REPEAT-PUNCTURED TURBO CODED COOPERATION

by

Jules Merlin Mouatcho Moualeu

This thesis is submitted in fulfillment of the requirements for the degree of Master of Science in Engineering (MScEng) in the School of Electrical, Electronic and Computer Engineering, University of KwaZulu-Natal.

March 2008.

Declaration

The research work presented in this dissertation was performed by Mr. Jules Merlin Mouatcho Moualeu, under the supervision of Prof. Hongjun Xu and Prof. Fambirai Takawira, at the University of KwaZulu-Natal's School of Electrical, Electronic and Computer Engineering, in the Centre for Radio Access and Rural Technologies, which is sponsored by Alcatel-Lucent and Telkom South Africa Ltd as part of the Centre of Excellence programme.

Parts of the work in this dissertation have been accepted for presentation at the 2008 IEEE Wireless Communications and Networking Conference (WCNC'08) in Las Vegas, Nevada, USA. Parts of this dissertation have also been submitted for publication in the SAIEE (South African Institute of Electronic Engineers) journal.

All the work in this dissertation is the author's original work, unless otherwise indicated, and has not been previously submitted in part or in whole for a degree in this or any other university.

As the candidate's supervisor and co-supervisor, we have approved this dissertation for submission.

Name: Prof. Hongjun Xu

Signed: _____

Date: _____

Name: Prof. Fambirai Takawira

Signed: _____

Date: _____

Acknowledgements

I would like to express my gratitude to my supervisor Prof. H Xu for his guidance, suggestions, valuable comments and support that helped me in completing this dissertation. I would also to express my appreciation to my co-supervisor Prof. Takawira, director of the Centre for Radio Access and Rural Technologies (C.R.A.R.T) for giving me the opportunity to pursue my studies at his laboratory. I am also thankful to Dr. Telex Ngatched for his support and for sharing his time and experience.

I am also indebted to my parents, my uncle J. Michel, my brothers and sister for their constant encouragement and endless support throughout this research work.

I would also like to thank my postgraduate friends at the C.R.A.R.T for their assistance and friendship. Finally, I would like to thank Lynne for the love, support and encouragement she has given me during these past eighteen months.

Abstract

Transmit diversity usually employs multiple antennas at the transmitter. However, many wireless devices such as mobile cellphones, Personal Digital Assistants (PDAs), just to name a few, are limited by size, hardware complexity, power and other constraints to just one antenna. A new paradigm called cooperative communication which allows single antenna mobiles in a multi-user scenario to share their antennas has been proposed lately. This multi-user configuration generates a virtual Multiple-Input Multiple-Output system, leading to transmit diversity. The basic approach to cooperation is for two single-antenna users to use each other's antenna as a relay in which each of the users achieves diversity. Previous cooperative signaling methods encompass diverse forms of repetition of the data transmitted by the partner to the destination. A new scheme called coded cooperation [15] which integrates user cooperation with channel coding has also been proposed. This method maintains the same code rate, bandwidth and transmit power as a similar non-cooperative system, but performs much better than previous signaling methods [13], [14] under various inter-user channel qualities.

This dissertation first discusses the coded cooperation framework that has been proposed lately [19], coded cooperation with Repeat Convolutional Punctured Codes (RCPC) codes and then investigates the application of turbo codes in coded cooperation.

In this dissertation we propose two new cooperative diversity schemes which are the Repeat-Punctured Turbo Coded cooperation and coded cooperation using a Modified Repeat-Punctured Turbo Codes. Prior to that, Repeat-Punctured Turbo codes are introduced. We characterize the performance of the two new schemes by developing the analytical bounds for bit error rate, which is confirmed by computer simulations. Finally, the turbo coded cooperation using the Forced Symbol Method (FSM) is presented and validated through computer simulations under various inter-user Signal-to-Noise Ratios (SNRs).

Table of Contents

List of Figures

List of Tables

List of Acronyms

3G	Third Generation
APP	A-Posteriori Probability
AWGN	Additive White Gaussian Noise
BCJR	Bahl, Cocke, Jelinek, and Raviv
BCH	Bose, Chaudhuri, and Hocquenghem
BER	Bit Error Rate
BLAST	Bell Labs Layered Space-Time Architecture
BLER	Block Error Rate
BPSK	Binary Phase Shift Keying
CDMA	Code Division Multiple Access
CRC	Cyclic Redundancy Check
DRPTC	Dual Repeat-Punctured Turbo Code
FER	Frame Error Rate
FSM	Forced Symbol Method
LAN	Local Area Network
LAPP	Log A-Posteriori Probability
LDPC	Low-Density Parity-Check (code)
LLR	Log Likelihood Ratio
LOS	Line of Sight
LVA	List Viterbi Algorithm
MAP	Maximum A Posteriori
MIMO	Multiple-Input Multiple-Output
ML	Maximum Likelihood
PCCC	Parallel Concatenated Convolutional Codes
PDA	Personal Digital Assistant
PEP	Pairwise Error Probability

RPTC	Repeat-Punctured Turbo Code
RS	Reed-Solomon
RSC	Recursive Systematic Convolutional
SCCC	Serial Concatenated Convolutional Codes
SISO	Soft-Input Soft-Output
SNR	Signal-to-Noise Ratio
SOVA	Soft Output Viterbi Algorithm
STC	Space-Time Coding
TC	Turbo Code
WLAN	Wireless LAN

CHAPTER 1

INTRODUCTION

1.1 The Wireless Channel

Wireless devices, technologies and networks such as cellular telephones, handheld Personal Digital Assistants (PDAs), bluetooth, walkie-talkie, Wireless LAN (WLAN) etc... are omni-present and remarkably used nowadays. This is due to the fact that wireless communication has increased to a great extent over the past decade. The main goal of wireless communications is to enable us to communicate with anybody from anywhere at anytime for anything flawlessly and in real or non-real time, which is impossible in most cases and impractical in wired communication systems. While accomplishing this goal, the challenge lies in how much information can be transferred through the system which is referred to as the capacity of the system. Two main characteristics of the wireless channel are its randomness and unpredictability. An important problem in the wireless channel is fading, which is a deep, delayed and phase shifted attenuation of the received signal caused by the out-of-phase reception of multipaths or multipath propagation. The time varying fading provokes distortion that is caused by the superposition of delayed, reflected, scattered and diffracted of multiple copies of the transmitted signal, each of them using a different path. The movements of the mobile units and objects in the environment cause variation over time which is also a source of difficulty in the wireless channel.

Single-antenna systems are communication systems used today in great number. As explained above, fading arises from multipath propagation present in wireless channel which causes the capacity of a single wireless channel to be very low. Single-antenna systems suffer another great setback which is, its high error rate. This is due to the fact that, the pairwise error probability (PEP), the probability of detecting a signal when a different signal is transmitted, decreases linearly with the signal-to-noise ratio (SNR) over a fading channel. Single-antenna systems are unlikely to be used since its disadvantages discussed above do not allow them to provide the needs of future wireless communications such as voice, multimedia and data services. It is then in this perspective that communication systems with high capacity and low error rate must be designed.

New communication systems have been introduced lately in order to meet the requirements of future wireless communication systems. These systems use multiple-input multiple-output (MIMO) wireless links, in other words, they use multiple transmit and receive antennas. The emergence of these systems is ranked among the most important discoveries in modern communications. MIMO systems have the ability to turn multipath propagation, causing unreliable communication, particularly in the case of single-antenna wireless channels, into a benefit to the users. Investigations on information theory in [1] and [2] have shown that MIMO communication systems can achieve much higher channel capacity [3] than single-antenna systems. It was also shown in [1] and [2] that the capacity increases linearly with the number of antennas at the transmitter or the number of antennas at the receiver, whichever is smaller. Multiple-antenna systems have been shown a lot of attention and great interest, because of its undisputed superiority over single-antenna systems. But results on MIMO systems based on Shannon capacity, do not represent in an appropriate way the performance of real transmission systems. Hence, the development of algorithms that take advantage of spatial diversity provided by multiple antennas is imperative and crucial. Various algorithms with fair complexity that yield reasonable performance such as the diversity schemes and diversity combining techniques [4, 22, 76, 77] have been proposed. Two of these algorithms include the Bell Labs Layered Space-Time Architecture (BLAST) and Space-Time Coding (STC). STC has attracted more attention than BLAST. The idea of

space-time coding was first proposed in [4] making use of the diversity (both transmit and receive) to improve the data rate as well as the bit error rate performance with no extra cost of spectrum over fading channels. The increase of the data rate and reliable communications is primarily due to the fact that transmission of independent copies of the signal generates diversity and consequently, can restrict the effects of fading in an effective way. In the case of MIMO systems, multiple replicas of the signal with decorrelated fading characteristics are transmitted from different spatial locations and obtained at the receiver(s), thus providing spatial diversity.

Despite all these advantages, transmit diversity is not practical in many scenarios. This is because certain wireless devices such as mobile handsets cannot accommodate more than one antenna due to their size, hardware complexity, power and other constraints. In the next section, cooperative diversity introduced in [5] and [6] is discussed. It is a new and effective method of generating spatial diversity in wireless networks.

1.2 Cooperative Communication

The basic idea behind cooperative communication is to have multiple mobiles, each of them with single antenna, sharing these antennas and create a virtual MIMO system that can reap the benefits of diversity.

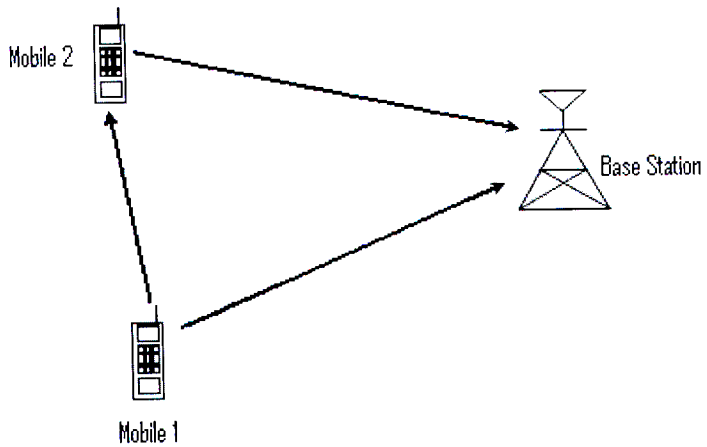

Figure 1.1: Cooperative communication between two mobiles and the base station.

