uniform interleaving and maximum likelihood decoding, while Turbo codes in fact employ random interleaving and maximum *a posteriori* decoding. Furthermore, the bounds are useful only in the error floor region at high signal-to-noise ratios (SNRs), but are not tight in the waterfall region [13, 26, 68]. Our goal is, however, to obtain the best waterfall performance. Hence, we need to use other tools to find good design criteria.

As in Chapter 4, we focus on symmetric 16-state encoders. First, as proved by

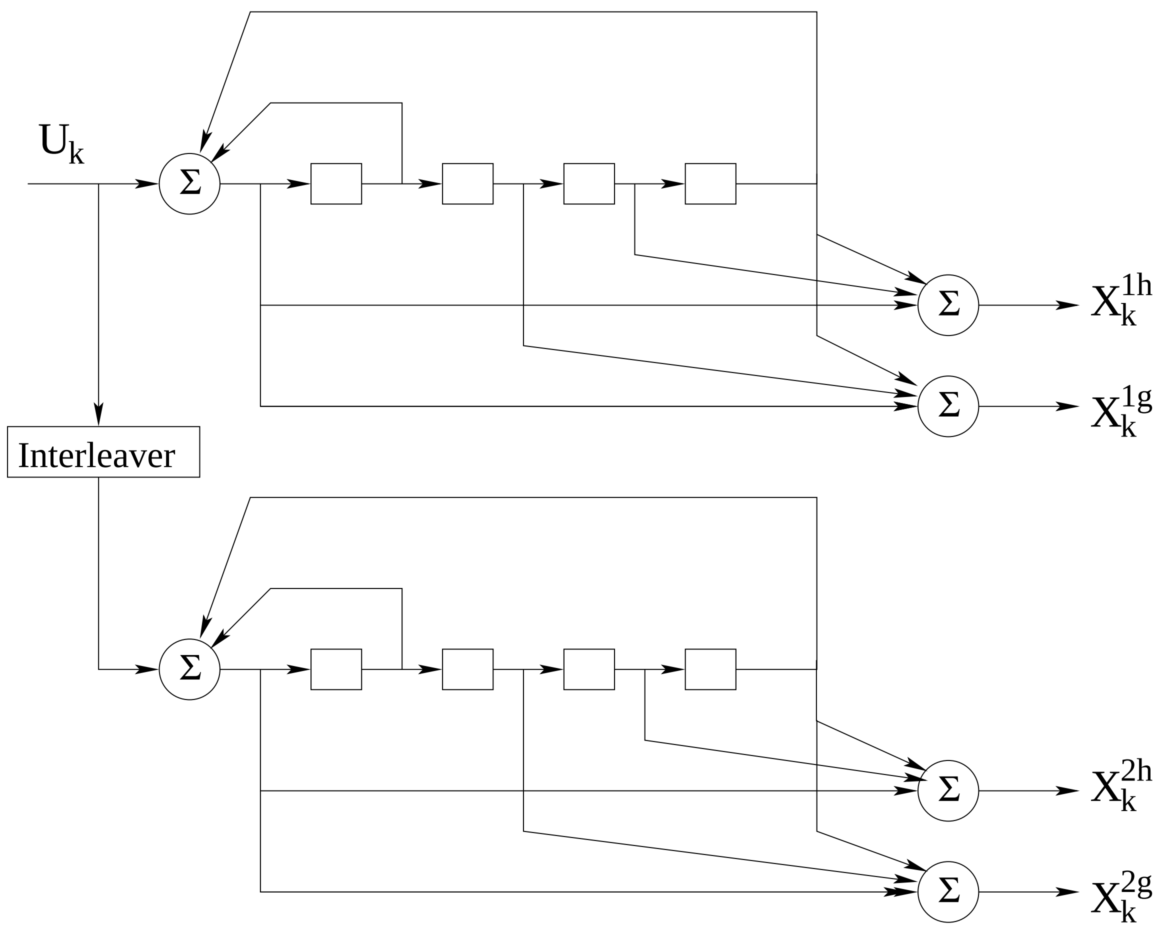

Figure 5.2: Symmetric non-systematic Turbo encoder structure.

Theorem 4.1 in Section 4.3, we choose to have the last feedback tap coefficient $f_4 = 1$ to fully exploit the encoder memory. Also, we choose $g_0 = 1$ besides the given $f_0 = 1$. This would reduce the total number of possible encoders in our search space down to $2^3 \times 2^4 \times 2^4 = 2048$, which is still impractical for an exhaustive search. Second, according to Theorem 5.1, we would only consider the choices of $G(D)$ and $F(D)$ such that $G(D)$ is not divisible by $F(D)$. Furthermore, by our empirically verified conjecture, given $f_4 = 1$, we choose $F(D)$ and $G(D)$ such that the greatest common divisor of $F(D)$ and $G(D)$ is 1, we can guarantee the encoder outputs have up to the 4^{th} order distribution asymptotically uniform. Thus additional inferior candidate encoder structures can be eliminated. Finally, again to avoid an exhaustive search, we take advantage of the optimization results found in systematic Turbo codes (see Section 4.3); i.e., we first find the best pair of $F(D)$ and $G(D)$, and then search for the best second feed-forward polynomial $H(D)$.

5.4.2 Decoder Modifications

When RNSC encoders are used as constituent encoders, unlike in systematic Turbo codes, the LLR $\Lambda(U_k)$ in the BCJR algorithm can only be decomposed into two separate terms as follows:

$$\Lambda(U_k) = L_{ex}(U_k) + L_{ap}(U_k),$$

where the new extrinsic term involves two parity sequences. For AWGN channels, we have

$$L_{ex}(U_k) = \log \frac{\sum_s \sum_{s'} \gamma(y_k^h, y_k^g|1, s, s') \cdot \alpha_{k-1}(s') \cdot \beta_k(s)}{\sum_s \sum_{s'} \gamma(y_k^h, y_k^g|0, s, s') \cdot \alpha_{k-1}(s') \cdot \beta_k(s)},$$

where for $i = 0, 1$,

$$\begin{aligned} \gamma(y_k^h, y_k^g|i, s, s') &= p(y_k^h|U_k = i, S_k = s) \cdot p(y_k^g|U_k = i, S_k = s) \\ &\quad \cdot Pr\{U_k = i|S_k = s, S_{k-1} = s'\}, \end{aligned}$$

and where S_k is the encoder state at time k, y_k^h and y_k^g are the noise corrupted versions of x_k^h and x_k^g, respectively, which are the parity bits generated from the two feed-forward polynomials.

For Rayleigh fading channels, the extrinsic term needs to be modified to appropriately incorporate the channel statistics. The extrinsic term therefore becomes

$$L_{ex}(U_k) = \log \frac{\sum_{s,s'} \gamma(y_k^h, y_k^g|1, s, s', a_k^h, a_k^g) \cdot \alpha_{k-1}(s') \cdot \beta_k(s)}{\sum_{s,s'} \gamma(y_k^h, y_k^g|0, s, s', a_k^h, a_k^g) \cdot \alpha_{k-1}(s') \cdot \beta_k(s)},$$

where a_k^h and a_k^g are the fading factors, and for $i = 0, 1$,

$$\begin{aligned} \gamma(y_k^h, y_k^g|i, s, s', a_k^h, a_k^g) &= p(y_k^h|U_k = i, S_k = s, a_k^h) \cdot p(y_k^g|U_k = i, S_k = s, a_k^g) \\ &\quad \cdot Pr\{U_k = i|S_k = s, S_{k-1} = s'\}. \end{aligned}$$

When the source is non-uniform i.i.d., $\log((1-p_0)/p_0)$ is used as the initial *a priori* input to the first decoder at the first iteration; then at the following iterations, $L_{ex} + \log((1-p_0)/p_0)$ is used as the new *a priori* information for both constituent decoders at each iteration.

5.5 Numerical Results and Discussion

In this section, we present simulation results of our non-systematic Turbo codes for non-uniform i.i.d. sources over BPSK-modulated AWGN and Rayleigh fading channels with known channel state information. The performance is measured in terms of bit error rate (BER) versus E_b/N_0, where E_b is the average energy per source bit and $N_0/2$ is the variance of the Gaussian additive noise process. All simulated Turbo codes have 16-state constituent encoders and use the same pseudo-random interleaver introduced in [15]. The sequence length is $L = 512 \times 512 = 262144$ and at least 200 blocks are used; this would guarantee a reliable BER estimation at the 10^{-5} level with 524 errors. The number of iterations used in the decoder is 20. All presented results are for Turbo codes with structure b) encoders as they have a better performance than the codes with structure a) encoders. Simulations are performed for rates $R_c = 1/3$ and $R_c = 1/2$ with p_0=0.8 and 0.9. From our simulations, for both rates 1/3 and 1/2, the best RNSC encoder structure found for p_0=0.8 and for both channels has each constituent encoder with the feedback polynomial 35 and feed-forward polynomials 23 and 25, denoted by (35,23,25); for p_0=0.9 the best structure is (31,23,27). Several other encoders give very competitive performance; for example, (35,23,31) for p_0=0.8, (31,23,35) and (31,23,37) for p_0=0.9 also give a good performance that is very close to the one offered by the above encoders.

Fig. 5.3 shows the performances of our rate-1/3 non-systematic Turbo codes over

AWGN channels in comparison with their systematic peers investigated in Chapter 4, as well as with Berrou's (37,21) code, which offers the best waterfall performance (among 16-state encoders) for uniform i.i.d. sources. At the 10^{-5} BER level, when p_0=0.8, our (35,23,25) non-systematic Turbo code offers a 0.45 dB gain over its (35,23) systematic peer, and the OPTA gap is thus reduced down to only 0.74 dB; when p_0=0.9, the improvement is 0.89 dB with the encoder structure (31,23,27), and the OPTA gap therefore becomes 1.13 dB. In comparison with Berrou's (37,21) code performance, the gains achieved by exploiting the source redundancy and encoder optimization are therefore 1.48 dB and 3.25 dB for p_0=0.8 and 0.9, respectively.

Fig. 5.4 shows similar results over AWGN channels for rate-1/2. We observe that the gains are generally more significant. In comparison with the best systematic Turbo code performances, at the 10^{-5} BER level, for p_0=0.8 and 0.9, the gains achieved are 0.69 dB and 1.56 dB, respectively, and the new OPTA gaps are 0.87 dB and 1.05 dB, respectively. Furthermore, the gains over Berrou's code due to combining the optimized encoder with the modified decoder that exploits the source redundancy are 1.57 dB (p_0=0.8) and 3.72 dB (p_0=0.9).

Performance comparisons over Rayleigh fading channels are also provided in Figures 5.5 and 5.6. For rate-1/3, at the 10^{-5} BER level, when p_0=0.8, our (35,23,25) non-systematic Turbo code offers a 0.40 dB gain over its (35,23) systematic peer, which brings the performance only 0.88 dB away from OPTA; when p_0=0.9, the improvement is 1.01 dB with the encoder structure (31,23,27), which narrows the OPTA gap from 2.18 dB down to 1.17 dB. In comparison with Berrou's (37,21) code performance, the gains achieved by exploiting the source redundancy and encoder optimization are therefore 1.76 dB and 3.87 dB for p_0=0.8 and 0.9, respectively. For rate-1/2, as shown in Fig. 5.6, we observe that the gains are generally more significant. In comparison with the best systematic Turbo code performances, at the 10^{-5} BER level, for p_0=0.8 and 0.9, the gains achieved are 0.77 dB and 1.84 dB, respectively; the OPTA gaps are therefore reduced to 1.11 dB and 1.15 dB, respectively. Furthermore, the gains due to combining the optimized encoder with the modified decoder that exploits the source redundancy are 2.01 dB (p_0=0.8) and 4.71 dB (p_0=0.9). The OPTA gaps are provided in Tables 5.3 and 5.4.

In the works of non-systematic Turbo codes for uniform i.i.d. sources, it has been verified via extensive simulations that most of the non-systematic Turbo codes show inferior performance to their systematic peers [21, 22, 23, 57], except the one found in [57] employing a non-systematic "quick-look-in" constituent code, which is basically "close" to a systematic code. Also, it has been pointed out in [23, 57] that at low SNRs, the initial extrinsic estimates for the information bits provided by non-systematic Turbo codes are not as good as those of systematic Turbo codes due to the lack of received channel values; this motivates their choice of a close-to-systematic code in the non-systematic family. However, as shown in Section 5.2, for heavily biased non-uniform i.i.d. sources, systematic Turbo codes result in relatively big OPTA losses; therefore, close-to-systematic may not be desired in the scenario of non-uniform i.i.d. source. Additionally, another encoder structure called the "big-numerator-little-denominator" (BNLD) code proposed in [22], which contributed to gains over Berrou's code for uniform i.i.d. sources, again may not be desirable for non-uniform i.i.d. sources because the little denominator results in non-uniform high-order output distributions.

In the scenario of non-uniform i.i.d. sources, there are two things playing against each other: the *a priori* knowledge of p_0, and the systematic structure. When the source is heavily biased, the systematic structure results in a big OPTA loss due to the distribution mismatch between the source and the capacity achieving channel input. Furthermore, the knowledge of a biased p_0 can give good initial extrinsic estimations for the information bits at the decoder, thus eliminating the need for the systematic structure. This may explain why non-systematic Turbo codes, which do not suffer from any distribution mismatch (at least asymptotically), offer superior performance over their systematic counterparts. On the other hand, when the source distribution gets close to uniform, very little *a priori* knowledge of p_0 is available; the systematic bits, however, even when corrupted by channel noise, can provide more reliable extrinsic estimations in the initial iterative decoding stages at low SNRs, while the OPTA loss becomes less significant. In particular, as pointed out in [21, 22, 23, 57], when the source is uniform, there is no *a priori* knowledge of p_0 at all; thus the systematic structure would play an important role. To investigate the role of the systematic bits for less biased non-uniform i.i.d. sources, we also studied the performance of so-called "extended" systematic Turbo codes.