Figure 1.1 illustrates communication between two mobiles and the base station or destination. Transmit diversity cannot be generated from each individual mobile or user because of their single antenna. Since antennas are almost omni-directional in wireless communication, users may hear each other, in which case they can retransmit a replica of the received information along with their own data. The fading paths from the two mobiles to the base station are independent, thus transmit diversity is generated. In the following sections, we review some background works on cooperative communication such as its genesis and different two-user signaling methods that have been proposed so far.

1.2.1 Background of Cooperative Communication

The basic idea behind cooperative communication can be traced back to the innovative work of Cover and El Gamal on the relay channel [7], more precisely on its information theoretic properties. Cover and Gamal's work focused on the analysis of the capacity of the three-node network consisting of a source, a relay and a destination. The relay channel model shown in Figure 1.2 works as follows: the transmitter A sends a signal X to both the relay B and destination C via channels 2 and 1, respectively. A noisy version of the signal X is then received by both B and C. Based on what is received from the transmitter A, the relay then transmits a signal X_1 to the destination. An assumption that all nodes operate in the same band was made. This allows the system to be separated into a broadcast channel from the point of view of the source and a multiple access from the point of view of the destination. Work on the relay channel first introduced by Cover and Gamal has been further developed. In [7], it was assumed that the relay can simultaneously receive and transmit in the same frequency slot due to the impracticality of current RF technology. Alternative capacity results on relay channel have also been developed in [8] for the systems in which the relay receives the source transmission in one band and then retransmits to the destination in a different frequency slot.

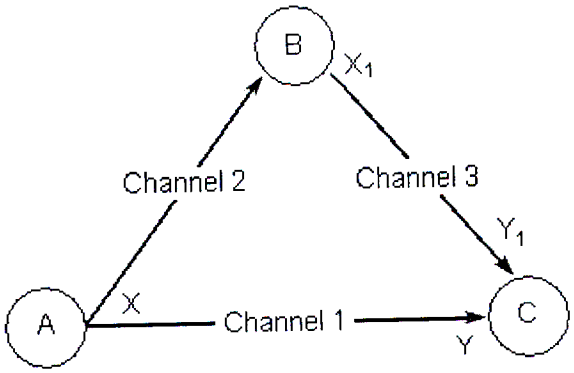

Figure 1.2: The relay channel.

The relay channel [7] and cooperative communication are different in many respects as elaborated in [9]. First of all, the work in [7] mostly analyzes capacity in an Additive White Gaussian Noise (AWGN) channel whereas latest developments are motivated by the concept of diversity in fading channel. Secondly, in cooperation the total system resources are fixed, and users act both as information sources as well as relays while in the relay channel, the relay's sole aim is to help the main channel. In the next section we review the main signaling schemes developed in cooperative communication.

1.2.2 Cooperative Signaling Methods

One of the dissimilarities between the relay channel model and cooperative communication mentioned previously is that, in cooperative communication each user is assumed to transmit data as well as play a partner role for another user. The cooperative signaling methods are therefore designed in such a way that users can help other users while still being able to send their own data, whenever that is possible. Figure 1.3 [9] shows the reciprocal arrangement in cooperative communication. The main signaling schemes previously proposed are detailed in the following sub-sections.

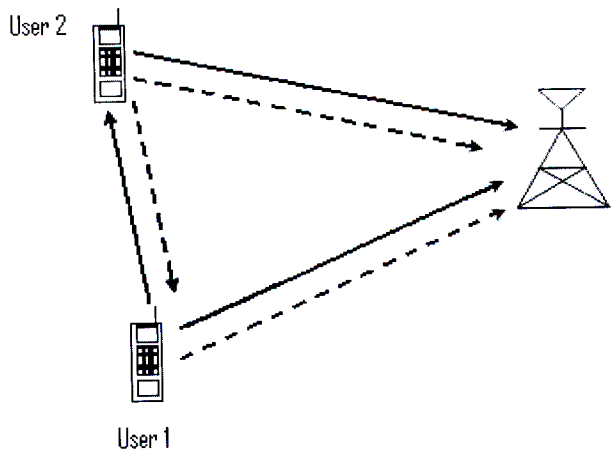

Figure 1.3: Reciprocal arrangements between two mobile users.

1.2.2.1 Detect-and-Forward Method

The detect-and-forward method was first proposed by Sendonaris et al. [5, 6, 10] and was the first work in the area of cooperative communication. In this method, both the users try to detect their partner's bits and then may retransmit estimates of the received bits obtained via hard detection. Figure 1.4 [9] illustrates a basic detect-and-forward method.

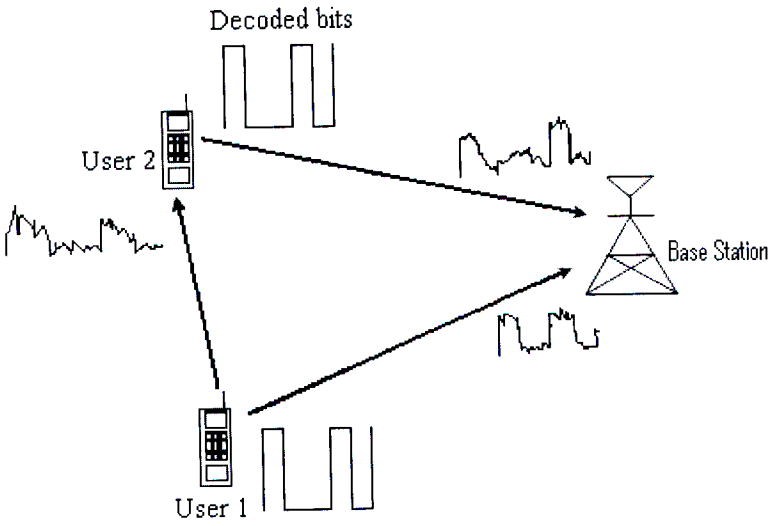

Figure 1.4: Detect-and-Forward signaling method.

1.2.2.2 Amplify-and-Forward Method

This method was proposed by Laneman *et al.* in [11]. In the amplify-and-forward method, each user receives a noisy version of the signal sent by the partner through the inter-user channel. The user then amplifies and retransmits the noisy signal to the base station as shown on Figure 1.5 [9]. Once the base station receives the noisy amplified version of the signal from each user, it then combines those signals and makes a final decision on the transmitted signal. The destination is still able to make better decisions on the transmitted signal, since it receives two statistically independent noisy versions of the signal, even though noise is amplified due to cooperation.

Laneman *et al.* analyzed this method for the two-user cooperation and showed that it achieves diversity of order two. In order to do optimal decoding, this signaling scheme assumes that the base station knows the inter-user channel coefficients. So techniques to estimate these coefficients must be integrated during the implementation process.

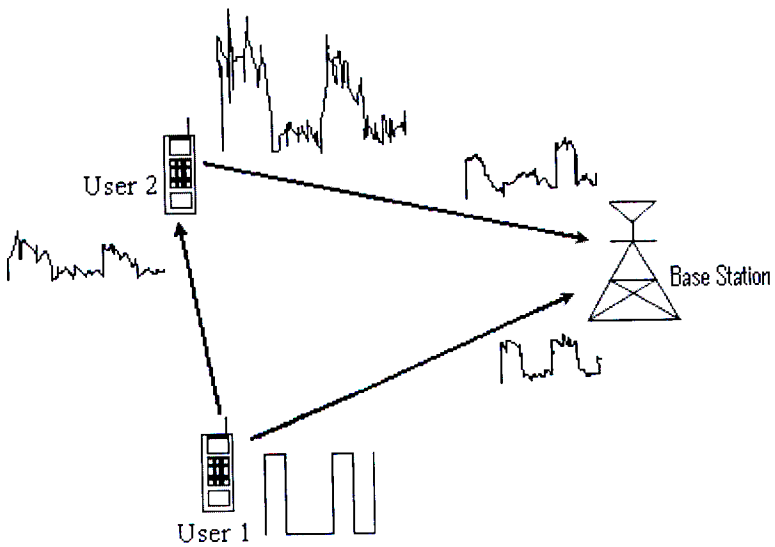

Figure 1.5: Amplify-and-Forward method.

1.2.2.3 Hybrid Decode-and-Forward Method

In order to avoid the problem of error propagation faced by the detect-and-forward and amplify-and-forward methods, a modified version of the decode-and-forward method termed hybrid decode-and-forward was proposed in [11, 12]. In this method, decode-and –forward is applied depending on the quality of the inter-user channel. That is, when the inter-user channel has high instantaneous signal-to-noise ratio (SNR), both users detect and forward their partner's data, but revert to a non-cooperative mode at times when the inter-user channel has low SNR. This protocol proposed by Laneman et al. is not similar to the method of Sendonaris et al. [5, 6], in the sense that the multiplicative factors $a_{i,j}$ do not adapt themselves based on the quality of the inter-user channel.

Laneman et al. [11] [12] showed that the hybrid decode-and-forward scheme does provide some gains over a non-cooperative mode similar to those of the amplify-and-forward signaling method [11] and also achieves diversity of order two.

1.2.2.4 Coded Cooperation

The protocols discussed above, however present some limitations. First of all, these signaling methods either include propagation of the partner's noise (amplify-and-forward) or allow forwarding of erroneous estimates of the partner's data (detect-and-forward protocol). Error propagation reduces the performance, especially when the quality of the inter-user channel deteriorates. Secondly, from a channel coding point of view, these schemes do not exploit the available bandwidth efficiently, since they involve some form of repetition. Thirdly, forwarding the analog signal in the amplify-and-forward method [11] takes place after some delay. This is impractical as the delay causes complications in storing sufficient information to reproduce the analog signal. Finally, knowledge of either the instantaneous SNR or bit error rate (BER) of the inter-user channel at the base station is required by the methods discussed above, for optimal maximum likelihood (ML) decoding or detection [13] [14].

To address these limitations, a new signaling protocol called *coded cooperation* was proposed in [15]-[19] in which cooperative signaling is integrated with channel coding. In the conventional user cooperation schemes (amplify-and-forward and detect-and-forward), the partners repeat the received data via forwarding or hard detection, whereas in coded cooperation framework, cooperation is done through partitioning a user's data or codeword. That is, one part of the codeword is transmitted by the user itself and the other part is transmitted by the partner both to the destination as shown on Figure 1.6 [9]. Whenever the user cannot transmit the incremental redundancy for its partner, the user automatically reverts to a non-cooperative mode. Thus, in the worst case scenario, coded cooperation performs as well as a comparable non-cooperative system. Another advantage of coded cooperation is that error propagation is avoided through error detection by the partner unlike previous schemes in which error propagation is unavoidable. It should also be noted that, no knowledge of the inter-user channel by the base station is required in order to perform optimal decoding. Coded cooperation also maintains the same bandwidth, code rate, information rate and transmit power as a similar non-cooperative system. Coded cooperation will be discussed in more detail in the next chapter since most of the systems discussed further in this dissertation are based on coded cooperation framework.

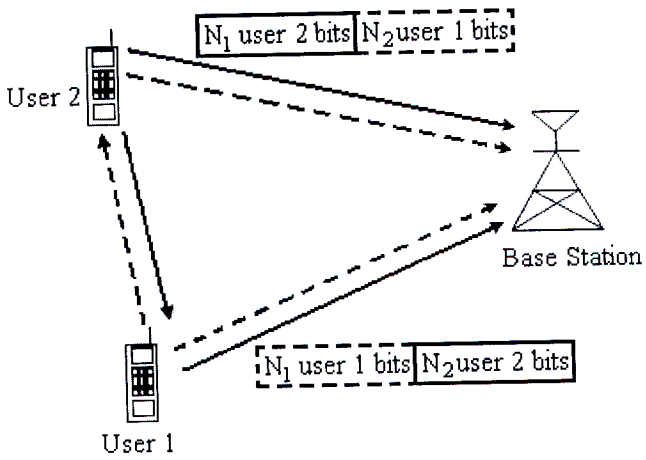

Figure 1.6: Coded cooperation scheme.

1.3 Dissertation Outline

The outline for the remainder of this dissertation is as follows. In Chapter 2, the coded cooperation scheme is reviewed in detail, starting with a description of its operation as well as implementation issues. We present and discuss the numerical results of the coded cooperation with rate-compatible punctured convolutional (RCPC) codes under various conditions. Turbo codes and their performance are also reviewed, starting with the description of the encoder and decoder structures. The principle of the Maximum A Posteriori (MAP) decoding algorithm is presented as well as a brief explanation of iterative decoding. An investigation of some of the factors affecting turbo code performance, such as the interleaver size, the number of decoding iterations, the generator polynomial and constraint length of the component codes, is performed. The application of turbo codes with coded cooperation is then investigated and the performance of the turbo coded cooperation is determined by computer simulations under various inter-user channel conditions.

Chapter 3 considers two extensions to turbo coded cooperation. The first extension involves implementing coded cooperation using repeat-punctured turbo codes. Prior to introducing the repeat-punctured turbo coded cooperation scheme, repeat-punctured turbo codes are investigated. The second extension to turbo coded cooperation is a modified version of the latter scheme, termed the modified repeat-punctured turbo coded cooperation scheme. Numerical results of both schemes are provided to demonstrate the performance improvement over the turbo coded cooperation under various inter-user channel qualities.

In Chapter 4, we develop the transfer function bounds and pairwise error probability expressions to obtain tight upper bounds on bit error rate. Transfer function bounds for repeat-punctured turbo codes are obtained by determining the term-by-term joint enumerator for all possible combinations of input weights and output weights of error events. The analyses of both schemes are then compared with the respective computer simulations presented in Chapter 3.

In Chapter 5, we present a new protocol termed turbo coded cooperation with the forced symbol method. The principle of the forced symbol method applied on single binary turbo codes is first presented and its performance is determined by computer simulations in the additive white Gaussian noise (AWGN) channel. Turbo coded cooperation using the Forced Symbol Method is presented. Two cases arise from this scheme: turbo coded cooperation scheme using the Forced Symbol Method at the destination and at each user. Simulation results show the performance of this new scheme and the gain achieved over turbo coded cooperation with standard decoding is discussed.

Chapter 6 presents conclusions by summarizing the results and work done in this dissertation and discusses some topics for future work in this area.

1.4 Original Contributions in this Dissertation

The original contributions of this dissertation can be summarized as follows:

1. In Chapter 3, we propose two extensions to coded cooperation using the repeat-punctured turbo codes (RPTC) and the dual repeat-punctured turbo codes (DRPTC) under quasi-static fading channel. System models of the proposed schemes are described and simulation results are presented
2. In Chapter 4, the theoretical bounds for bit error probability via numerical analysis of the proposed schemes described in Chapter 3 are presented. The numerical analysis of the proposed schemes confirms the simulation results performed in Chapter 3.
3. In Chapter 5, we present a scheme that improves turbo coded cooperation frame error performance under various channel SNRs. This scheme combines turbo coded cooperation with the Forced Symbol Method. Comparison of the proposed scheme and turbo coded cooperation is presented via computer simulations.

1.5 Published Work

The following publications have resulted from this work.

Conference Papers:

- J.M. Mouatcho Moualeu, H. Xu, and F. Takawira, "Double repeat-punctured turbo coded cooperative diversity scheme," accepted for presentation at the *IEEE Wireless Communications and Networking Conference (WCNC) 2008*, 31March-3April 2008, Las Vegas, Nevada, USA.

Journal Papers:

- J. M. Mouatcho Moualeu, H. Xu and F. Takawira, "Cooperative diversity using repeat-punctured turbo codes," Submitted for publication in the *SAIEE journal*.

CHAPTER 2

TURBO CODES AND TURBO CODED COOPERATION

In this chapter, we discuss the coded cooperation method that has been introduced in Chapter 1. The coded cooperation scheme can be implemented using various channel coding schemes. We first review the implementation of the coded cooperation using RCPC codes as developed in [15]-[19]. The application of turbo codes in coded cooperation is then investigated. Prior to that, a survey on turbo codes is given.

2.1 Overview of the Coded Cooperation Framework

As mentioned in the previous chapter, the basic idea behind coded cooperation protocol [19] is that each user transmits incremental redundancy for its partner whenever possible. If it is not the case, both users automatically return to a non-cooperative case. It should be noted that, all this is managed automatically through code design, with no need for feedback between the users. Coded cooperation occurs via partitioning, that is, a part of the user's data is sent by itself and the remainder of the data is transmitted by the partner to the destination. In order to avoid error propagation, this method uses error detection at the partner's node. It is shown in [15]-[19] that, partitioning of codeword and employing error detection in coded cooperation

could be implemented in a natural and simple manner by a method that uses error control codes.

This method operates by sending various portions of each user's data via two independent fading paths for two user cooperation. The users partition their source data into blocks that are appended with cyclic redundancy check (CRC) code [20], which is used for error detection. Each user's block is encoded into a codeword of $N = K/R$, where K is the total bits per source block, R is the overall code rate and N represents the total code bits per block. The CRC bits are included in the total bits per source block. After the encoding process, the N bits are divided into two segments containing N_1 bits and N_2 bits respectively. In the first time segment or frame, each user transmits a code partition $N_1 = K/R_1$ bits which is a valid codeword despite being weaker (codeword), to the destination and the partner. Both users try to decode the transmission of each other's partner. If the decoding is successful (determined by checking the CRC code), the users then calculate and sends the second code partition $N_2 = K/R_2$ bits of its partner in the second time segment as shown on Figure 2.1 [15, 19]. If the user does not successfully decode the transmission of its partner, the user transmits its own second code partition of N_2 bits. So each user always sends $N = N_1 + N_2$ bits per source block in total over the two time segments.

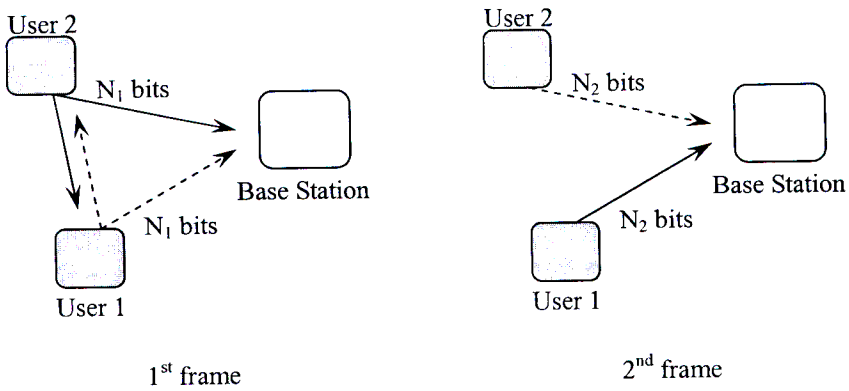

Figure 2.1: Transmission in coded cooperation.

In the second frame, both users work independently with no knowledge of what happened in the first frame. That is, whether their own first frame was correctly decoded or not. Consequently, four possible cases arise for the transmission of the second frame. Figure 2.2 [15, 19] illustrates the four cooperative cases that are likely to occur.