The encoder of an "extended" systematic Turbo code consists of one systematic output, two RNSC encoders, each of which produces two parity outputs. The overall rate is therefore 1/5. Higher rates (e.g., 1/3 and 1/2) are obtained by partial puncturing of the systematic part and partial puncturing of the four parity sequences.

For p_0=0.6, 0.7 and 0.8, and for R_c=1/2 and 1/3, we performed simulation using the following puncturing patterns: delete $j/8$ of the systematic sequence, where $j = 0, 1, 2, \cdots, 8$, and delete evenly the four parity sequences to maintain the desired overall rates. Our simulations show that when p_0=0.7, starting from $j = 0$, the performance gets improved as j increases up to $j = 4$, which yields approximately a 0.1 dB gain at the 10^{-5} BER level over that of $j = 0$, then the performance degrades as j increases from 4 to 8. Therefore, when p_0=0.7, puncturing half of the systematic sequence yields the best performance. When p_0=0.8, we observe monotonic performance improvement as j increases from 0 up to 8, which indicates that purely non-systematic Turbo codes should be preferred. When p_0=0.6, we observe exactly the opposite: preserving all systematic gives the best performance. This agrees with the works of [21, 22, 23, 57] that systematic bits are especially important when not enough *a priori* knowledge of the source distribution is available.

To achieve a desired rate by puncturing, using different puncturing patterns may result in different performances. For example, when structure b) is used for an overall rate of 1/3, we may choose to puncture 1/4 of each parity sequence according to various patterns, or we may puncture half of two parity sequences, and leave the other two sequences intact. Simulations show the best puncturing pattern is to keep the parity sequence generated from feed-forward 23 intact and puncture half of the one generated from the other feed-forward. The performance of this puncturing pattern is about 0.2 dB better than other patterns; in particular it is 0.3 dB better than the performance offered by structure a). For an overall rate of 1/2, structure b) is also better than a), and the best puncturing pattern is to delete all even (odd) position bits of the sequences generated from feed-forward 23, and delete all odd (even) position bits of the sequences generated from the other feed-forward polynomial.

As a final remark to this section, we should point out that the value of p_0 does not have to be known to the Turbo code decoder, since at each iteration, it can be estimated from the decoded sequence and thus be used as part of the *a priori* information for the next iteration. The estimated value of p_0 is updated at each new iteration. This issue was earlier addressed in [35] for estimating parameters of hidden Markov sources in the Turbo code decoder, where almost no performance loss was observed.

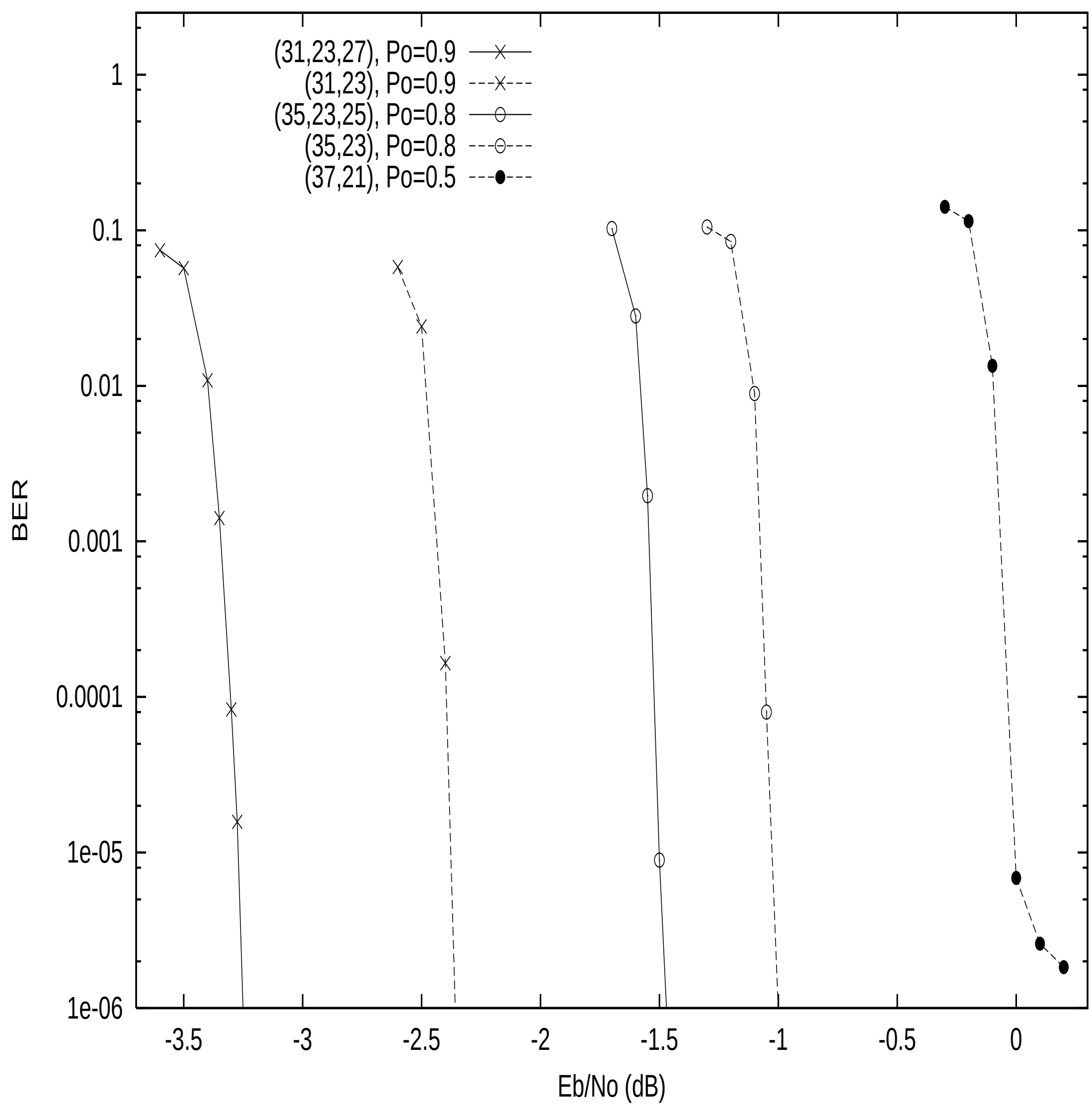

Figure 5.3: Turbo codes for non-uniform i.i.d. sources, R_c=1/3, L=262144, AWGN channel.

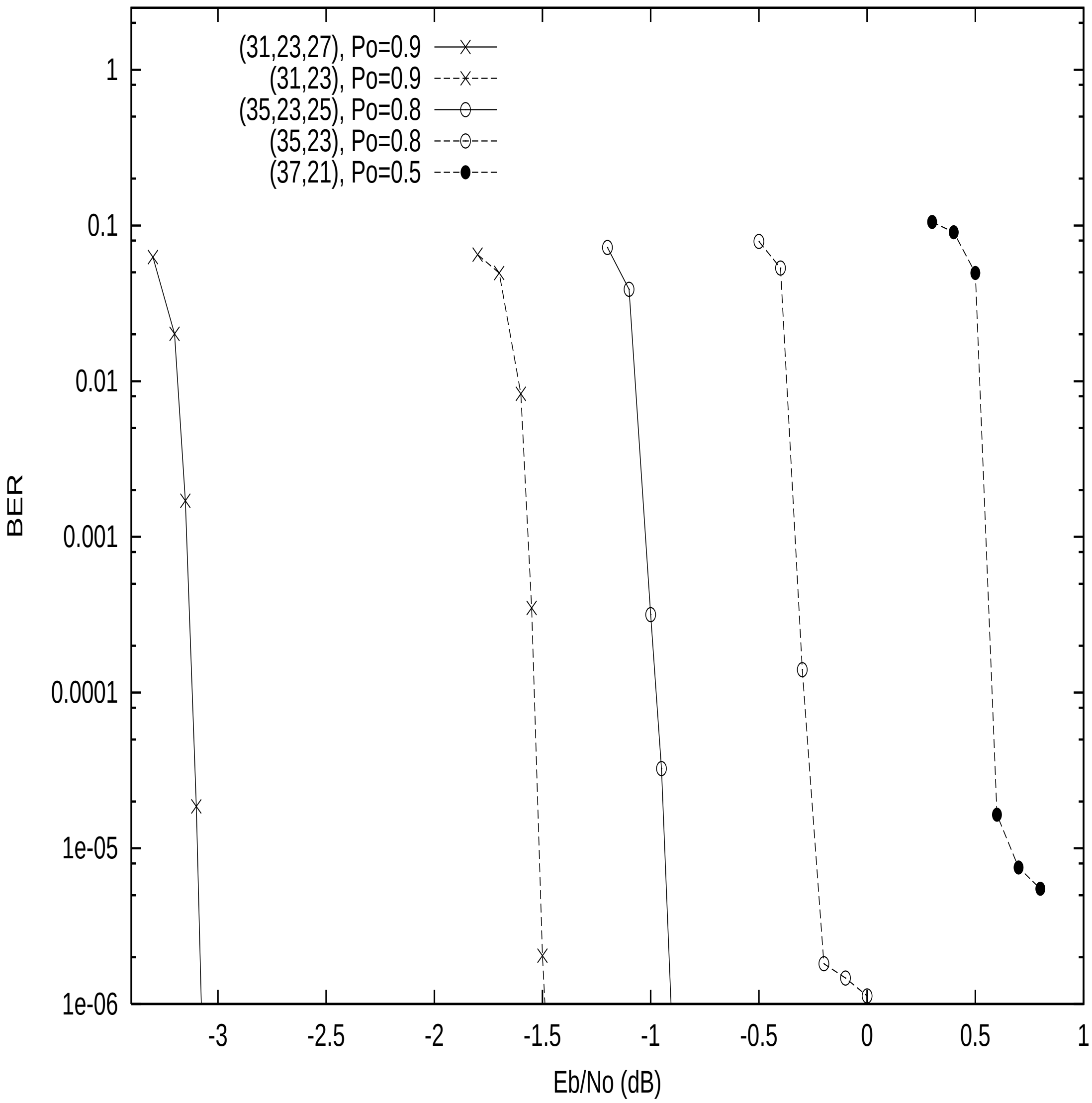

Figure 5.4: Turbo codes for non-uniform i.i.d. sources, R_c=1/2, L=262144, AWGN channel.

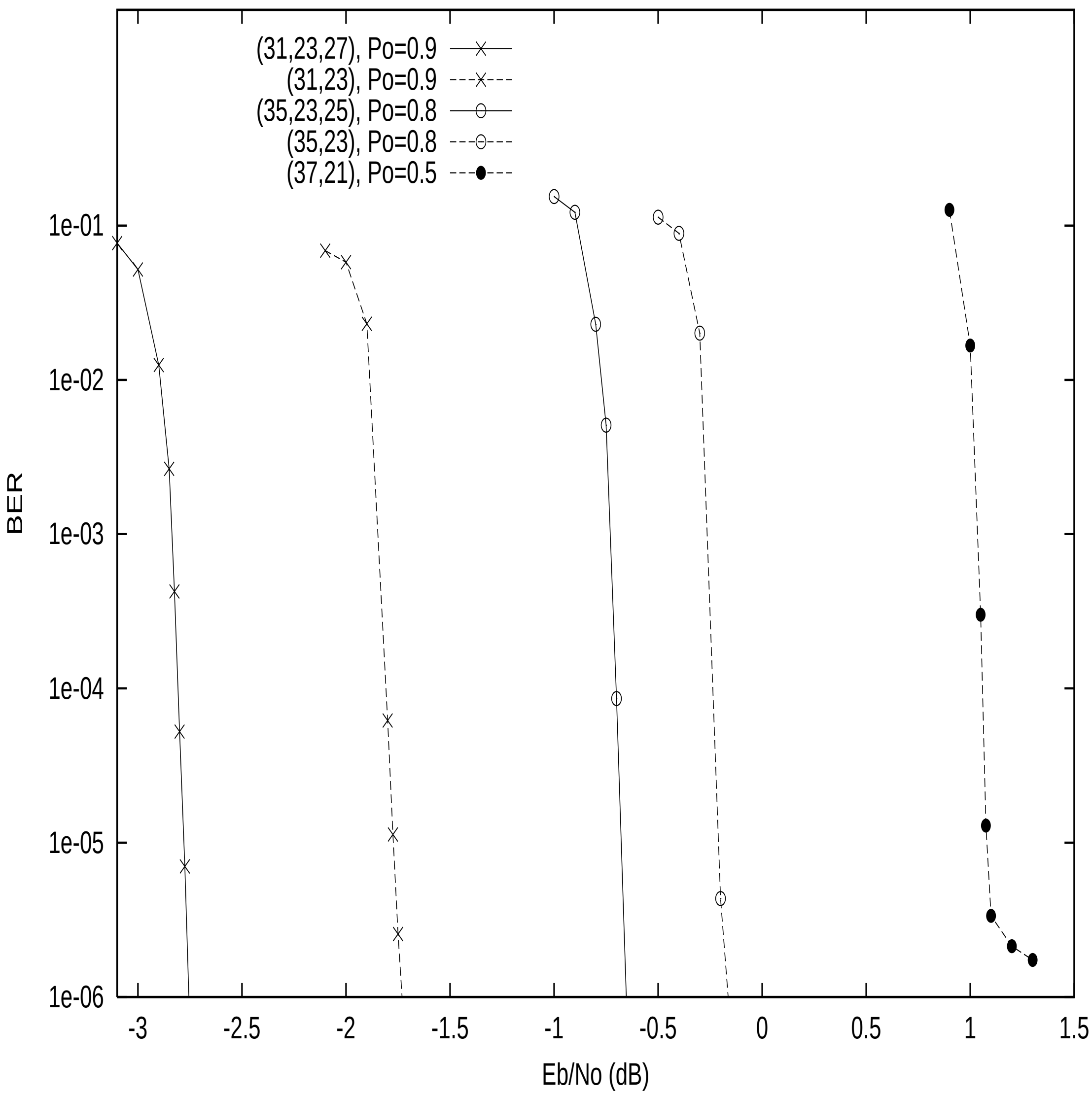

Figure 5.5: Turbo codes for non-uniform memoryless sources, R_c=1/3, L=262144, Rayleigh fading channel.

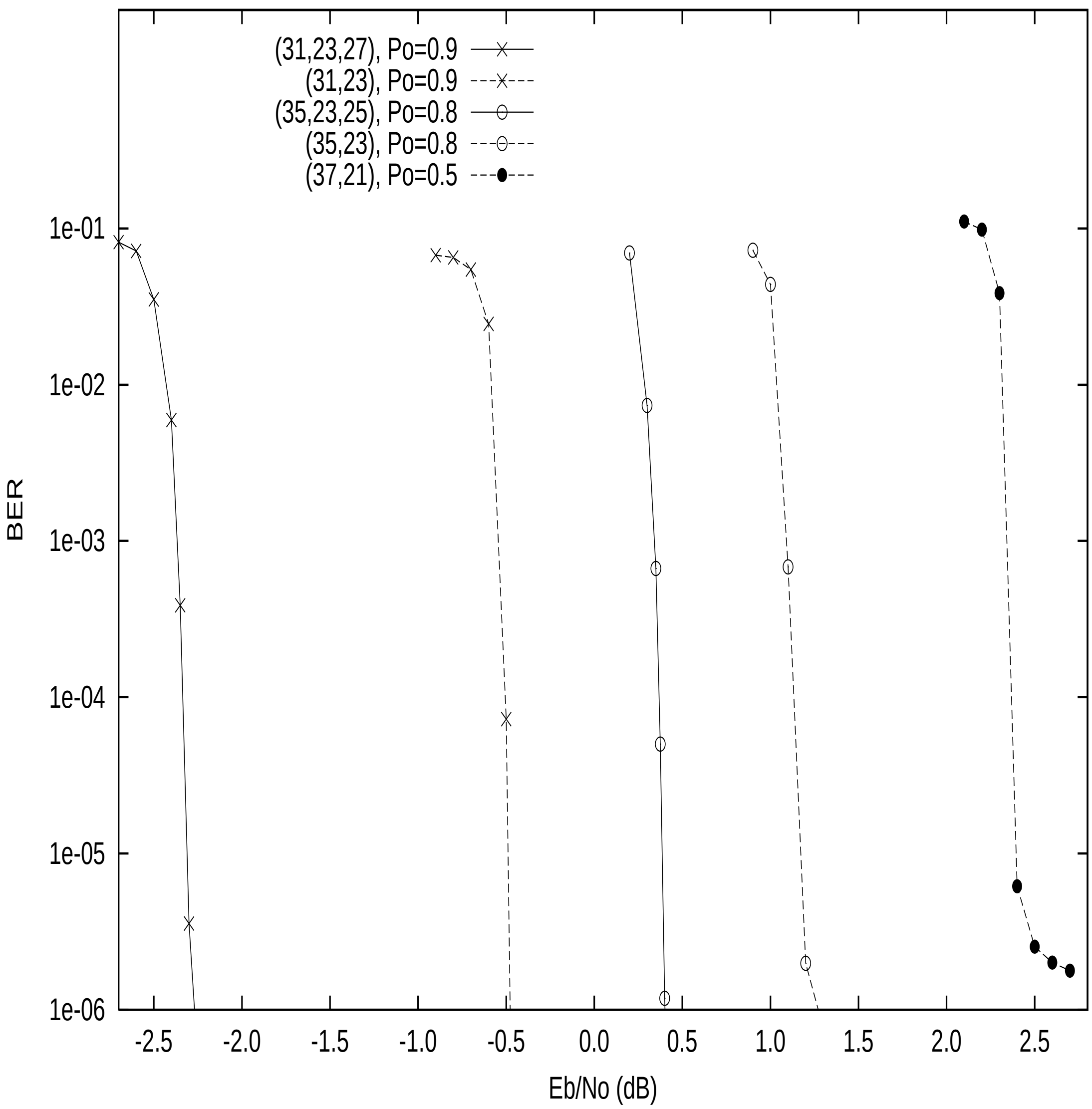

Figure 5.6: Turbo codes for non-uniform memoryless sources, R_c=1/2, L=262144, Rayleigh fading channel.

Source distribution	Systematic		Non-Systematic	
	$R_c = 1/2$	$R_c = 1/3$	$R_c = 1/2$	$R_c = 1/3$
$p_0 = 0.8$	1.56	1.19	0.87	0.74
$p_0 = 0.9$	2.61	2.02	1.05	1.13

Table 5.3: OPTA gaps in E_b/N_0 at BER=10^{-5} level (in dB), AWGN channel.

Source distribution	Systematic		Non-Systematic	
	$R_c = 1/2$	$R_c = 1/3$	$R_c = 1/2$	$R_c = 1/3$
$p_0 = 0.8$	1.88	1.28	1.11	0.88
$p_0 = 0.9$	2.99	2.18	1.15	1.17

Table 5.4: OPTA gaps in E_b/N_0 at BER=10^{-5} level (in dB), Rayleigh fading channel.

5.6 Comparison with Tandem Schemes

A tandem scheme consists of a source coding and a channel coding system designed independently. That is, the source is compressed first, and then channel coded assuming the output from the source encoder is already uniform memoryless. Although in theory, when the sequence length approaches infinity, there is no loss of optimality due to the separate design, this is seldom the case in practice.

In this section, we compare the performance of our joint source-channel system with that of two tandem schemes for the same overall transmission rate. Each tandem scheme here consists of a 4^{th}-order Huffman code followed by a rate R_c=1/3 Turbo code. The overall rate for both tandem and joint source-channel coding systems is $r = R_c/R_s$=1/2 source symbol/channel symbol (in the joint source-channel coding scheme, $R_c = 1/2$ and $R_s = 1$ since no source coding is performed). Therefore, the Huffman code needs to be of rate R_s=2/3 code bits/source symbol. Since the average rate of the Huffman code depends on the source probability distribution, we need to find the value of p_0 which renders the Huffman code rate (not the entropy) R_s=2/3 code bits/source symbol with satisfactory accuracy. By using the bisection method, we can find that when p_0=0.83079, a 4^{th}-order Huffman code has rate R_s=0.666668 code bits/source symbol. Note that the 4^{th}-order Huffman code performs near-optimal data compression since R_s is very close to the entropy (0.656) of the non-uniform i.i.d. source with p_0=0.83079. Therefore, in this section, simulations are performed for this value of p_0.

Berrou's pseudo-random interleaver [15] requires that the sequence length has to be an even power of 2; this inflexibility becomes an obstacle in the design of the tandem scheme, since the Huffman code is a variable-length code. The S-random interleaver [25], however, can take an input sequence with arbitrary length and yields good BER performance; therefore, in this section we adopt the S-random interleaver in the Turbo encoder and decoder. Another small modification to the Turbo code is that the first constituent encoder is not terminated; because otherwise errors in the tail bits of the Turbo-decoded sequence would introduce irrecoverable errors in the Huffman-decoded sequence. For fair comparison, our system also adopts the S-random interleaver, and also with the first constituent encoder not terminated.

The tandem scheme is implemented as follows:

1) the source generates a non-uniform i.i.d. sequence with length L, and p_0=0.83079;

2) the Huffman encoder produces a compressed sequence with variable length, whose mean is approximately $(2/3)L$;

3) an S-random interleaver is generated for this given length;

4) the sequence is Turbo-encoded using the S-random interleaver generated in 3);

5) the sequence is BPSK modulated and transmitted over a Rayleigh fading channel;

6) the sequence is Turbo-decoded and then Huffman-decoded.

Another issue is the choice of the source sequence length L. Due to error propagation in the Huffman decoder, a few errors in the Turbo-decoded sequence could result in a big percentage of errors in the final Huffman-decoded sequence. On the other hand, what matters is not only the number of errors in the Turbo-decoded sequence, but also the positions of the erroneous bits. Considering how Huffman decoding is performed, we can see that an error among the first few bits in an input sequence is much more disastrous than an error in the tail bits. Therefore, a sufficiently large number of blocks is necessary to obtain a good average performance. Considering computation limitations and after several trials, we chose to use L=12000, and 60000 blocks. The number of iterations in the Turbo decoder is also 20.

For a given sequence length, the larger we choose the "spread" S of the S-random interleaver, the better performance we can achieve. However, in practice, generating an S-random interleaver with a large spread requires a substantial amount of computation time, and sometimes such an S-random interleaver may not be generated successfully. Therefore, in order to reduce the computation time and also to guarantee the successful generation of S-random interleavers of arbitrary size, the spread S is chosen to be only 10.

Figure 5.7 shows the performance comparison of our system with two tandem schemes over Rayleigh fading channels. The comparison is made in terms of BER versus E_b/N_0. In the first tandem scheme, the rate R_c=1/3 Turbo code we simulated is Berrou's (37,21) code, which offers an excellent waterfall performance for uniform sources among all 16-state codes. However, due to a relatively high error-floor provided by Berrou's code, this tandem scheme suffers from a high-BER performance caused by error propagation in the Huffman decoder. Thus, we also provide simulation results of a second tandem scheme using the (35,23) Turbo code, which has a significantly lower error-floor at the expense of a slight waterfall performance loss. Although at very high BER levels, both tandem schemes offer better performance than that of our joint source-channel coding system, their error-floors occur at high BER levels (10^{-3} for the (37,21) code, and 10^{-4} for the (35,23) code). Therefore, at low BER levels, our joint source-channel coding system offers superior performance than both tandem schemes. Interestingly, most traditional joint source-channel coding schemes outperform tandem schemes at high BER levels, while here the opposite result is observed. Recently, alternative joint source-channel coding schemes in the context of Huffman codes with Turbo codes are proposed in [40, 42]. It would be interesting to make performance comparisons with these schemes. However, our system has lower complexity, since the source encoding and decoding parts are omitted.

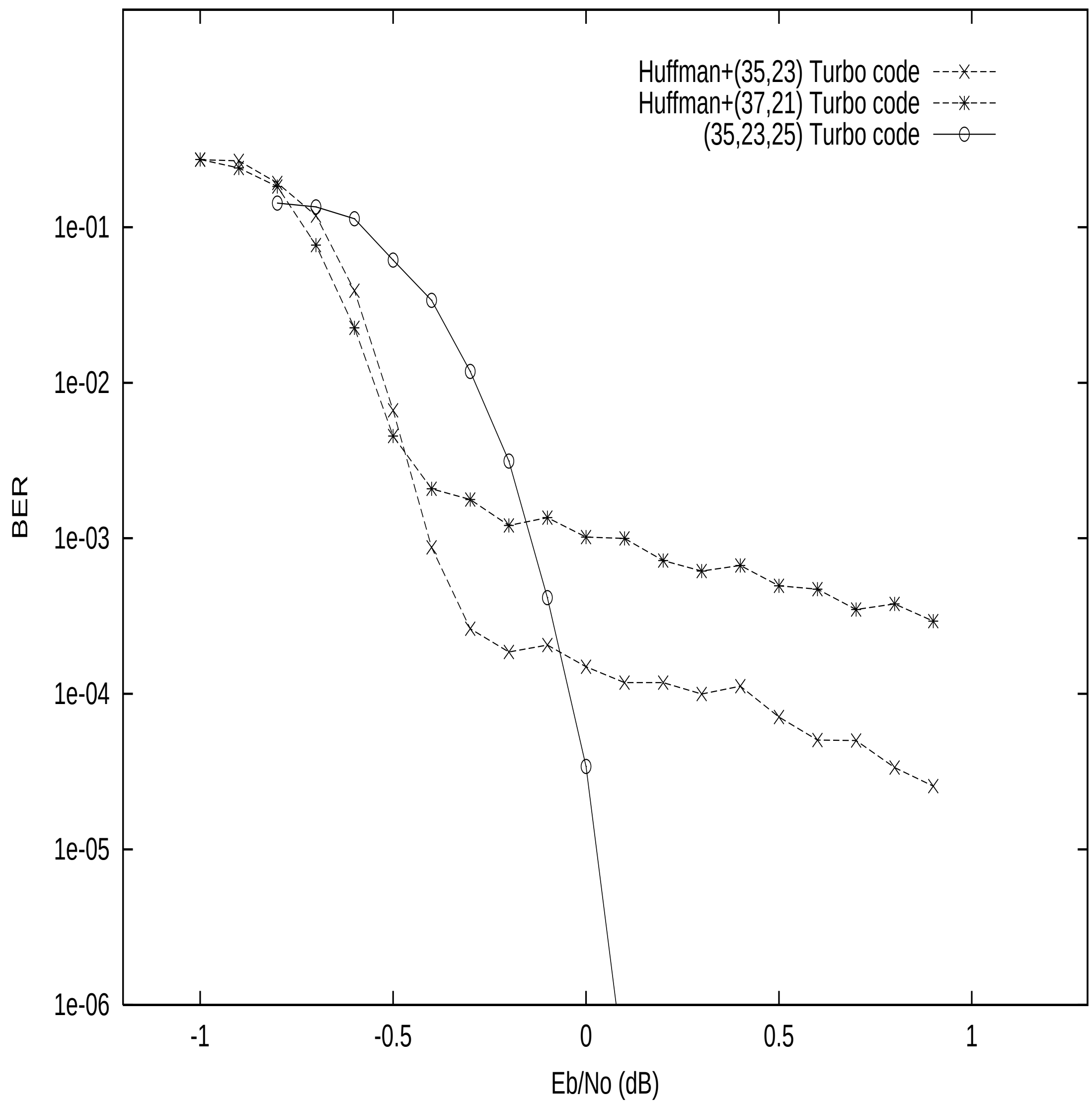

Figure 5.7: Performance comparison of our system ($R_s = 1, R_c = 1/2$) with that of tandem schemes ($R_s = 2/3$, $R_c = 1/3$), non-uniform i.i.d. sources, p_0=0.83079, L=12000, Rayleigh fading channel.