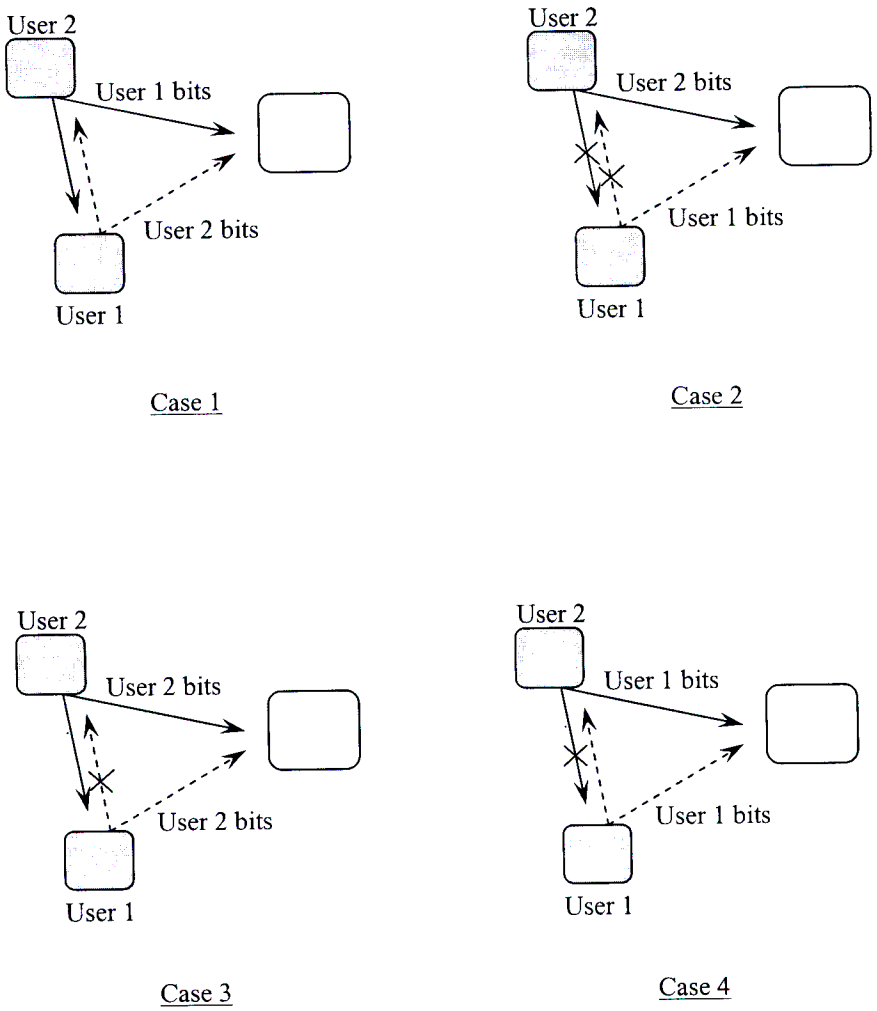

Figure 2.2: Cooperative cases for second frame transmission based on the first frame decoding results.

In Case 1, each user successfully decodes each other's first frame, so both users transmit their partner's second frame. This results in a full cooperation mode. In Case 2, neither User 1 nor User 2 successfully decodes its partner's first frame; thus each user transmits its own second

frame. This corresponds to the non-cooperative case, since both users send their own bits in both frames. In Case 3, User 1 successfully decodes User 2's first frame, but User 2 does not. So in the second time segment, both users transmit User 1's bits. Case 4 is similar to Case 3 with the roles of Users 1 and 2 interchanged. In Case 3 or 4, either User 2 or User 1 does benefit from cooperation while the other user does not. It should be noted that the destination must know which cooperative case has taken place in order to make correct decisions of the received bits.

2.2 Implementation Issues

The coded cooperation scheme offers great flexibility compared to just relaying the user's partner bits. It can be implemented with various codes such as convolutional codes, block codes or a combination of both classes of codes. Also the codeword partitioning can be achieved through product codes, puncturing or other forms of concatenation. The other advantage of coded cooperation over simple relay of the partner's codeword is that the level of cooperation which is defined as N_2 / N can be varied easily by changing the level of puncturing. This leads to various degree of cooperation and consequently matches the channel conditions. Namely, if the inter-user channel is poor, both users will cooperate less and if the inter-user channel is good both users will cooperate more. Hence the level of cooperation can be varied to adapt inter-user code rate as well as provide robustness for the system.

A simple but very effective implementation of coded cooperation using RCPC codes [21] is developed in [15]-[19]. In this implementation, the first frame codeword is obtained by puncturing a codeword of length N to obtain N_1 bits by applying the puncturing matrix corresponding to rate R_1. In the second time segment, the user transmits the remaining bits which are the punctured code bits, namely N_2 bits. The overall code rate R is chosen from a given RCPC code or mother code. Figure 2.3 [15, 19] illustrates a user's implementation of coded cooperation framework using RCPC codes.

This scheme operates as follows. In the first time frame, a user starts working by taking its own bits (N), appends a CRC and encodes it (in this particular implementation, RCPC codes [21] are used). Since this is a punctured code, the encoder outputs the puncture N_1 bits and the punctured N_2 bits. N_1 bits are first transmitted. In the same time, the partner is also transmitting and his partner is going to be received. A soft decision Viterbi decoder is then employed and two cases arise. The CRC either checks YES or NO. If the CRC checks YES, the partner recalculates the puncture bits of the user and then transmits in the second frame. If the CRC checks NO, the switch goes to the NO position and the user transmits the punctured N_2 bits itself. The Viterbi decoder used here works with 3-bit precision, that is, $2^3 = 8$ quantization levels. A Viterbi decoder working with 1-bit precision is called a hard decision decoder and does perform about 2dB worse than a soft-decision Viterbi decoder.

Four cooperative cases are possible and have been discussed above. This comes from the fact that each user acts independently in the second frame. In the particular implementation of coded cooperation with RCPC codes, if both users do not decode each other (CRC is unsuccessful), then both transmit their own puncture bits in the second frame. This is the worst case scenario and corresponds to the non-cooperation mode.

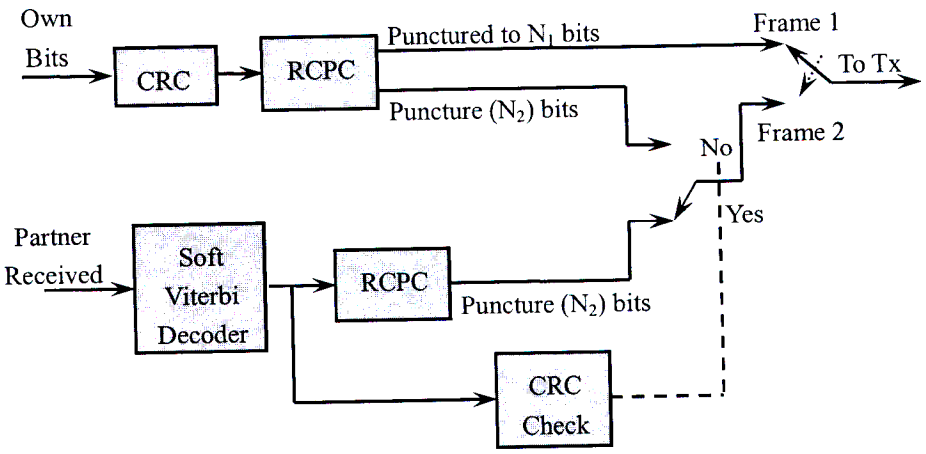

Figure 2.3: Implementation of coded cooperation with RCPC codes.

2.3 Performance of Coded Cooperation with RCPC Codes

In order to evaluate its performance, implementation of coded cooperation using RCPC codes as described in the previous section, is performed in [15]-[19]. A family of RCPC codes with memory $M = 4$ is used, puncturing period $P = 8$ and rate $1/4$ mother code given in [21]. Unless otherwise specified, the source block size is $K = 128$ bits. For simulation purposes, a 16-bits CRC with generator polynomial given by coefficients $(15935)_{HEX}$ is used. The results in Figures 2.4 and 2.5 were obtained they match those from [15]-[19]. All the simulations in this dissertation were performed with the C++ compiler.

Figure 2.4 illustrates the Bit Error Rate (BER) for quasi-static (slow) fading with different qualities of the inter-user channels. In quasi-static or slow fading, the fading coefficients are assumed to be constant over a certain period. The level of cooperation is 50% and it is assumed that both the uplink channels are statistically similar. The improvement of coded cooperation for various qualities of inter-user channels over the non-cooperative case is spectacular. For example, when the inter-user channel has 0dB average SNR, that is, the inter-user channel is sitting at 20dB below the uplink channels, the gain is about 2dB over the range of 0-20dB average uplink SNR. For the case of 10dB inter-user channel, that is, the inter-user channel is still worse than the uplink channels (10dB worse), it is still good to cooperate. The gain is about 9-10dB at a BER of 10^{-3} over no cooperation. The gain increases as the inter-user channel improves. The best scenario in coded cooperation is when the inter-user channel is much better than the uplink channels and is referred to as perfect inter-user channel (equivalent to Case 1 since both users always decode each other successfully). It can be noted that coded cooperation with a perfect inter-user channel [16] performs almost similarly to a comparable two-antenna transmit diversity system. The transmit diversity system used here, is the space-time block code proposed by Alamouti [22], concatenated with an outer code, in this case the RCPC codes that was used for coded cooperation. Therefore, coded cooperation does achieve full diversity of order two as the Alamouti code [22], for perfect inter-user channel or Case 1 [19].

Figure 2.5 show the BER performance comparison of coded cooperation with various degrees of cooperation, in this case 50% and 25% (level of cooperation) for both a 10dB inter-user channel and a perfect inter-user channel. The uplinks channels are assumed to be statistically similar. For the perfect inter-user channel, 50% cooperation (that is, 50% of the total code bits are transmitted cooperatively) performs better than 25% cooperation. This make sense since each user would like to cooperate more (by sending as many punctured bits as possible to the partner) when the inter-user channel is very good. When the inter-user channel deteriorates, this is no longer the case. Figure 2.5 shows that for the 10dB inter-user channel, the 25% cooperation outperforms the 50% cooperation by as much as 2dB for higher uplink SNRs. Hence, for poor quality inter-user channel the overall performance of the system depends on the strength of the first frame, i.e. the stronger the code in the first frame is, the better the overall performance becomes.

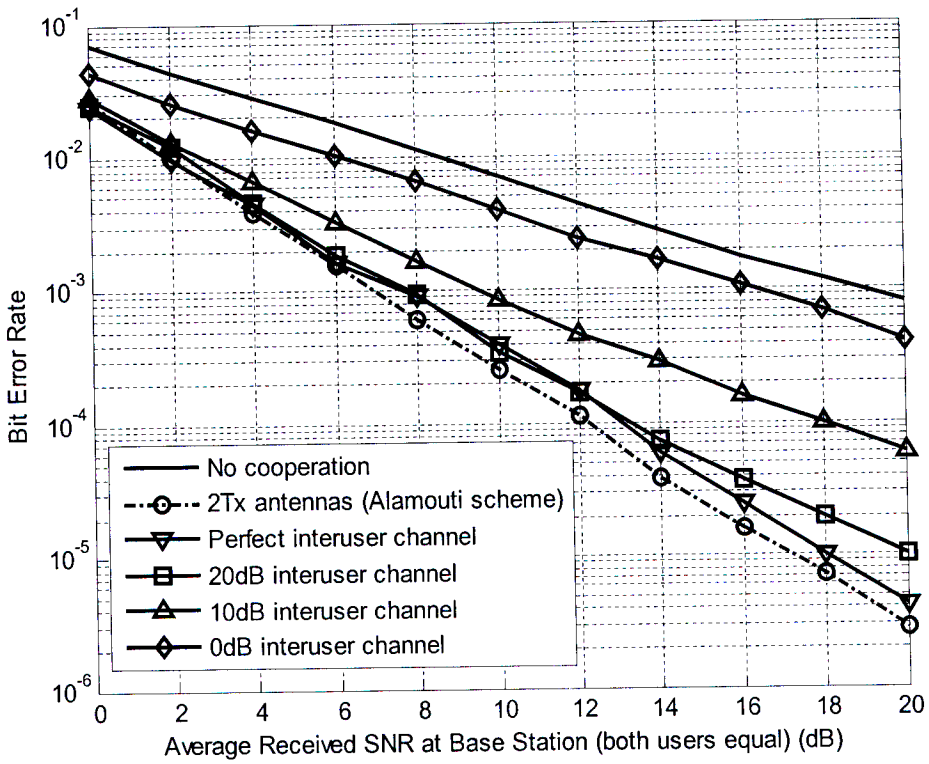

Figure 2.4: Performance in slow fading with 50% cooperation, equal uplink SNR and reciprocal inter-user channel.

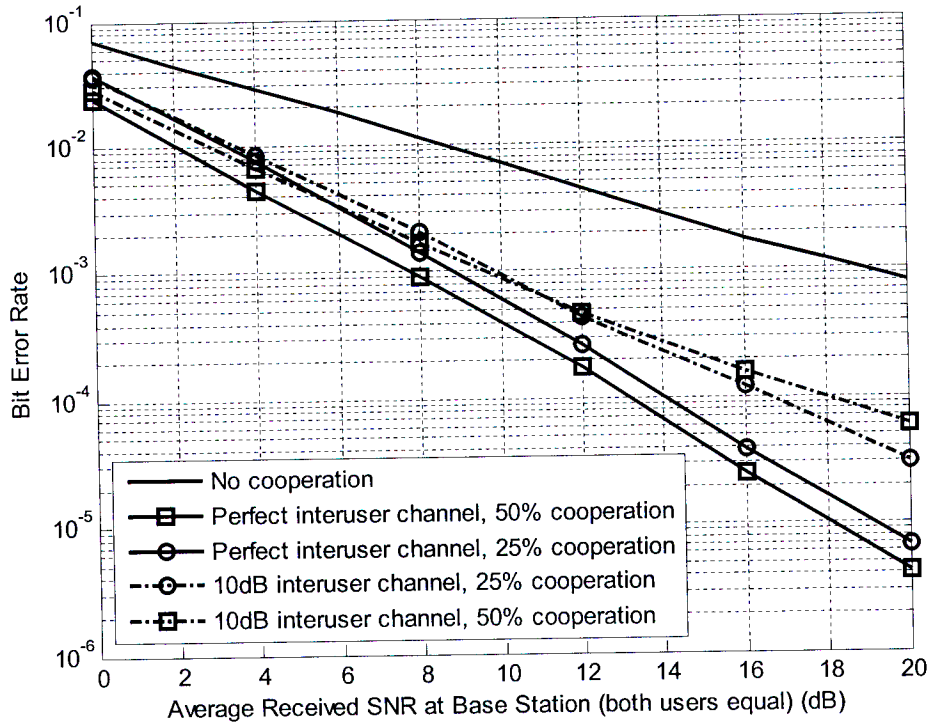

Figure 2.5: BER performance comparison of 50% and 25% in slow fading, equal uplink SNR, and reciprocal inter-user channel.

2.4 Iterative Decoding in Coded Cooperation

In this section, we discuss an extension to coded cooperation framework introduced in [16] [23]. This protocol termed turbo coded cooperation, significantly improves the BER performance of the system under various inter-user channel qualities. Coded cooperation involves two code components and therefore turbo codes are a perfect candidate to this framework. Prior to that, we give a survey on turbo codes.

2.4.1 Introduction to Turbo Codes

Turbo codes which are one of the most powerful error correcting codes presently available, were introduced in the early 90's by Berrou, Glavieux and Thitimajshima [24] [25]. Turbo

codes were reported to yield extremely impressive results. Berrou *et al.* showed that turbo codes could approach Shannon limit [26] to about 0.7dB with reasonable complexity at BER of 10^{-5}. They have become a popular area of communications research as well as being implemented into standardized systems such as Third Generation (3G) systems.

A conventional turbo encoder consists of two parallel concatenated recursive systematic convolutional (RSC) encoders with their inputs separated by an interleaver or permuter. An optional puncturer can be combined with the RSC encoders as in [1]. The turbo decoder comprises two Maximum A Priori (MAP) decoders connected in series through interleavers and deinterleavers. The output of the second decoder is fed back to the input of the first decoder. In the remainder of this section, we investigate turbo codes in detail by describing the structure of turbo encoders and decoders and by showing their performance with various parameters.

2.4.1.1 Turbo Code Encoder

The turbo code is also called parallel concatenated coding. This is because both encoders separated by an interleaver act on the same set of input bits, unlike a serial concatenated scheme in which the output of one encoder is encoded by the other encoder. For this reason, turbo codes are also referred to as parallel concatenated convolutional codes (PCCC) [27]. Figure 2.6 shows a block diagram of a standard turbo code encoder. Both convolutional encoders [69] are recursive systematic and are assumed to be identical. The choice of recursive convolutional encoders over non-recursive systematic encoders will be discussed later. In the remainder of this sub-section, we describe the building blocks that form a standard turbo code encoder: constituent encoders, interleaver and puncturer. In Figure 2.6, x_k^s, x_{1k}^p and x_{2k}^p represent the systematic, first parity and second parity bits respectively.

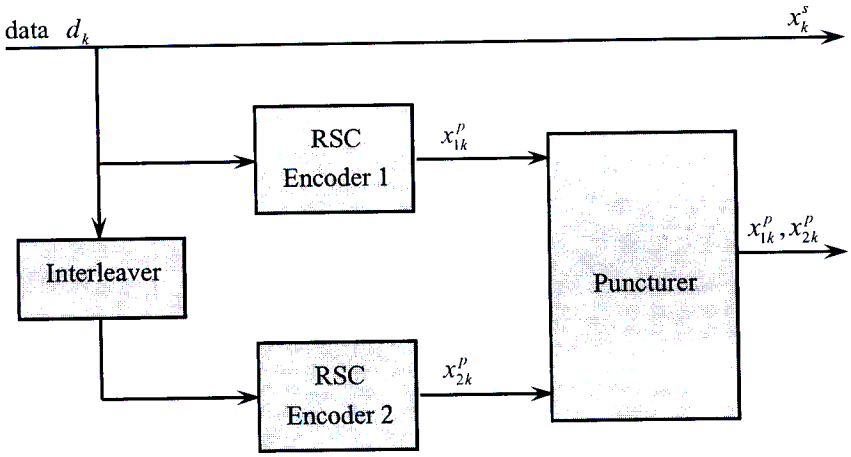

Figure 2.6: A standard turbo code encoder.

2.4.1.1.1 Recursive Systematic Encoders

Most convolutional codes have been used in their non-recursive or feed-forward form and are represented by a set of generator polynomials $G_{NR} = (g_1(D), g_2(D))_8$. Turbo codes use recursive systematic convolutional (RSC) encoders which are convolutional codes with feedback ("recursive or feedback") and in which the uncoded data sequence appears in the transmitted data sequence ("systematic"). A feedback or recursive systematic encoder can easily be obtained from a feed-forward convolutional code without changing its distance property, in other words feed-forward and feedback convolutional encoders generate the same set of encoded sequence. This is done by feeding back the generator polynomials $g_1(D)$, $G_R = (1, g_2(D)/g_1(D))_8$ where $g_1(D)$ is the feedback polynomial and $g_2(D)$ is the feed-forward polynomial, both in octal notation. The term $g_2(D)/g_1(D)$ is not an actual division but rather a notation allowing the feed-forward and feedback taps of the code to be distinctly defined. Figure 2.7 gives a description of a recursive systematic encoder, where $g_1(D) = 1 + D^2$, $g_2(D) = 1 + D + D^2$ (D is the delay operator and D^m represent a delay of m symbol times).