Chapter 6

Turbo Codes for Binary Markov Sources

6.1 Introduction

As mentioned in Chapter 1, the source redundancy can take two different forms: non-uniformity and memory. The issue of designing Turbo codes for non-uniform i.i.d. sources has been addressed in Chapters 4 and 5. In this chapter, we investigate how the source redundancy in the form of memory can be exploited in the decoder of Turbo codes.

We consider a stationary ergodic binary first-order Markov source $\{U_k\}_{k=1}^{\infty}$. The

Markov chain is described by the transition matrix:

$$\Pi = [\pi_{ij}] = \begin{bmatrix} q_0 & 1-q_0 \\ 1-q_1 & q_1 \end{bmatrix},$$

where the transition probability is defined by

$$\pi_{ij} \triangleq Pr\{U_k = j | U_{k-1} = i\}, \quad i,\ j \in \{0,1\}.$$

Also, define the marginal distribution of the source as $p_0 \triangleq Pr\{U_k = 0\}$, and $p_1 \triangleq Pr\{U_k = 1\}$. By stationarity, it can be easily shown that p_0 and p_1 are given by the stationary distribution:

$$p_0 = 1 - p_1 = \frac{1-q_1}{2-q_0-q_1}. \tag{6.1}$$

In general, the source described above is an *asymmetric* Markov source; its redundancy is in the form of both memory and non-uniformity. When $q_0 = 1 - q_1$, the source reduces to a memoryless source with marginal distribution $p_0 = q_0$ and $p_1 = q_1$; in this case the source redundancy is purely in the form of non-uniformity. When $q_0 = q_1 \neq 1/2$, the source becomes a *symmetric* Markov chain with a *uniform* marginal distribution, i.e., $p_0 = p_1 = 1/2$; the source redundancy is therefore strictly in the form of memory. When the source is symmetric, we denote $q_0 = q_1 = q$.

6.2 Shannon Limit

Shannon's *Lossy Information Transmission Theorem* described in Section 4.5 was originally proved for transmitting memoryless sources over memoryless channels; however, it is also applicable when the source and the channel additive noise are stationary ergodic [7]. In general, for asymmetric Markov sources, there are no closed-form expressions for the rate-distortion function $R(D)$. In this case, Blahut algorithm [17] can numerically give relatively tight upper and lower bounds for $R(D)$. Therefore, for a desired BER level and a given overall rate $r = R_c/R_s$ in source symbol/channel symbol where R_s and R_c are the source coding and channel coding rates, respectively, we may substitute $R(D)$ with its upper or lower bound in

$$r \cdot R(D) < C(E_b/N_0),$$

and hence obtain an upper or lower bound on the corresponding Shannon limit.

For a special case, when the source is binary *symmetric* Markov with transition probability

$$Pr\{U_{k+1} = j | U_k = i\} = q, \quad \text{for } i \neq j, \tag{6.2}$$

and assuming $q > 1/2$ without loss of generality, Gray proved that [38]

$$R(D) = H_b(q) - H_b(D), \quad \text{if } 0 \leq D \leq D_c,$$

where D_c is the *critical distortion* defined by

$$D_c = \frac{1}{2}\left(1 - \sqrt{1 - \frac{(1-q)^2}{q^2}}\right).$$

Therefore, when the desired distortion D is no greater than the critical distortion D_c, $R(D)$ admits a closed-form expression identical to that of a non-uniform i.i.d. source with $Pr\{U_k = 0\} = q$. Gray's expression for $R(D)$ (which is also known as Shannon's lower bound) was noted by Berger [8] to be a good lower bound for $D_c \leq D \leq D_2$:

$$R(D) \geq H_b(q) - H_b(D), \quad \text{if } D_c \leq D \leq D_2,$$

where

$$D_2 = \frac{1}{2}\left(1 - \sqrt{2q-1}\right), \quad q > \frac{1}{2}.$$

Trans. Prob.	D_c (Gray)	D_2 (Berger)
$q = 0.8$	1.59×10^{-2}	1.13×10^{-1}
$q = 0.9$	3.10×10^{-3}	5.28×10^{-2}

Table 6.1: Gray and Berger's expressions for the critical distortion under the Hamming distortion measure for binary symmetric Markov sources for two values of the transition probability q.

From Table 6.1, we can see that using the Hamming distortion measure, for the values of q we simulated, and at the BER levels of our interest (10^{-5}), Gray's expression for the critical distortion can indeed guarantee the validity of the closed-form expression for $R(D)$. Therefore, the OPTA values for a symmetric Markov source with transition probability q are identical to that of a non-uniform i.i.d. source with $Pr\{U_k = 0\} = q$.

6.3 Decoder Modifications for Markov Sources

To exploit the memory in the source, the BCJR algorithm employed in the Turbo code decoder has to be modified. However, this is only the case for the first constituent decoder. Because of interleaving, the Markovian property is destroyed in the second constituent encoder's input sequence. Therefore, the second constituent decoder remains identical to the original BCJR algorithm designed for memoryless sources.

6.3.1 Modified BCJR Algorithm in the First Decoder

The modified BCJR algorithm in this section is derived for systematic Turbo codes; however, the modifications can be readily extended to non-systematic Turbo codes.

Given that the L-tuple y_1^L is received at the channel output (as described in

Chapter 2), we have

$$\begin{aligned} Pr\{U_k = i|y_1^L\} &= \sum_s Pr\{U_k = i, S_k = s|y_1^L\} \\ &= \sum_s \frac{Pr\{U_k = i, S_k = s, y_1^k, y_{k+1}^L\}}{Pr\{y_1^k, y_{k+1}^L\}} \\ &= \sum_s \frac{Pr\{U_k = i, S_k = s, y_1^k\} \cdot Pr\{y_{k+1}^L|U_k = i, S_k = s, y_1^k\}}{Pr\{y_1^k\} \cdot Pr\{y_{k+1}^L|y_1^k\}}. \end{aligned}$$

Since given $U_k = i$ and $S_k = s$, the observation y_{k+1}^L does not depend on y_1^k, we have

$$Pr\{y_{k+1}^L|U_k = i, S_k = s, y_1^k\} = Pr\{y_{k+1}^L|U_k = i, S_k = s\}.$$

Now if we define

$$\alpha_k(i,s) \triangleq Pr\{U_k = i, S_k = s|y_1^k\}, \tag{6.3}$$

$$\beta_k(i,s) \triangleq \frac{Pr\{y_{k+1}^L|U_k = i, S_k = s\}}{Pr\{y_{k+1}^L|y_1^k\}}, \tag{6.4}$$

then the conditional probability becomes

$$Pr\{U_k = i|y_1^L\} = \sum_s \alpha_k(i,s)\beta_k(i,s). \tag{6.5}$$

In order to compute $\alpha_k(i,s)$, we consider the previous encoder state S_{k-1} and the corresponding input bit U_{k-1}. Then

$$
\begin{aligned}
\alpha_k(i,s) &= \frac{Pr\{U_k = i, S_k = s, y_1^k\}}{Pr\{y_1^k\}} \\
&= \sum_{i',s'} \frac{Pr\{U_k = i, S_k = s, U_{k-1} = i', S_{k-1} = s', y_1^{k-1}, y_k\}}{Pr\{y_1^{k-1}, y_k\}} \\
&= \sum_{i',s'} Pr\{U_k = i, S_k = s, y_k | U_{k-1} = i', S_{k-1} = s', y_1^{k-1}\} \\
&\quad \cdot \frac{Pr\{U_{k-1} = i', S_{k-1} = s', y_1^{k-1}\}}{Pr\{y_k | y_1^{k-1}\} \cdot Pr\{y_1^{k-1}\}} \\
&= \sum_{i',s'} \frac{\gamma(i,s,y_k|i',s') \cdot \alpha_{k-1}(i',s')}{Pr\{y_k|y_1^{k-1}\}}, \qquad (6.6)
\end{aligned}
$$

where

$$\gamma(i,s,y_k|i',s') \stackrel{\triangle}{=} Pr\{U_k = i, S_k = s, y_k | U_{k-1} = i', S_{k-1} = s'\}.$$

The term y_1^{k-1} is dropped in the conditional probability because once $U_{k-1} = i'$ and $S_{k-1} = s'$ are known, all events at time k do not depend on y_1^{k-1}.

The denominator $Pr\{y_k|y_1^{k-1}\}$ can be computed also by considering the encoder state and input at time k,

$$
\begin{aligned}
Pr\{y_k|y_1^{k-1}\} &= \sum_{i,s} \sum_{i',s'} Pr\{U_k = i, S_k = s, y_k, U_{k-1} = i', S_{k-1} = s' | y_1^{k-1}\} \\
&= \sum_{i,s} \sum_{i',s'} Pr\{U_k = i, S_k = s, y_k | U_{k-1} = i', S_{k-1} = s', y_1^{k-1}\} \\
&\quad \cdot Pr\{U_{k-1} = i', S_{k-1} = s' | y_1^{k-1}\} \\
&= \sum_{i,s} \sum_{i',s'} \gamma(i,s,y_k|i',s') \cdot \alpha_{k-1}(i',s'). \qquad (6.7)
\end{aligned}
$$

Combining (6.6) and (6.7), we obtain a recursive relation for computing $\alpha_k(i,s)$

as follows:

$$\alpha_k(i,s) = \frac{\sum_{i',s'} \gamma(i,s,y_k|i',s')\alpha_{k-1}(i',s')}{\sum_{i,s}\sum_{i',s'} \gamma(i,s,y_k|i',s')\alpha_{k-1}(i',s')}. \tag{6.8}$$

Similarly, for $\beta_k(i,s)$, we consider the encoder state and the corresponding input bit at the next time index $(k+1)$; thus

$$\begin{aligned}
\beta_k(i,s) &= \sum_{i',s'} \frac{Pr\{y_{k+2}^L, y_{k+1}, U_{k+1}=i', S_{k+1}=s'|U_k=i, S_k=s\}}{Pr\{y_{k+2}^L, y_{k+1}|y_1^k\}} \\
&= \sum_{i',s'} \frac{Pr\{y_{k+2}^L|y_{k+1}, U_{k+1}=i', S_{k+1}=s', U_k=i, S_k=s\}}{Pr\{y_{k+2}^L|y_{k+1}, y_1^k\}} \\
&\quad \cdot \frac{Pr\{y_{k+1}, U_{k+1}=i', S_{k+1}=s'|U_k=i, S_k=s\}}{Pr\{y_{k+1}|y_1^k\}} \\
&= \sum_{i',s'} \frac{Pr\{y_{k+2}^L|U_{k+1}=i', S_{k+1}=s'\}}{Pr\{y_{k+2}^L|y_1^{k+1}\}} \cdot \frac{\gamma(i',s',y_{k+1}|i,s)}{Pr\{y_{k+1}|y_1^k\}} \\
&= \sum_{i',s'} \beta_{k+1}(i',s') \cdot \frac{\gamma(i',s',y_{k+1}|i,s)}{Pr\{y_{k+1}|y_1^k\}}.
\end{aligned} \tag{6.9}$$

Using the expression (6.7) for $Pr\{y_{k+1}|y_1^k\}$, we get the following backward recursion for $\beta_k(i,s)$:

$$\beta_k(i,s) = \frac{\sum_{i',s'} \gamma(i',s',y_{k+1}|i,s)\beta_{k+1}(i',s')}{\sum_{i,s}\sum_{i',s'} \gamma(i',s',y_{k+1}|i,s)\alpha_k(i,s)}. \tag{6.10}$$

To solve the forward and backward recursions in (6.8) and (6.10), properly defined boundary conditions are needed. Since both constituent encoders start at the all-zero state, and the Markov source is stationary with marginal distribution p_0 and p_1 as given in (6.1), we have

$$\alpha_0(0,0) = p_0, \quad \alpha_0(1,0) = p_1,$$

$$\alpha_0(i,s) = 0, \quad i = 0,1, \ \forall\, s \neq 0.$$

For the first constituent encoder, the boundary conditions for $\beta_k(i,s)$ are given by

$$\beta_L(0,0) = p_0, \quad \beta_L(1,0) = p_1,$$

$$\beta_L(i,s) = 0, \quad i = 0,1, \ \forall\, s \neq 0.$$

For the second constituent encoder, since the final state is not terminated, its boundary conditions for $\beta_k(i,s)$ are

$$\beta_L(0,s) = \frac{p_0}{2^M}, \quad \beta_L(1,s) = \frac{p_1}{2^M} \quad \forall\, s.$$

Thus, the conditional probability in (6.5) can be computed via $\alpha_k(i,s)$ and $\beta_k(i,s)$ in a forward-backward manner. However, for iterative decoding, the LLR needs to be decomposed. We first examine the components of $\gamma(i,s,y_k|i',s')$ as follows.

$$\begin{aligned}
\gamma(i,s,y_k|i',s') &\triangleq Pr\{U_k = i, S_k = s, y_k | U_{k-1} = i', S_{k-1} = s'\} \\
&= p(y_k^s|U_k = i) \cdot p(y_k^p|U_k = i, S_k = s) \\
&\quad \cdot Pr\{S_k = s|U_k = i, S_{k-1} = s'\} \cdot Pr\{U_k = i|U_{k-1} = i'\} \\
&\triangleq p(y_k^s|i) \cdot p(y_k^p|i,s) \cdot q(s|i,s') \cdot \pi(i|i'). \qquad (6.11)
\end{aligned}$$

Combining (6.11) with (6.8) and (6.5), we can decompose the LLR of U_k into two separate terms:

$$\Lambda_k(U_k) = L_{ch}(U_k) + L_{ex}(U_k), \qquad (6.12)$$

where

$$L_{ch}(U_k) = \log \frac{P(y_k^s|U_k = 1)}{P(y_k^s|U_k = 0)} \qquad (6.13)$$

and

$$L_{ex}(U_k) = \log \frac{\sum_s \sum_{i',s'} p(y_k^p|1,s,s')q(s|1,s')\pi(1|i')\alpha_{k-1}(i',s')\beta_k(1,s)}{\sum_s \sum_{i',s'} p(y_k^p|0,s,s')q(s|0,s')\pi(0|i')\alpha_{k-1}(i',s')\beta_k(0,s)}. \tag{6.14}$$

Comparing with the decomposition of Λ_k in the uniform i.i.d. source case, where

$$\Lambda_k^{iid}(U_k) = L_{ch}(U_k) + L_{ex}^{iid}(U_k) + L_{ap}^{iid}(U_k),$$

we observe that in (6.12), the extrinsic information term $L_{ex}(U_k)$ is in essence the combination of $L_{ex}^{iid}(U_k)$ and $L_{ap}^{iid}(U_k)$, and $L_{ap}^{iid}(U_k)$ is actually the extrinsic information generated from the second constituent decoder. In iterative decoding, a general principle is that the estimation generated by a constituent decoder should not be fed back to itself, otherwise the noise corruption will be highly correlated [15]. On the other hand, the decomposition in (6.12) renders the first constituent decoder unable to "update" its *a priori* information by using the extrinsic information generated from the second constituent decoder in the same way as depicted in Fig. 2.5. Therefore, the extrinsic information term for the first constituent decoder must be further modified.