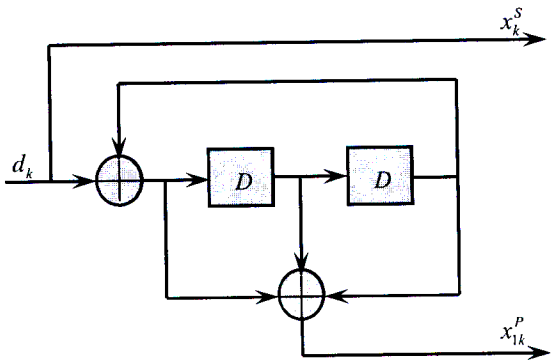

Figure 2.7: Recursive systematic convolutional encoder.

Unlike feed-forward convolutional encoders, recursive systematic encoders prevent the encoders from going back to the all-zero state by zero symbols. For a data sequence $d_k = (0,\ldots,0,1,0,\ldots,0)$ and assuming that the RSC codes terminate in the all-zero state, the data sequence d_k will end up with at least two non-zero bits or symbols. It is probable that one of the RSC codes have high inputs, since the data sequence d_k is permuted by an interleaver before going through the second RSC encoder. This is not the case for a non-RSC encoder, because a data sequence $d_k = (0,\ldots,0,1,0,\ldots,0)$ and its permuted version (despite the permutation) will always generate a low output codeword. It is known that codes with low minimum Hamming distance have poor error correction capability. This is why feedback convolutional or RSC encoders are suitable in turbo codes.

2.4.1.1.2 The Interleaver

An interleaver is a device for permuting or reordering bits or symbols in an information sequence. The primary role of the interleaver in communications is to protect data transmission against burst errors. This is done by spreading the errors apart in a systematic pattern. Another role of the interleaver is to decrease the number of codewords with small Hamming distance by breaking low weight input sequences that can generate low weight

outputs. The final function of the interleaver is to have the inputs to the two decoders uncorrelated, so that an iterative decoding algorithm based on information exchange between the two decoders can be employed. There is a high probability that after correction of some errors in the first component decoder, some of the remaining errors can be corrected in the second decoder provided that the inputs sequence to both encoder are uncorrelated.

In turbo coding, structured codes do not perform as good as random codes. Codes with structure can be obtained not only with block interleaver which is a rectangular matrix that permutes the input sequence in a systematic fashion, but also with random interleavers. But some structured codes can permit decoding with reasonable complexity. Berrou *et al.* introduced a pseudo-random interleaver to solve this coding dilemma. Also the size of the interleaver or the number of elements to permute N should be large, since large block-length random codes approach the capacity limit.

2.4.1.1.3 The Puncturing Unit

The code obtained from a turbo code encoder shown in Figure 2.6 without the puncturer is a $(3N, N)$ linear block code. Puncturing is the process of periodically deleting some selected bits from the codeword. The main role of the puncturer is to increase or vary the rate of the overall system. In turbo coding, puncturing is only applied on both parity bits or symbols of both encoders and can lead to rates of $1/2, 3/4 \ldots$ depending on the puncturing matrix used. Rates $1/2$ and $3/4$ can be obtained from the unpunctured $(3N, N)$ code by using the respective puncture matrices:

$$P_{1/2} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \text{ and } P_{3/4} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}.$$

2.4.1.1.4 Trellis Termination

The term *trellis termination* refers to a process in which the encoder is driven back to the all-zero state. As discussed above, in turbo coding it is impossible to terminate the trellis of any of the encoders by transmitting the all-zeroes tail bits. This is because the encoders are recursive. It is very difficult to terminate both encoders simultaneously with the use a pseudo-random interleaver, because the terminating sequence of the first encoder is interleaved and may not by itself terminate the second encoder. Interleaver designs have been devised [29-32] which can drive both encoders back to all-zero state. It was shown in [28] that the performance degradation produced by terminating both encoders is negligible for large interleaver size.

We assume in the sequel of the dissertation that only the first encoder is terminated leaving the second encoder in an unknown state.

2.4.1.2 Turbo code Decoder

A block diagram of the turbo code decoder structure is shown in Figure 2.8. The two constituent decoders are *soft-in soft-out* (SISO) and are linked by interleavers in a manner reminiscent of the turbo code encoder structure. Each of the SISO decoders takes three inputs:

- The corrupted version of the transmitted systematic symbols y_k^s, from the output of the demodulator.
- The faded version of the transmitted parities associated with the SISO, that is, y_{1k}^p for decoder 1 and y_{2k}^p for decoder 2, from the output of the demodulator.
- The information referred to as *a-priori*, from the other decoder about the likely values of the bits concerned. The *a-priori* information is also termed intrinsic information [24].

In other words, the component decoder 1 is used to decode the sequences from the encoder 1 and the same applies for decoder 2. Each decoder provides two soft outputs:

- An extrinsic information [24] which is to be exchanged between the two SISO decoders. It is the additional information provided by the decoder based on the *a-priori* information and the received sequence.
- A posteriori information which is the information given by the decoder considering all variable sources of information about a particular bit.

We assume BPSK modulation is used in the following discussion. The outputs of the two decoders are expressed in terms of the Log Likelihood Ratios (LLRs), which are the logarithm of the ratio of the two probabilities (for single-binary turbo codes). For example the LLR for the value of a decoded bit $\tilde{d}_k$ (*a-priori* information) is given by

$$L(\tilde{d}_k) = \ln\left(\frac{p(\tilde{d}_k = +1)}{p(\tilde{d}_k = -1)}\right), \tag{2.1}$$

where $p(\tilde{d}_k = +1/-1)$ is the probability that the decoded bit $\tilde{d}_k = +1$ or $\tilde{d}_k = -1$ respectively. The LLRs give two valuable information: the amplitude of the probability for the correct decision and its magnitude which determines the sign of the bit.

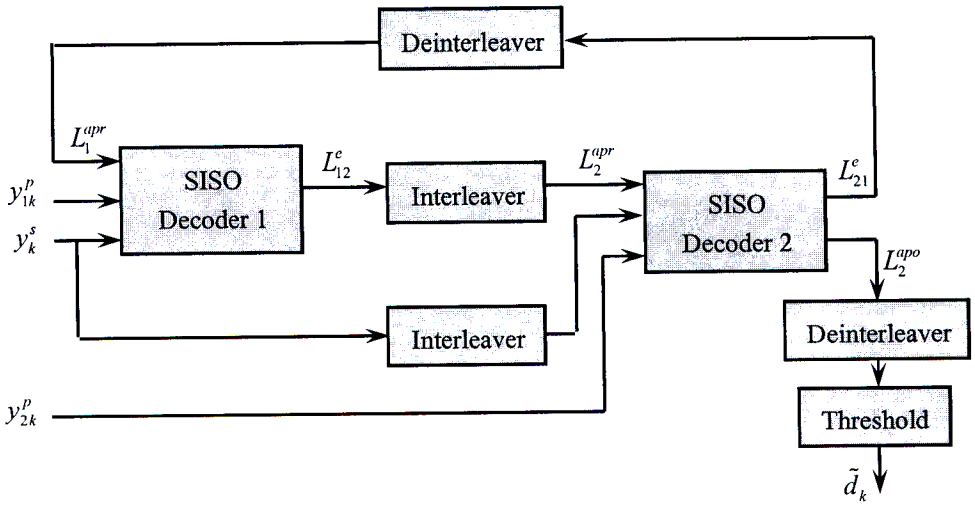

Figure 2.8: Structure of an iterative (turbo code) decoder.

There are two types of SISO decoding algorithms which are appropriate for turbo codes:

MAP algorithm also known as BCJR algorithm introduced by Bahl, Cocke, Jelinek and Raviv [33] and the Soft Output Viterbi Algorithm (SOVA) proposed by Hagenauer and Hoeher [34]. In this dissertation, we will only describe the MAP algorithm and its modifications, since the performance of the MAP algorithm is better than SOVA.

2.4.1.2.1 Iterative Decoding

The turbo code decoder shown in Figure 2.8 works in an iterative manner. In the first iteration, the first MAP decoder takes as inputs the received sequences of the systematic y_k^s and first parity symbols y_{1k}^p from the output of the demodulator and calculates a soft output as its estimate of the information sequence. This soft output is the extrinsic information L_{12}^e. L_{12}^e is then interleaved, using an interleaver similar to the one used in the turbo code encoder, and then used as a priori information L_2^{apr} to the second MAP decoder along with the other channels outputs, which are in this case, an interleaved version of the systematic bits and the second parity bits y_{2k}^p. The second MAP decoder yields the extrinsic information L_{21}^e which is deinterleaved before being fed back to the first MAP decoder as its *a-priori* information before the second iteration starts. This cycle is repeated and after reaching a number of iterations, the second decoder yields *a posteriori* information L_2^{apo} which is deinterleaved and then passed through a threshold to determine the decoded bit $\tilde{d}_k$.

The BER performance of the turbo codes improves as the number of iterations used increases. The error performance of most error correcting codes is divided into three regions as shown in Figure 2.9. In the low SNR region, the error performance of turbo codes is very poor as it is the case for other error correcting codes. However the error rate can be lowered by increasing the number of iterations, but is not appropriate for most communication systems. In the waterfall region, the error performance drops quickly and can be further lowered by increasing the number of iterations. In the error floor region, the error rate is almost flat and

does not improve much after a certain number of iterations.

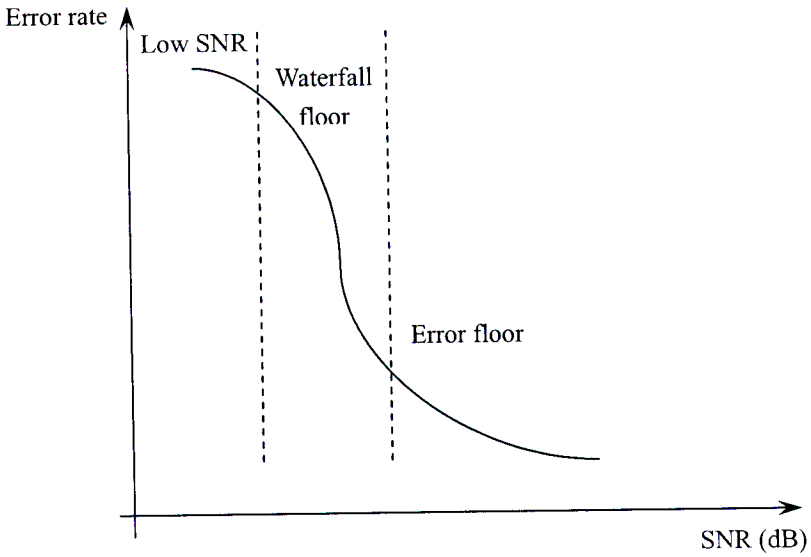

Figure 2.9: Regions of Turbo code error rate performance.

2.4.1.2.2 The MAP Algorithm

The MAP algorithm also referred to as the BCJR algorithm or a posteriori probability (APP) proposed by Bahl *et al.* [33] was modified for recursive convolutional codes by Berrou *et al.* [24] to produce APP for each information symbol. The reasons of using MAP algorithm over the Viterbi algorithm [69, 72, 73] are various. Firstly, unlike the Viterbi algorithm which finds the most likely data sequence initially sent, the MAP algorithm determines the most likely information symbol given the received coded sequence. Secondly, as mentioned above MAP which is also SISO algorithm yields a priori probabilities, appropriate for iterative decoding, whereas Viterbi algorithm is a hard output algorithm making it inappropriate for iterative decoding. Some modified versions of the MAP algorithm have also been proposed in order to reduce the complexity of the original MAP algorithm. The derivation of the MAP algorithm has been well documented and described in [24, 28, 33, 35 and 36] and is repeated in Appendix C.

2.4.1.2.3 Simplifications of the MAP Algorithm

Some modifications of the MAP algorithm have been proposed in order to reduce its complexity due to the multiplications and divisions required to compute the forward and backward recursive terms. Koch and Baier [37], and Erfanian *et al.* [38] proposed the Max-Log-MAP algorithm, in which the multiplications and divisions are transformed to additions and subtractions and an approximation is used as shown in (2.2), thus reducing the computational complexity of the MAP algorithm,

$$\ln \sum_j e^{x_j} \approx \max_j (x_j). \tag{2.2}$$

But its error rate performance diminishes due to this approximation. Another algorithm that reduces the deterioration in performance by correcting the approximation while still reducing the complexity of the MAP algorithm was proposed in [39]. This algorithm is known as the Log-MAP algorithm with a performance similar to that of the MAP algorithm.

2.4.1.3 Effects of Various Parameters on Turbo Code Performance

The performance of turbo codes is affected by various parameters as described in [36], some of which are:

- The interleaver size or frame-length.
- The number of decoding iterations used.
- The component decoding algorithm used.
- The generator polynomials and constraint length of the component codes.
- The puncturing pattern on component codes.
- The design or type of interleaver used.

In this sub-section, we illustrate the performance of turbo codes using Binary Phase Shift Keying (BPSK) over AWGN channels as a function of the interleaver size or frame- length, the number of decoding iterations used, puncturing pattern on component codes used and the generator polynomials. The RSC encoders of memory length $m = 3$ and generator polynomials $(1, 15/13)$ in octal representation are used. The MAP decoding algorithm is used in all the simulations, with a total of 8 iterations, a pseudo-random interleaver and a

terminated first component code unless specified otherwise.

2.4.1.3.1 Turbo Code Performance as a Function of the Interleaver Size

The performance of conventional turbo codes greatly depends on the interleaver size or frame-length. Many research papers on turbo codes including the seminal paper by Berrou *et al.* [24] have presented impressive performance results for large frame lengths. Namely, the performance of the turbo codes improves as the frame length increases which results in the code's free distance being dependent on the interleaver size. This comes at a cost, since large frame lengths generate large delays. The large interleaver length systems would only be important in non-real time applications where large delays are acceptable. Figure 2.10 shows the BER performance of turbo codes versus increasing frame length. We observe that the BER dramatically decreases as the interleaver size increases.

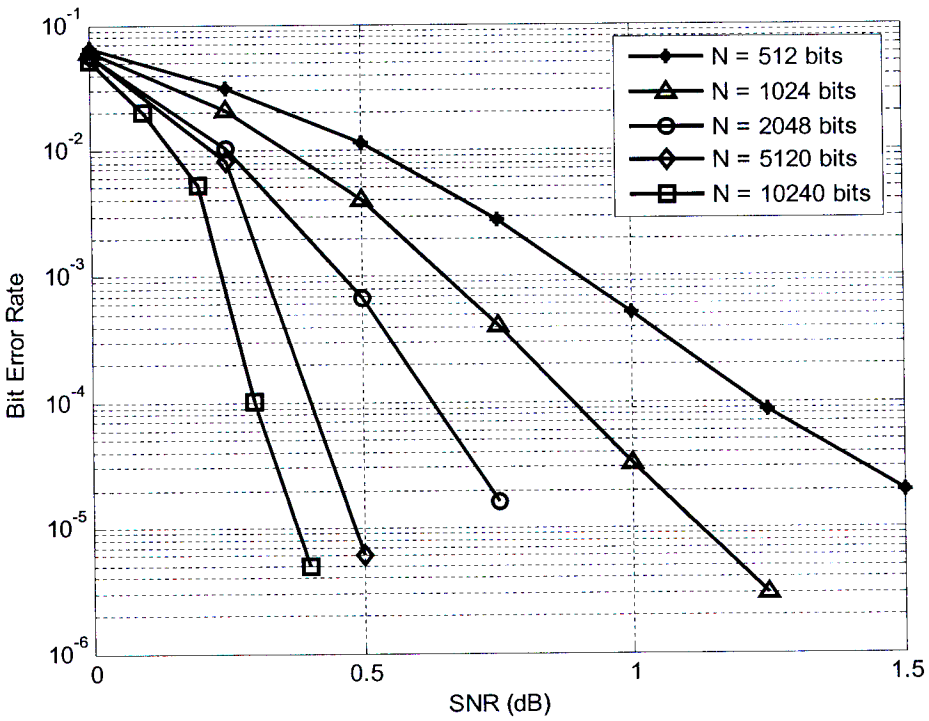

Figure 2.10: BER performance of an 8-state, rate 1/3 turbo code with MAP algorithm on an AWGN, interleaver length 1024 bits, 8 iterations.

2.4.1.3.2 Turbo Code Performance as a Function of the Number of Iterations Used

Turbo code decoder uses an iterative decoder (MAP decoding algorithm) and yields error rates which are function of the number of the iterations used. Figure 2.11 shows the BER performance of a turbo code decoder versus the number of iterations used. The BER performance improves as the number of iterations used by the turbo code decoder increases.

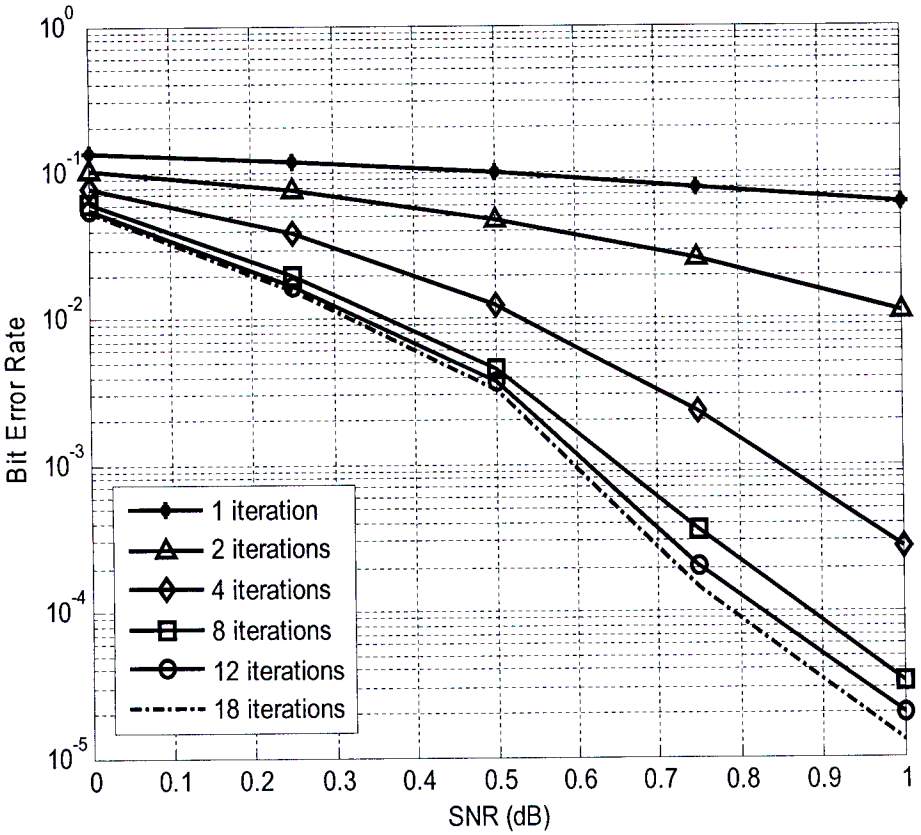

Figure 2.11: BER performance of an 8-state, rate 1/3 turbo codes using different number of iterations of MAP algorithm on AWGN channel, inerleaver size 1024 bits.