The method we adopt here is from the first Turbo code paper by Berrou *et al.* [14] in 1993. This is in contrast to the Robertson decoding technique [63] that we used in Chapters 3–5 (after appropriately modifying it to tailor to non-uniform i.i.d. sources and non-systematic Turbo codes). The input to the first decoder now has three components:

$$\mathbf{y}_k = (y_k^s, y_k^{1p}, y_k^{ex(2)}),$$

where $y_k^{ex(2)}$ is the extrinsic information term from the second decoder, $L_{ex}^{(2)}(U_k)$, after de-interleaving. As observed by Berrou *et al.* and Colavolpe *et al.* [20], $y_k^{ex(2)}$ can be approximated as Gaussian with estimated mean M_{ex} and variance σ_{ex}^2. As a result, in (2.1), $\gamma(i, s, y_k|i', s')$ has one more factor described by

$$p(y_k^{ex(2)}|U_k = i) = \frac{1}{\sqrt{2\pi\sigma_{ex}^2}} e^{-\frac{\left[y_k^{ex(2)} - (2i-1)M_{ex}\right]^2}{\sigma_{ex}^2}}, \quad i = 0, 1,$$

in which M_{ex} and σ_{ex}^2 are obtained via an on-line estimation. Thus, in the forward-backward recursion, both $\alpha_k(i, s)$ and $\beta_k(i, s)$ are computed according to this modification.

Finally, in the decomposition of $\Lambda^{(1)}(U_k)$, due to interleaving, $y_k^{ex(2)}$ can be regarded as weakly correlated with y_k^s and y_{1k}^p; therefore, the LLR soft-output generated by the first decoder is:

$$\Lambda_k^{(1)} = L_{ch}(U_k) + L_{ex}^{(1)}(U_k) + \frac{2M_{ex}}{\sigma_{ex}^2} y_k^{ex(2)}.$$

At this point, we can use $L_{ex}^{(1)}(U_k)$ as the extrinsic information to the second constituent decoder.

In [30], Fazel considers designing a Turbo code for the 2400 bps MELP speech coder's output modeled by non-binary Markov chains. However, in [30], both $\alpha_k(s)$ and $\beta_k(s)$ have the same forms as for memoryless sources, and only γ is changed to take account of the source Markovian property. Therefore, the source memory is not fully exploited by such modifications. In [33, 35], Garcia-Frias *et al.* consider the issue of using Turbo codes for Hidden Markov sources. However, it is not clear in [33, 35] how the problem of modifying the extrinsic information in (6.14) is addressed.

6.3.2 Extrinsic Information Modification in the Second Decoder

For the second constituent decoder, due to interleaving, the Markovian property in the input sequence is destroyed; this apparently renders the second constituent decoder unable to adopt the same modifications in the decoding algorithm as in Section 6.3.1. Therefore, the second constituent decoder remains identical to the one used for non-uniform i.i.d. sources. When the source is asymmetric Markov, the second decoder can exploit the source redundancy in the form of non-uniformity. In [19], Cai *et al.* proposed a different modification to exploit the source memory. In their scheme, the original BCJR algorithm is employed in both decoders; however, the extrinsic information is updated by adopting a modified probability according to:

$$P'_{ex}(u_k) = \frac{1}{2}\left[P_{ex}(u_k) + \sum_{u_{k-1}} \pi(u_k|u_{k-1})P_{ex}(u_{k-1})\right].$$

Indeed, the complexity of this method is much lower than ours as well as those methods by Garcia-Frias *et al.* and Fazel. However, one potential computational problem with such modification is that the modified probabilities may not sum up to 1 due to round-up errors. On the other hand, the reliability of the correction term is unknown; therefore, putting equal weight on the original probability and the correction term is questionable. Thus, we propose the following modification to the extrinsic information generated from the second constituent decoder:

$$L'_{ex}(U_k) = c_1 L_{ex}(U_k) + c_2 \log \left[\frac{\sum_i \pi(1|i) Pr\{\hat{U}_{k-1} = i\}}{\sum_i \pi(0|i) Pr\{\hat{U}_{k-1} = i\}} \right],$$

where $c_1, c_2 \in [0, 1]$ and $c_1 + c_2 = 1$. The values of c_1 and c_2 are empirically chosen to yield the best possible improvement. $Pr\{\hat{U}_{k-1} = i\}$ (with i=0, 1) is directly obtained from the extrinsic information described by

$$\begin{aligned} Pr\{\hat{U}_{k-1} = 1\} &= \frac{e^{L_{ex}(U_{k-1})}}{1 + e^{L_{ex}(U_{k-1})}}, \\ Pr\{\hat{U}_{k-1} = 0\} &= \frac{1}{1 + e^{L_{ex}(U_{k-1})}}. \end{aligned}$$

6.4 Encoder Structure Optimization

When the transition probabilities of the Markov source are close to 1 or 0, the input sequence to the encoder would contain long segments of 1s or 0s. As a result, similar to the case of heavily biased non-uniform i.i.d. sources, the encoder structure plays an important role in determining the system performance. Therefore, encoder structure optimization is also necessary in the design of Turbo codes for Markov sources. Furthermore, the two constituent decoders employ different decoding algorithms and thus exploit the source redundancy with different degrees. Therefore, the two constituent encoders do not have to be identical.

Due to the increased number of possible encoder structures by using different constituent encoders, an exhaustive search becomes even more computationally expensive. Our search for the (sub-)optimal encoder structure is performed as follows. Both constituent encoders have $f_0{=}f_4{=}g_0 = 1$ as the initial restriction.

1) Fix the second constituent encoder as, for example, (31, 23), find (by simulation) the best feed-forward and feedback polynomials of the first constituent encoder via the iterative steps described in Section 4.3.

2) Fix the best structure for the first constituent encoder as found in step 1), find the best feed-forward and feedback polynomials of the second constituent encoder.

The initial structure of the second constituent encoder in step 1) is selected according to our results obtained in Chapter 4. For example, for a general asymmetric Markov source, if the marginal distribution $p_0 \geq 0.9$, then the second constituent encoder is initially chosen as (31, 23); if $0.7 \leq p_0 \leq 0.8$, we may choose it as (35, 23). On the other hand, if the source is symmetric, since there is no redundancy in the form of non-uniformity that can be exploited by the second constituent decoder, we may choose the second constituent encoder as Berrou's (37, 21) code, which offers very good performance for uniform i.i.d. sources. Thus we may avoid continuing the iterative procedure between steps 1) and 2), and our selected encoder structure should offer reasonably good performance close to that of the optimal one obtained via more iterative steps between 1) and 2).

6.5 Transformation Between Symmetric Markov Sources and Non-Uniform I.I.D. Sources

When a binary Markov source is symmetric, we may have an alternative approach by using the original BCJR algorithm for this special case. This is because via a simple rate-one convolutional code, we can transform a symmetric Markov source into a non-uniform i.i.d. source [3].

The rate-one convolutional code is described by

$$V_k = \begin{cases} U_1, & \text{if} \quad k = 1, \\ U_k \oplus U_{k-1}, & \text{if} \quad k > 1, \end{cases} \tag{6.15}$$

for $k = 1, 2, \cdots, L$, where $\oplus$ is the modulo-2 summation, and $\{U_k\}$ are generated from a binary symmetric Markov source with transition probability $Pr\{U_k = j | U_{k-1} = i\} = q$, for $i \neq j$. It can be easily verified that after the transformation, $\{V_k\}$ becomes a non-uniform binary i.i.d. sequence with distribution given by $Pr\{V_k = 0\} = q$, and $Pr\{V_k = 1\} = 1 - q$. Therefore, for a binary symmetric Markov source, we can also design a new system as follows.

A rate-one convolutional encoder described by (6.15) is placed between the source and the Turbo encoder to realize the above transformation; at the decoder end, after the Turbo decoder makes a final hard decision $\hat{V}_k$, a convolutional decoder is used, which is described by

$$\hat{U}_k = \begin{cases} \hat{V}_1, & \text{if} \quad k = 1, \\ \hat{V}_k \oplus \hat{V}_{k-1}, & \text{if} \quad k > 1, \end{cases} \tag{6.16}$$

for $k = 1, 2, \cdots, L$. This would perform the inverse transformation, i.e., from a binary non-uniform i.i.d. source into a binary symmetric Markov source.

Via this rate-one convolutional code transformation, we convert the scenario of applying Turbo codes to symmetric Markov sources into that of applying Turbo codes to non-uniform i.i.d. sources, which was studied in Chapters 4 and 5. This transformation is only valid between binary symmetric Markov sources and binary non-uniform i.i.d. sources. For more general *asymmetric* Markov sources, the existence of such transformation is still unknown.

However, one practical problem with employing this transformation is that very few decoding errors in $\{\hat{V}_k\}$ may become disastrous in $\{\hat{U}_k\}$ due to the error-propagation via the rate-1 convolutional code. To limit this effect, it was suggested in [3] that the long source information sequence $\{U_k\}_{k=1}^{L}$ should be grouped into smaller blocks, which is much shorter than the whole sequence length L. By doing this, the error propagation is restricted within small blocks. This method, on the other hand, contradicts with the use of Turbo codes, since the error-correcting performance of Turbo codes degrades as the block length decreases. Therefore, this method might not give a satisfactory performance.

6.6 Numerical Results and Discussion

In this section, we present simulation results of our designed systematic Turbo codes for the transmission of binary symmetric Markov sources over BPSK-modulated AWGN and Rayleigh fading channels with known channel state information. All simulated Turbo codes have 16-state constituent encoders and use the same pseudo-random interleaver introduced in [15]. Due to the increased complexity and the limitations on our computing facilities, the sequence length is chosen to be $L = 512 \times 512 = 262144$ and at least 200 blocks are simulated; this would guarantee a reliable BER estimation at the 10^{-5} level with 524 errors. The number of iterations in the Turbo code decoder is also 20. Simulations are performed for rate $R_c = 1/3$ with q=0.8 and 0.9.

The best encoder we found for q=0.9 has the first constituent encoder as (31, 23), and the second as (35, 23). For q=0.8, (35, 23) for both constituent encoders turns out to be the best choice.

In the modification to the extrinsic information generated from the second constituent decoder, the weight between the original extrinsic information term and the correction term is determined by simulations. Given $c_1 + c_2 = 1$, we simulated with $c_1 = 0, 0.1, 0.2, \cdots, 0.9, 1.0$. And the best choice is to set c_1=0.8 and c_2=0.2 for both q=0.8 and 0.9, although the gain due to this change is no more than 0.1 dB.

Fig. 6.1 shows the performances of our rate-1/3 systematic Turbo codes designed for transmitting binary symmetric Markov sources over AWGN channels in comparison with the performance of Berrou's (37, 21) code which does not exploit the source memory. At the 10^{-5} BER level, when q=0.8, our system offers a gain of 1.29 dB over Berrou's code that assumes the source is uniform i.i.d.; when $q = 0.9$, the above gain becomes 3.03 dB. Furthermore, we observe that for $q = 0.8$ and 0.9, the OPTA gaps of our designed system are 0.94 dB and 1.36 dB.

Fig. 6.2 illustrates similar results over BPSK-modulated Rayleigh fading channels with known channel state information. When $q = 0.8$, the gain due to exploiting the source memory in our design is 1.55 dB, which brings the performance at a distance of 1.08 dB away from OPTA. When $q = 0.9$, the gain offered by our system over Berrou's code for uniform memoryless sources increases up to 3.57 dB; in this case, the OPTA gap is 1.45 dB. The OPTA values and gaps are summarized in Table 6.2.

Our system is clearly not optimal as it suffers from the following deficiencies.

First, due to interleaving, the second decoder is unable to exploit the source redundancy in the form of memory. By adopting the modified extrinsic information from the second constituent decoder via a weighted correction term, only marginal improvement has been observed. Therefore, the second constituent decoder has an inferior error-correcting capability, rendering the system sub-optimal.

Second, for symmetric Markov sources, although the systematic sequence has a uniform marginal distribution, its high-order distributions are non-uniform. According to Shamai and Verdú's work [67], there is still a performance loss due to the non-uniformity in high-order distributions. Non-systematic Turbo codes may still seem a more suitable solution to further reduce the OPTA gaps; however, our initial simulations did not show satisfactory performance results.

Third, our method relies on the Gaussian assumption of the extrinsic information generated by the Turbo code decoder. This imposes limitations on our designed algorithm to be applied accurately for wireless fading channels. Although when the channel state information is known to the decoder, Rayleigh fading channels behave like AWGN channels, the mean of the Gaussian distribution is different for each transmitted bit. However, the estimated mean M_{ex} is only an average over the whole transmitted sequence. Therefore, more performance loss would occur in the Rayleigh fading channel case due to this mismatch.

With the above considerations, designing a Turbo coding system for Markov sources which can perform closer to OPTA remains an interesting and challenging problem.

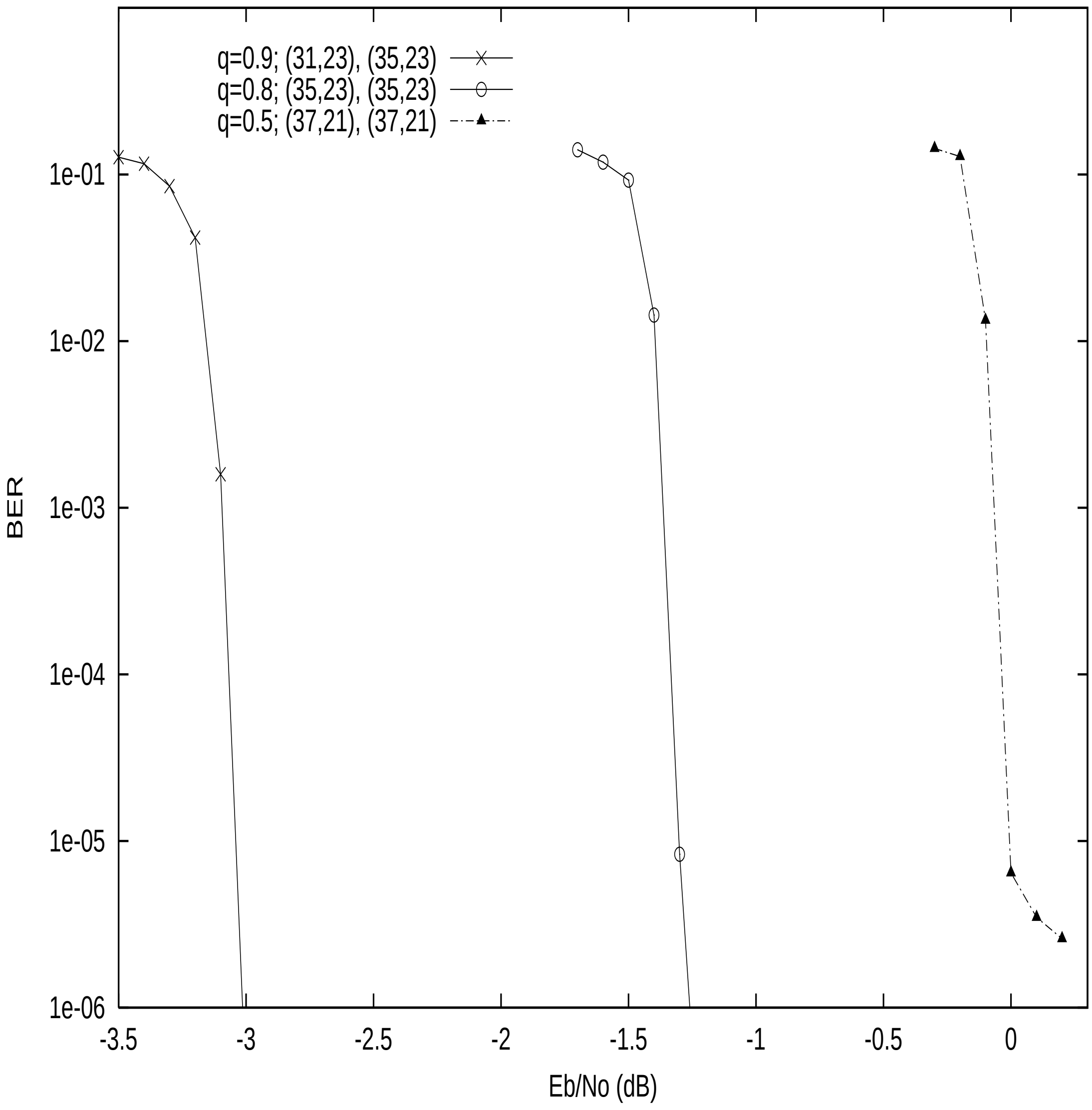

Figure 6.1: Turbo codes for binary symmetric Markov sources, R_c=1/3, L=262144, AWGN channel.

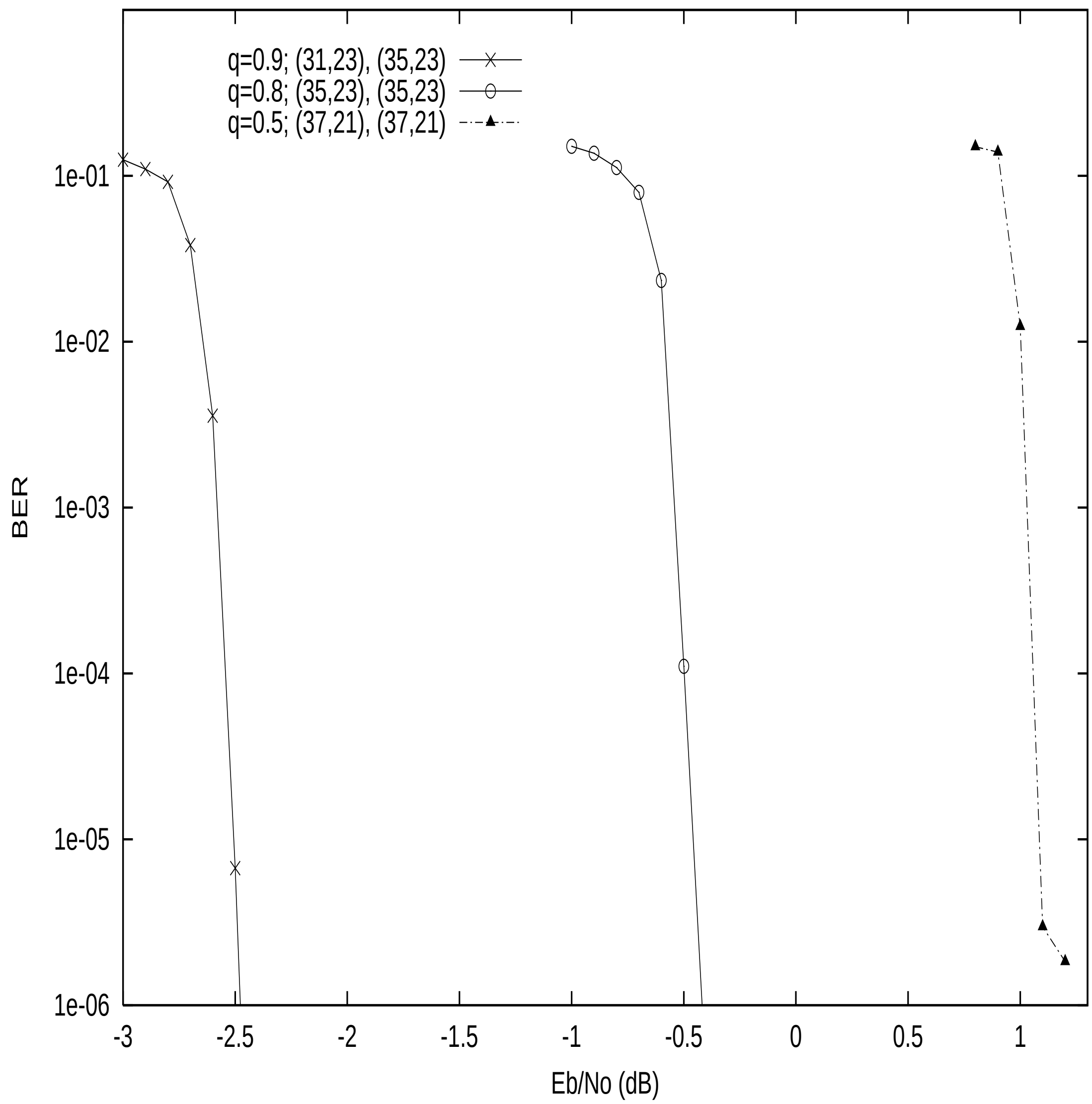

Figure 6.2: Turbo codes for binary symmetric Markov sources, R_c=1/3, L=262144, Rayleigh fading channel with known channel state information.

Transition Probability	AWGN		Rayleigh	
	OPTA value	OPTA gap	OPTA value	OPTA gap
$p_0 = 0.8$	-2.24	0.94	-1.56	1.08
$p_0 = 0.9$	-4.40	1.36	-3.96	1.45

Table 6.2: OPTA values and gaps in E_b/N_0 at BER=10^{-5} level (in dB), Turbo codes for binary symmetric Markov source, R_c=1/3, L=262144, AWGN and Rayleigh fading channel.

6.7 Comparison with Tandem Schemes

In this section, we present the performance comparison between our Turbo codes designed for binary Markov sources with two tandem schemes. The Markov source used here is symmetric with transition probability $q_{00} = q_{11} = q$. Similar to Section 5.6, the tandem schemes consist of a Huffman code followed by a standard symmetric Turbo code with constituent encoders (37, 21) or (35, 23), where the (35, 23) Turbo code offers a lower error-floor performance than that of its (37, 21) peer at the price of a slight performance loss in the water-fall region.

Again the comparison should be made at the same overall rate. Our joint source channel coding system has $R_c = 1/2$ and $R_s = 1$; the tandem scheme, however, has $R_c = 1/3$, therefore it needs to have $R_s = 2/3$. This can be achieved via numerically choosing the value of q such that the Huffman code produces an output sequence with average codeword length close to 2/3. Due to the source memory, a Markov source is harder to compress than a non-uniform i.i.d. source. In fact, using a 4^{th}-order Huffman code, to guarantee $R_s = 2/3$, we need to have q=0.881685. However, at this value of q, the entropy of the symmetric Markov source is 0.524. To achieve higher efficiency in compression, we adopt a 8^{th}-order Huffman code, which can achieve R_s=0.666667 at q=0.848315, while the entropy of the Markov source is 0.614.

As in Section 5.6, the sequence length is L=12000, and at least 60000 blocks are simulated to produce a reliable average performance in the error-floor region. An

S-random interleaver with S=10 is adopted replacing the pseudo-random interleaver. The number of iterations in the Turbo decoder is also 20. The detailed implementation procedure is similar to that described in Section 5.6.

Figure 6.3 shows the performance comparison of our system with the two tandem schemes over AWGN channels. We observe that although initially the tandem schemes offer better water-fall performance, they soon lose the battle to our system due to their high BER performance from medium to high SNRs. In the two tandem schemes, the error-floor performance is significantly lowered (from the 10^{-3} down to the 10^{-4} BER level) by using the (35, 23) code as constituent encoders. However, unless the channel coding part is absolutely error-free, this high BER performance is inevitable in tandem schemes due to the disastrous error-propagation problem in variable-length source code decoders. On the other hand, although fixed-length source codes can eliminate this error-propagation problem, they can hardly achieve near-optimal data compression. Therefore, a tandem scheme consisting of a fixed-length source code and a regular Turbo code is not optimal and therefore may not yield a satisfactory overall performance. Our joint source-channel coding design, however, enjoys a lower complexity since no source encoding and decoding are performed, and offers a superior BER performance which is robust to channel errors.

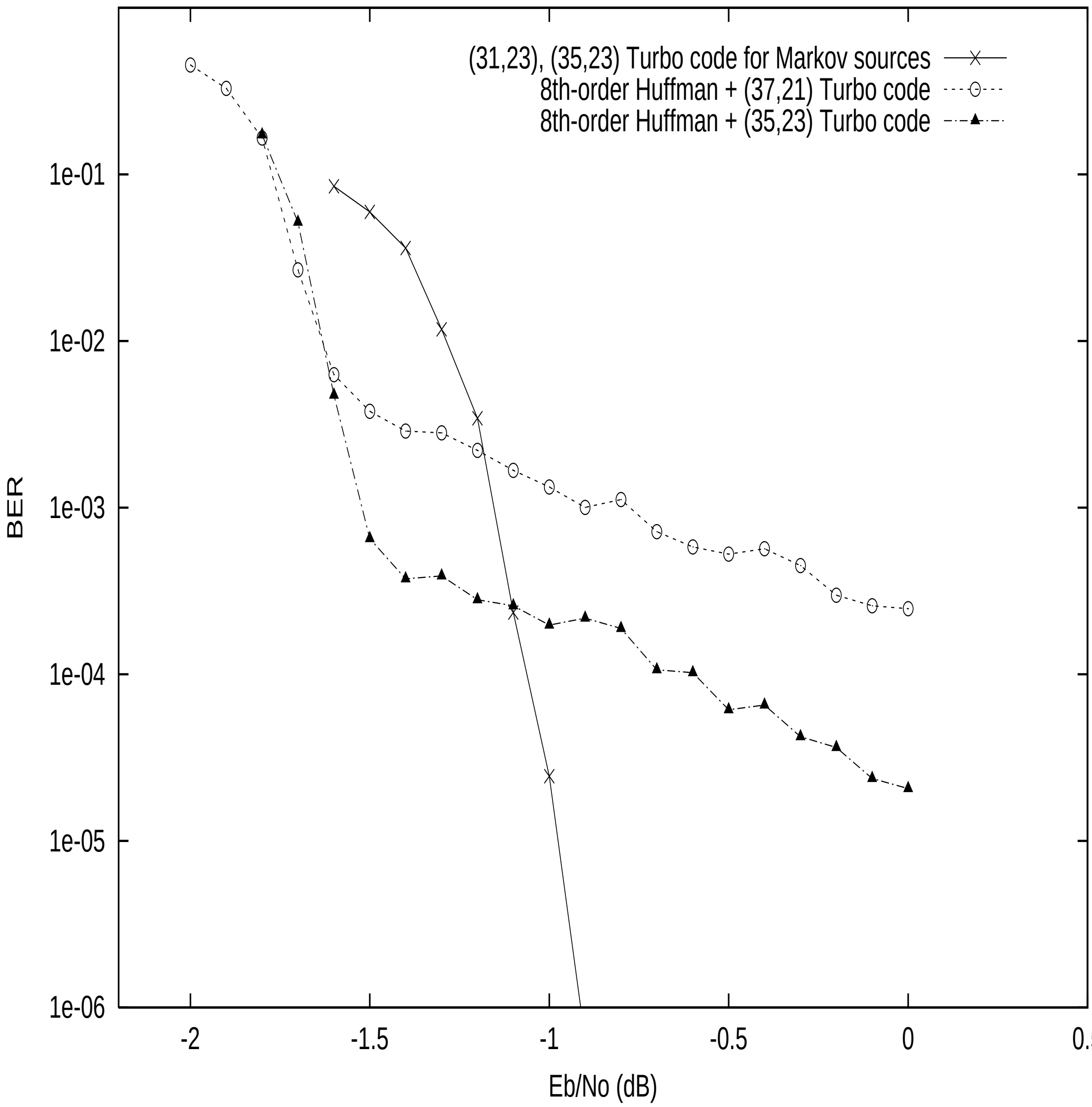

Figure 6.3: Performance comparison of our system ($R_s = 1, R_c = 1/2$) with that of tandem schemes ($R_s = 2/3$, $R_c = 1/3$), binary symmetric Markov sources, q=0.848315, L=12000, AWGN channel.

Chapter 7

Conclusion and Future Work

7.1 Summary

This thesis addresses three joint source-channel coding issues via Turbo codes.

The first topic is to design a channel optimized vector quantization (COVQ) system for Turbo-coded AWGN and Rayleigh fading channels with known channel state information. To incorporate Turbo codes in the COVQ design, we propose modeling the concatenation of Turbo code encoder, BPSK-modulator, AWGN or Rayleigh fading channel, Turbo code decoder, as well as a q-level soft-decision scalar quantizer, as an expanded discrete channel. Two models are proposed via different approaches. The first one is the discrete memoryless channel (DMC) model for Turbo-coded AWGN channels, which is obtained based on the Gaussian approximation to the log-likelihood ratio (LLR) soft output generated from the Turbo code decoder. The DMC model enjoys a closed form expression for its channel transition matrix. The second model is obtained via direct estimation of the transition matrix for Turbo-coded AWGN and Rayleigh fading channels. It is a more accurate method since it captures the memory within blocks; hence it is called the discrete block-memoryless channel (DBMC) model. A COVQ system is then designed for these Turbo-coded channel models. For a wide range of channel signal-to-noise ratio (CSNR) values, our Turbo-COVQ system offers substantially better performance than that of the original COVQ as well as the tandem scheme. Furthermore, in comparison with a recent Turbo decoding COVQ system proposed by Ho, in which the entire soft-decision LLR from the Turbo decoder is utilized, our DMC-based system offers comparable performance, while the DBMC-based design yields better performance. Our COVQ decoding can be implemented by a table-lookup and thus enjoys lower decoding complexity than Ho's scheme, although the memory requirement might be higher.

The second part of this thesis is devoted to designing Turbo codes for the transmission of non-uniform i.i.d. sources. When systematic Turbo codes are used, via an easy modification to the extrinsic information, the source redundancy in the form of non-uniformity is exploited in the decoder. Furthermore, for a given source distribution, the encoder structure is optimized. Despite the significant gains achieved, the OPTA gaps remain relatively wide. An analysis reveals that the drawback lies in the *systematic* structure. Under certain conditions and when the length of the input source sequence tends to infinity, we proved that the encoder state distribution and the marginal distribution of the constituent recursive convolutional encoders output become asymptotically uniform regardless of the degree of non-uniformity in the source. These conditions are both necessary and sufficient. Based on empirical observations, we also conjecture that under certain conditions, recursive convolutional encoders can produce asymptotically uniform higher-order distribution irrespective of the source distribution. Consequently, these conditions serve as design criteria for the choice of good encoder structures. As a result, the outputs of our selected non-systematic Turbo codes are suitably matched to the channel input since a uniformly distributed input maximizes the channel mutual information and hence may achieve capacity. Simulation results show substantial gains by the non-systematic codes over previously designed systematic Turbo codes; furthermore, their performance is within 0.74 to 1.17 dB from the Shannon limit. Finally, we compare our joint source-channel coding system with two tandem schemes which employ a 4^{th}-order Huffman code (which performs near optimal data compression) and a Turbo code that either gives excellent waterfall BER performance or a low error-floor performance. At the same overall transmission rate, our system offers robust and superior performance at low BER levels (below 10^{-4}), while its complexity is lower.

The third part of the thesis investigates the design of Turbo codes for stationary ergodic Markov sources. In the first constituent decoder, the decoding algorithm is modified to exploit the source redundancy in the form of memory. Due to interleaving, the second constituent decoder is unable to adopt the same modifications. However, the extrinsic information from the second decoder is modified via a weighted correction term. When the source is asymmetric, the source redundancy is in the form of both memory and non-uniformity. Although the second constituent decoder cannot utilize the source memory, it can take advantage of the source non-uniform distribution as shown in the second part of the thesis. When the source is a stationary symmetric Markov source, it can be transformed into a non-uniform i.i.d. source via a rate-1 convolutional encoder. Practical implementation issues for this approach are discussed, while the limitations on our design are examined. In our simulations, substantial gains (from 1.29 up to 3.57 dB) over the original Turbo codes' performance are demonstrated. The OPTA gaps of our achieved performance are in the range of 0.94–1.45 dB. Finally, performance comparisons are also made between our joint source-channel coding system with two tandem schemes where a 8^{th}-order Huffman code is adopted to achieve near-optimal data compression, and two regular rate-1/3 Turbo codes are employed. We observe a similar phenomenon as in the scenario of non-uniform i.i.d. sources — our system provides robust and substantially better performance than that of the tandem schemes with a lower system complexity.

7.2 Future Work

The following directions are of great interest for future research.

- In Chapter 5, we gave a conjecture on the conditions when a recursive convolutional encoder's output has high-order asymptotically uniform distributions regardless of the degree of non-uniformity in the source distribution. We were able to prove it only for some special cases. However, for a general pair of $F(D)$ and $G(D)$, the proof does not seem obvious. This is an interesting problem on its own and deserves further investigation.

- So far, only limited attention has been paid to designing Turbo codes for channels with memory, which are known to have higher capacity (if they are information stable [1]) than their equivalent memoryless channels obtained via ideal channel interleaving. The only previous work on Turbo codes for channels with memory is by Garcia-Frias *et al.*, who consider the Gilbert-Elliot channel model [34, 36]. In [1], Alajaji and Fuja proposed the binary finite-memory *Polya contagion channel* model, which has fewer parameters than that of the Gilbert-Elliot channel, and enjoys a closed-form expression for its block channel probability and capacity. Therefore, we propose designing Turbo codes for the Polya contagion channel model, and investigating the performance in terms of the corresponding Shannon limit. The source to be transmitted over the contagion channel, can first be assumed to be uniform i.i.d.. Then, extensions for non-uniform i.i.d. and Markov sources can be considered. This study can also serve as a good model for the design of powerful coding systems for wireless channels with correlated fading.

- In this thesis, only BPSK modulation is considered. In [73], Takahara *et al.* considered constellation mappings for two-dimensional modulation signaling of non-uniform sources. We propose to investigate, for the case of non-uniform i.i.d. sources, the design of systematic Turbo codes in conjunction with a two-dimensional constellation mapping strategy for the systematic sequence. This promises of establishing high-performing high-rate bandwidth efficient joint source-channel trellis coded modulation (TCM) schemes based on Turbo codes.

Bibliography

[1] F. Alajaji and T. E. Fuja, “A communication channel modeled on contagion,” *IEEE Trans. on Inform. Theory*, vol. 40, pp. 2035–2041, Nov. 1994.

[2] F. Alajaji and N. Phamdo, “Soft-decision COVQ for Rayleigh-fading channels,” *IEEE Commun. Lett.*, vol. 2, pp. 162–164, Jun. 1998.

[3] F. Alajaji, N. Phamdo, N. Farvardin and T. Fuja, “Detection of binary Markov sources over channels with additive Markov noise,” *IEEE Trans. Inform. Theory,* vol. 42, pp. 230–239, Jan. 1996.

[4] F. Alajaji, S. A. Al-Semari and P. Burlina, “Visual communication via trellis coding and transmission energy allocation,” *IEEE Trans. Commun.*, vol. 47, pp. 1722–1728, Nov. 1999.

[5] F. Alajaji, N. Phamdo and T. Fuja, “Channel codes that exploits the residual redundancy in CELP-encoded speech,” *IEEE Trans. Speech and Audio Processing*, vol. 4, pp. 325–336, Sept. 1996.

[6] S. A. Al-Semari, F. Alajaji and T. Fuja, "Sequence MAP decoding of trellis codes for Gaussian and Rayleigh channels," *IEEE Trans. Veh. Technol.*, vol. 48, pp. 1130–1140, Jul. 1999.

[7] T. Berger, *Rate Distortion Theory: A Mathematical Basis for Data Compression*, Prentice Hall, 1971.

[8] T. Berger, "Explicit bounds to $R(D)$ for a binary symmetric Markov source," *IEEE Trans. Inform. Theory*, vol. 23, pp. 52–59, Jan. 1977.

[9] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. Inform. Theory,* vol. IT-20, pp. 248–287, Mar. 1974.

[10] J. Bakus and A. K. Khandani, "Combined source-channel coding using Turbo-codes," *Electron. lett.*, vol. 33, pp. 1613–1614, $11th$ Sept. 1997.

[11] J. Bakus and A. K. Khandani, "Quantizer design for Turbo-code channels," Technical Report E&CE#99-04, University of Waterloo, Jul. 1999.

[12] A. S. Barbulescu and S. S. Pietrobon, "Terminating the trellis of turbo-codes in the same state," *Electr. Lett.*, vol. 31, pp. 22–23, 5th Jan. 1995.

[13] S. Benedetto, and G. Montorsi, "Unveiling turbo codes: some results on parallel concatenated coding schemes," *IEEE Trans. Inform. Theory*, vol. 42, pp. 409–428, Mar. 1996.

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

[15] C. Berrou and A. Glavieux, "Near optimum error correcting coding and decoding: Turbo-codes," *IEEE Trans. Commun.*, vol. 44, pp. 1261–1271, Oct. 1996.

[16] C. Berrou, and A. Glavieux, "Reflections on the prize paper: 'Near optimum error-correcting coding and decoding: turbo codes'," *IEEE Inform. Theory Soc. Newsletter,* vol.48, pp. 23–31, Jun. 1998.

[17] R. E. Blahut, *Principles and Practice of Information Theory*, Addison Wesley, Massachusetts, 1988.

[18] P. Burlina, F. Alajaji and R. Chellappa, "Transmission of two-tone images over noisy communication channels with memory," Technical Report, CAR-TR-814, Center for Automation Research, Univ. of Maryland, College Park, 1996.

[19] Z. Cai, K. R. Subramanian and L. Zhang, "Source-controlled channel decoding using nonbinary turbo codes," *Elect. Lett.*, vol. 37, pp. 39–40, $4th$ Jan. 2001.

[20] G. Colavolpe, G. Ferrari, and R. Raheli, "Extrinsic information in Turbo decoding: a unified view," *Proc. of Globecom'99*, pp. 505–509, 1999.

[21] O. M. Collins, O. Y. Takeshita and D. J. Costello, *Jr.*, "Iterative decoding of non-systematic Turbo codes," *Proc. of ISIT 2000*, Sorrento, Italy, p. 172, Jun. 2000.

[22] D. J. Costello, *Jr.*, P. C. Massey, O. M. Collins and O. Y. Takeshita, “Some reflections on the mythology of turbo-codes,” in *Proc. 3rd ITG Conf. on Source and Channel Coding*, pp. 157–160, Munich, Germany, Jan. 2000.

[23] D. J. Costello, *Jr.*, H. A. Cabral and O. Y. Takeshita, “Some thoughts on the equivalence of systematic and nonsystematic convolutional encoders,” *Codes, Graphs and Systems,* (edited by R. E. Blahut and R. Koetter), Kluwer Academic, 2002.

[24] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, Wiley series in Telecommunications, 1991.

[25] D. Divsalar and F. Pollara, “Multiple Turbo codes for deep-space communications,” TDA Progress Report 42-121, Jet Propulsion Laboratory, Pasadena, CA, pp. 66–77, May 15, 1995.

[26] T. M. Duman and M. Salehi, “New performance bounds for turbo codes,” *IEEE Trans. Commun.,* vol. 46, pp. 717–723. Jun. 1998.

[27] I. J. Fair, V. K. Bhargava and Q. Wang, ”On the power spectral density of self-synchronizing scrambled sequences,” *IEEE Trans. Inform. Theory*, vol. 44, pp. 1687-1693, Jul. 1998.

[28] N. Farvardin, “A study of vector quantization for noisy channels,” *IEEE Trans. Inform. Theory*, vol. 36, pp. 799–809, Jul. 1990.

[29] N. Farvardin and V. Vaishampayan, "On the performance and complexity of channel-optimized vector quantizers," *IEEE Trans. Inform. Theory*, pp. 155–160, Jan. 1991.

[30] T. Fazel, *Design of Joint Source-Channel Decoding Algorithms for the 2400 bps MELP Speech Coder*, Ph.D. dissertation, University of Maryland, 1999.

[31] T. Fingsheidt, T. Hindelang, R. V. Cox and N. Sheshadri, "Joint source-channel (de)coding for mobile communications," *IEEE Trans. Commun.*, vol. 50, pp. 200–212, Feb. 2002.

[32] R. G. Gallager, *Information Theory and Reliable Communication.* New York: Wiley, 1968.

[33] J. Garcia-Frias, and J. D. Villasenor, "Combining hidden Markov source models and parallel concatenated codes," *IEEE Commun. lett.*, vol. 1, pp. 111–113, Jul. 1997.

[34] J. Garcia-Frias, and J. D. Villasenor, "Turbo decoders for Markov channels," *IEEE Commun. Lett.*, vol. 2, pp. 257–259, Sept. 1998.

[35] J. Garcia-Frias and J. D. Villasenor, "Joint turbo decoding and estimation of hidden Markov sources," *IEEE J. Selec. Areas in Commun.*, vol. 19, pp. 1671–1679, Sept. 2001.

[36] J. Garcia-Frias, and J. D. Villasenor, "Turbo decoding of Gilbert-Elliot channels," *IEEE Trans. Commun.*, vol. 50, pp. 357–363, Mar. 2002.

[37] A. Gersho and R. M. Gray, *Vector Quantization and Singal Compression*, Norwell, MA: Kluwer, 1992.

[38] R. M. Gray, "Information rates for autoregressive processes," *IEEE Trans. Inform. Theory*, vol. 16, pp. 412-421, Jul. 1970.

[39] E. Ayanoglu and R. M. Gray, "The design of joint source and channel trellis waveform coders," *IEEE Trans. on Inform. Theory*, vol. 33, pp. 855–865, Nov. 1987.

[40] L. Guivarch, J.-C. Carlach and P. Siohan, "Joint source-channel soft decoding of Huffman codes with Turbo-codes," in *Proc. DCC 2000*, pp. 83–92, Snowbird, Utah, USA, Mar. 2000.

[41] J. Hagenauer, "Source controlled channel decoding," *IEEE Trans. Commun.*, vol. 43, pp. 2449–2457, Sept. 1995.

[42] J. Hagenauer and R. Bauer, "The Turbo principle in joint source channel decoding of variable length codes," in *Proc. IEEE Inform. Theory Workshop*, pp. 128–130, Cairns, Australia, Sept. 2001.

[43] J. Hagenauer, P. Hoeher, "A Viterbi algorithm with soft-decision outputs and its applications," *Proc. GLOBECOM'89*, pp. 1680–1686, Nov. 1989.

[44] J. Hagenauer, E. Offer and L. Papke, "Iterative decoding of binary block and convolutional codes," *IEEE Trans. Inform. Theory,* vol. 42, pp. 429–445, Mar. 1996.

[45] J. Hagenauer, P. Robertson, and L. Papke, "Iterative (turbo) decoding of systematic convolutional codes with MAP and SOVA algorithms," *Proc. ITG Conf. on "Source and Channel Coding,"* Frankfurt, Germany, pp. 1-9, Oct. 1994.

[46] E. K. Hall and S. G. Wilson, "Design and analysis of Turbo codes on Rayleigh fading channels," *IEEE Journal on Selected Areas in Commun.*, vol. 16, pp. 160–174, Feb. 1998.

[47] C. Heegard and S. B. Wicker, *Turbo Coding,* Kluwer Academic Publishers, 1999.

[48] K.-P. Ho, "Soft-decoding vector quantizer using reliability information from Turbo-codes," *IEEE Commun. lett.*, vol. 3, pp. 208–210, Jul. 1999.

[49] J. Kroll and N. Phamdo, "Source-channel optimized trellis codes for bitonal image transmission over AWGN channels," *IEEE Trans. Image Processing,* vol. 8, pp. 899–912, Jul. 1999.

[50] H. Kumazawa, M. Kasahara, and T. Namekawa, "A construction of vector quantizers for noisy channels," *Electronics and Engineering in Japan*, vol. 67-B, pp. 39-47, Jan. 1984.

[51] A. Kurtenbach and P. Wintz, "Quantizing for noisy channels," *IEEE Trans. Commun. Technology*, vol. 17, pp. 291–302, Apr. 1969.

[52] Y. Linde, A. Buzo, and R. M. Gray, "An algorithm for vector quantizer design," *IEEE Trans. Commun.*, vol. COM-28, pp. 84–95, 1980.

[53] F.-H. Liu, P. Ho and V. Cuperman, "Sequential reconstruction of vector quantized signals transmitted over Rayleigh fading channels," *Proc. of IEEE Int. Conf. on Commun.*, pp. 23-27, New Orleans, 1994.

[54] S. P. Lloyd, "Least squares quantization in PCM," *IEEE Trans. Inform. Theory*, vol. 28, pp. 129–137, Mar. 1982.

[55] J. Makhoul, S. Roucos, and H. Gish, "Vector quantization in speech coding," *Proc. IEEE*, vol. 23, pp. 1551–1588, Nov. 1985.

[56] P. C. Massey, O. Y. Takeshita, and D. J. Costello, *Jr.*, "Contradicting a myth: Good turbo-codes with large memory order," *Proc. IEEE Int. Symp. on Inform. Theory 2000*, p. 122, Sorrento, Italy, Jun. 2000.

[57] P. C. Massey and D. J. Costello, *Jr.*, "Turbo codes with recursive non-systematic quick-look-in constituent codes," *Proc. ISIT 2001*, Washington, DC, Jun. 24–29, 2001.

[58] J. Max, "Quantizing for minimum distortion," *IRE Trans. Inform. Theory*, vol. 6, pp. 7–12, Mar. 1960.

[59] R. J. McEliece, *The Theory of Information and Coding,* Addison-Wesley, 1977.

[60] R. J. McEliece, “Are Turbo-like codes effective on nonstandard channels?” *IEEE Inform. Theory Soc. Newsletter*, vol. 51, pp. 1–8, Dec. 2001.

[61] L. C. Perez, J. Seghers, and D. J. Costello, Jr., “A distance spectrum interpretation of turbo codes,” *IEEE Trans. on Inform. Theory,* vol. 42, pp. 1698–1709, Nov. 1996.

[62] N. Phamdo and F. Alajaji, “Soft-decision demodulation design for COVQ over white, colored and ISI Gaussian channels,” *IEEE Trans. Commun.*, vol. 48, pp. 1499–1506, Sept. 2000.

[63] P. Robertson, “Illuminating the structure of parallel concatenated recursive systematic (turbo) codes,” *Proc. GLOBECOM'94* San Francisco, CA, vol. 3, pp. 1298-1303, Nov. 1994.

[64] P. Robertson, E. Villebrun, and P. Hoeher, “A comparison of optimal and sub-optimal MAP decoding algorithms operating in the log domain,” *Proc. of Int. Conf. on Commun.*, pp. 1009–1013, Seattle, WA, Jun. 18–22, 1995.

[65] K. Sayood and J. C. Borkenhagen, “Use of residual redundancy in the design of joint source-channel codes,” *IEEE Trans. Commun.*, vol. 39, pp. 838–846, Jun. 1991.

[66] P. L. Secker and P. O. Ogunbona, "Methods of channel-optimized trellis source coding for the AWGN channel," *Proc. IEEE ISIT*, p. 236, Trondheim, Norway, Jun. 1994.

[67] S. Shamai and S. Verdú, "The empirical distribution of good codes," *IEEE Trans. Inform. Theory*, vol. 43, pp. 836–846, May 1997.

[68] S. Shamai and I. Sason, "Variations on Gallager's bounding techniques: performance bounds for Turbo codes in Gaussian and fading channels," *Proc. 2nd Int. Symp. on Turbo Codes and Related Topics*, Brest, France, pp. 27–34, Sept. 2000.

[69] C. E. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J.*, vol. 27, pp. 379–423 and pp. 623-656, Jul. and Oct. 1948.

[70] C. E. Shannon, "Coding theorems for a discrete source with a fidelity criterion," *IRE Nat. Conv. Rec.*, pt. 4, pp. 142–163, Mar. 1959.

[71] R. Y. Shao, S. Lin, and M. P. Fossorier, "Two simple stopping criteria for turbo decoding," *IEEE Trans. Commun.*, vol. 47, pp. 1117-1120, Aug. 1999.

[72] M. Skoglund and P. Hedelin, "Hadamard-based soft-decoding for vector quantization over noisy channels," *IEEE Trans. on Inform. Theory*, vol. 45, pp. 515–532, Mar. 1999.

[73] G. Takahara, F. Alajaji, N. C. Beaulieu and H. Kuai, "Constellation mappings for two-dimensional signaling of non-uniform sources," *IEEE Trans. Commun.*, to appear.

[74] V. Vaishampayan and N. Farvardin, "Joint design of block source codes and modulation signal sets," *IEEE Trans. on Inform. Theory*, vol. 36, pp. 1230–1248, Jul. 1992.

[75] R. E. Van Dyck and D. Miller, "Transport of wireless video using separate, concatenated, and joint source-channel coding", *Proc. of the IEEE*, vol. 87, pp. 1734-1750, Oct. 1999.

[76] A. J. Viterbi, and J. K. Omura, *Principles of Digital Communication and Coding*, McGraw-Hill, 1979.

[77] W. Xu, J. Hagenauer and J. Hollmann, "Joint source-channel decoding using the residual redundancy in compressed images," *Proc. Int. Conf. Commun.*, pp. 142–148, Dallas, TX, Jun. 1996.

Vita

Guang-Chong Zhu

EDUCATION:

- Ph.D. in Mathematics and Engineering, January 2003, Queen's University, Kingston, Ontario, Canada.
  - Thesis title: *Joint Source-Channel Coding via Turbo Codes.*
  - Supervisor: Dr. Fady Alajaji.
- M.Sc. in Applied Mathematics (Communications), September 1998, Queen's University, Kingston, Ontario, Canada.
  - Project title: *Performance Evaluation of Turbo Codes.*
  - Supervisor: Dr. Fady Alajaji.
- M.Sc. in Computational Mathematics (Numerical Analysis), June 1997, Shandong University, Jinan, Shandong, China.
  - Thesis title: *Discrete Numerical Solutions for Two Types of Time-Developing Equations.*
  - Supervisor: Professor Yirang Yuan.
- B.Sc. in Computational Mathematics, July 1994, Shandong University, Jinan, Shandong, China.

AWARDS AND HONORS:

- OGSST (Ontario Graduate Scholarship in Science and Technology), 2000–2002.
- Queen's Graduate Fellowship, 1999–2000.
- Queen's Graduate Awards, 1998–1999.
- R. Samuel McLaughlin Fellowship, 1997–1998.
- Shandong University's 1997 Best Thesis (M.Sc.) Award.
- Shandong University's Excellent Graduate Student Scholarship, 1995–1996.

PUBLICATIONS:

- G.-C. Zhu and F. Alajaji, "Turbo codes for binary Markov sources," submitted to the *2003 Canadian Workshop on Information Theory*, January 2003.
- G.-C. Zhu, F. Alajaji, J. Bajcsy and P. Mitran, "Transmission of Non-Uniform Memoryless Sources via Non-Systematic Turbo Codes," submitted to *IEEE Transactions on Communications*, November 2002.
- G.-C. Zhu, F. Alajaji, J. Bajcsy and P. Mitran, "Transmission of Non-Uniform Memoryless Sources over Wireless Fading Channels via Non-Systematic Turbo Codes," *Proceedings of IASTED Wireless Optical Communication Conference (WOC'2002)*, pp. 119–124, Banff, Alberta, Canada, July 17–19, 2002.
- G.-C. Zhu, F. Alajaji, J. Bajcsy and P. Mitran, "Non-systematic Turbo codes for non-uniform i.i.d. sources over AWGN channels," *Proceedings of the 2002 Conference on Information Sciences and Systems (CISS)*, pp. 770–774, Princeton, NJ, March 20–22, 2002.
- G.-C. Zhu and F. Alajaji, "Turbo Codes for Non-uniform Memoryless Sources over Noisy Channels," *IEEE Communications Letters*, vol. 6, pp. 64–66, February 2002.

- G.-C. Zhu and F. Alajaji, "Design of Turbo Codes for Non-Equiprobable Memoryless Sources," *Proceedings of the 39th Annual Allerton Conference on Communication, Control and Computing*, pp. 1253–1262, Monticello, IL, October 3–5, 2001.

- G.-C. Zhu and F. Alajaji, "Soft-Decision COVQ for Turbo-Coded AWGN and Rayleigh fading channels," *IEEE Communications Letters*, vol. 5, pp. 257-259, June 2001.

- G.-C. Zhu and F. Alajaji, "Soft-Decision COVQ Based on Turbo Codes," *Proceedings of the 2nd International Symposium on Turbo Codes and Related Topics*, pp. 451–454, Brest, France, September 4–7, 2000.

- G.-C. Zhu, "The Lumped Method for Parabolic Integro-Differential Equations," *Proceedings of Computational Physics*, p. 734, Beijing, China, August 26–30, 1997.

- G.-C. Zhu, "The Discrete Operator Method and Its Analysis on the Transient Behavior of Semiconductor," *Proceedings of Chinese Society of Industrial and Applied Mathematics (CSIAM) Conference*, pp. 512–516, Shanghai, China, September 1–4, 1996.

TEACHING EXPERIENCE:

- Instructor for STAT 356, Probability for Electrical and Computer Engineering, Department of Mathematics and Statistics, Queen's University, Winter term of 2002–2003.

- Special Tutor for the challenge problems of APSC 171 (first year Calculus for students in Mathematics and Engineering program), Department of Mathematics and Statistics, Queen's University, September 2002–August 2003.

- Tutorial Lecturer for APSC 171, 172, (Calculus), APSC 174 (Linear Algebra). Department of Mathematics and Statistics, Queen's University, September 2000–August 2002.

- Teaching Assistant for various undergraduate courses, including Probability and Stochastic Processes, Information Theory, Source Coding, Differential Equations, Fourier Transform, etc. Department of Mathematics and Statistics, Queen's University, September 1997– August 2000.

- Replacement Lecturer for courses of MATH 474/874, Information Theory, (November 6–10, 2000), and STAT 356, Probability for Electrical and Computer Engineering, (March 15–19, 1999, and March 13–17, 2000), Department of Mathematics and Statistics, Queen's University.

- Assistant in supervising undergraduate fourth year projects, Department of Mathematics and Statistics, Queen's University, September 1999–April 2000.

- Teaching Assistant at Shandong University (September 1995–June 1996): Tutoring undergraduate courses mainly in Calculus and Numerical Analysis.

MEMBERSHIPS IN PROFESSIONAL ORGANIZATIONS:

- Institute of Electrical and Electronics Engineers (IEEE) student member.

- IEEE Communications Society student member.

- American Mathematical Society (AMS) student member.

EXTRACURRICULAR ACTIVITIES:

- Tai-Chi Instructor for over 30 seniors in Kingston, November 1998–present, (all for free). (I learned Tai-Chi under the instruction of Mr. Jin-Xian Sun from 1991 to 1995 and won the Champion of *Tai-Chi Sword* in the 1994 Martial Art Competition at Shandong University, China).

- VP-Internal of Queen's Chinese Students and Scholars Association, September 1999–August 2000.