2.4.1.3.3 Effect of Puncturing Component Codes on Turbo Code Performance

The overall code rate of turbo codes can be increased by puncturing the parity bit sequences. For a rate 1/3 turbo code, puncturing can achieve rates 1/2, 2/3, 3/4 ... Figure 2.12 compares the performance of rates 1/3 and 1/2 turbo codes, where the rate 1/2 turbo code is obtained by puncturing half of both parity bit sequences. The increase in rate comes with a little performance loss of about 0.7dB at a BER of 10^{-4}.

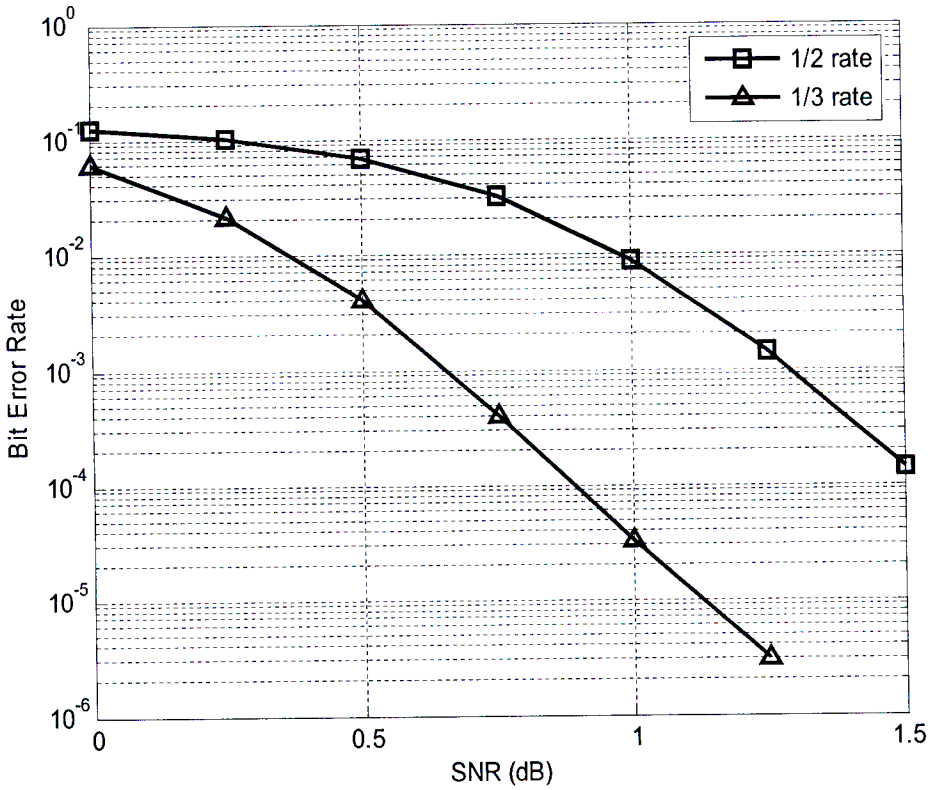

Figure 2.12: BER performance comparison between rates 1/2 and 1/3 8-state turbo codes in AWGN channel.

2.4.1.3.4 Effect of the Generator Polynomials on Turbo Code performance

As shown in Figure 2.13, the BER performance of turbo code is affected by different generator polynomials of the component codes. We note that by swapping $(13,17)_8$ and $(17,13)_8$, this leads to a significant degradation in turbo code. This is not the case for convolutional codes, as they both have same distance properties. It is important to mention that the generator polynomials yielding the best performance for convolutional codes do not yield the same results (good performance) for turbo codes due to the interleaver.

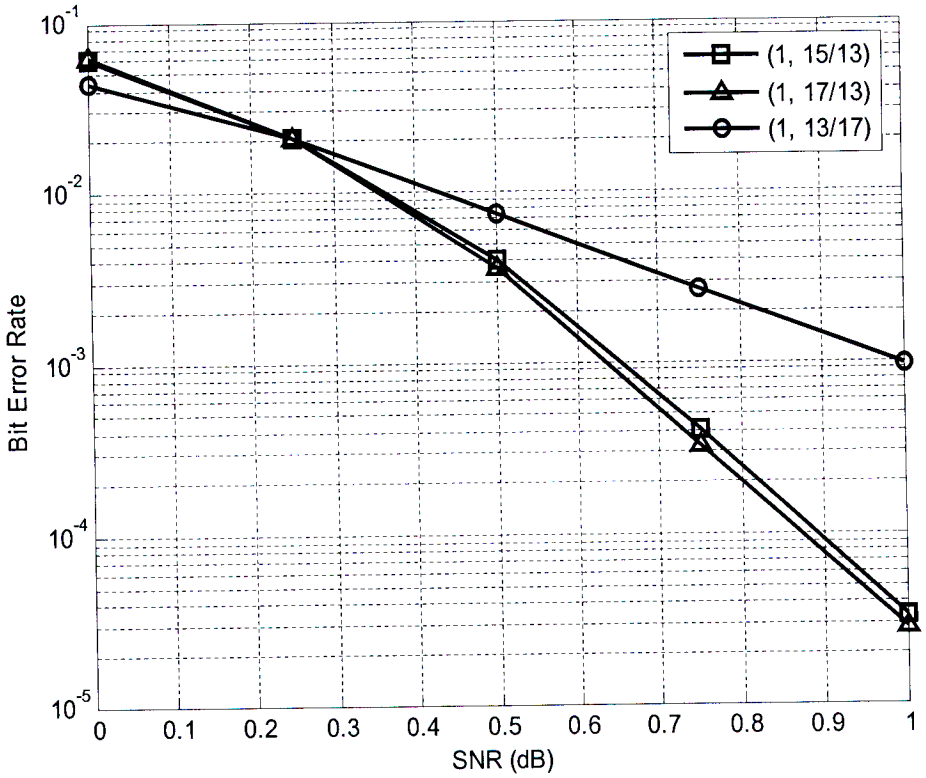

Figure 2.13: BER performance of an 8-state, rate 1/3 turbo codes using different generator polynomials.

2.4.2 Turbo Coded Cooperation

2.4.2.1 Implementation of Turbo Coded Cooperation

Figure 2.14 [23] shows a block diagram of coded cooperation using turbo codes. This is a rather straightforward extension to coded cooperation using RCPC codes. In the first time segment, because of the RSC encoders, each user transmits the systematic and first parity bits which constitute the frame 1. Prior to that, CRC bits are appended to the bits to be transmitted. Each partner receives frame 1 and attempts to decode. If the user decodes successfully its partner's bits, it then interleaves the systematic bits, sends it through the second RSC encoder. The second parity sequence is combined with frame 1(systematic + first parity sequence from the other user) at the destination where turbo decoding is performed. If the CRC does check NO, in other words, the decoding has been unsuccessful, each user interleaves its own systematic bits sequence and encodes it (second RSC encoder) to get the second parity sequence in frame 2. This is merged with frame 1 at the destination prior to turbo decoding.

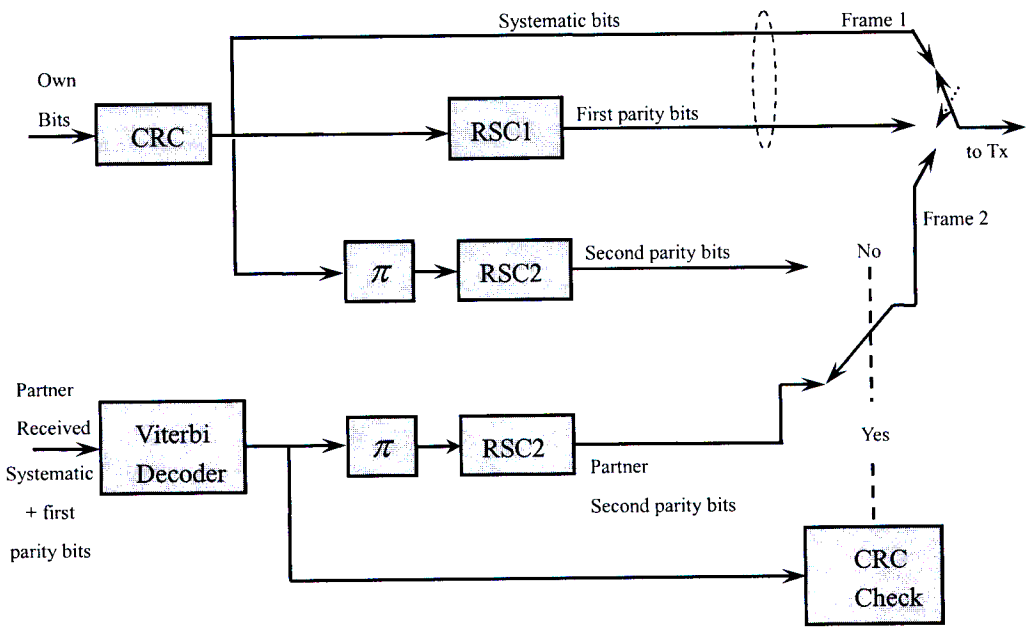

Figure 2.14: Turbo code encoding in coded cooperation framework.

Near-optimum decoding performance of turbo codes is obtained with low complexity iterative decoder [24, 40]. The description of iterative decoding turbo codes has been discussed in great detail above as documented in [24, 28, 33, 35 and 36].

2.4.2.2 Performance Evaluation

Figure 2.15 illustrates the performance of turbo coded cooperation in quasi-static fading under various scenarios. The code used for computer simulations is an 8-state turbo code with generator polynomials $G=(1,15/13)_8$ and 1/2 rate RSC encoders. The length of the source block code is $K=128$ bits with an overall code rate of 1/3. The level of cooperation is 33% and it is assumed that the uplink channels are statistically similar. The improvement of coded cooperation for various qualities of inter-user channels over the non-cooperative mode is dramatic. There are approximately 6dB, 8dB and 10dB gain at 10^{-3} for the respective 6dB, 12dB and perfect inter-user channels in slow fading.

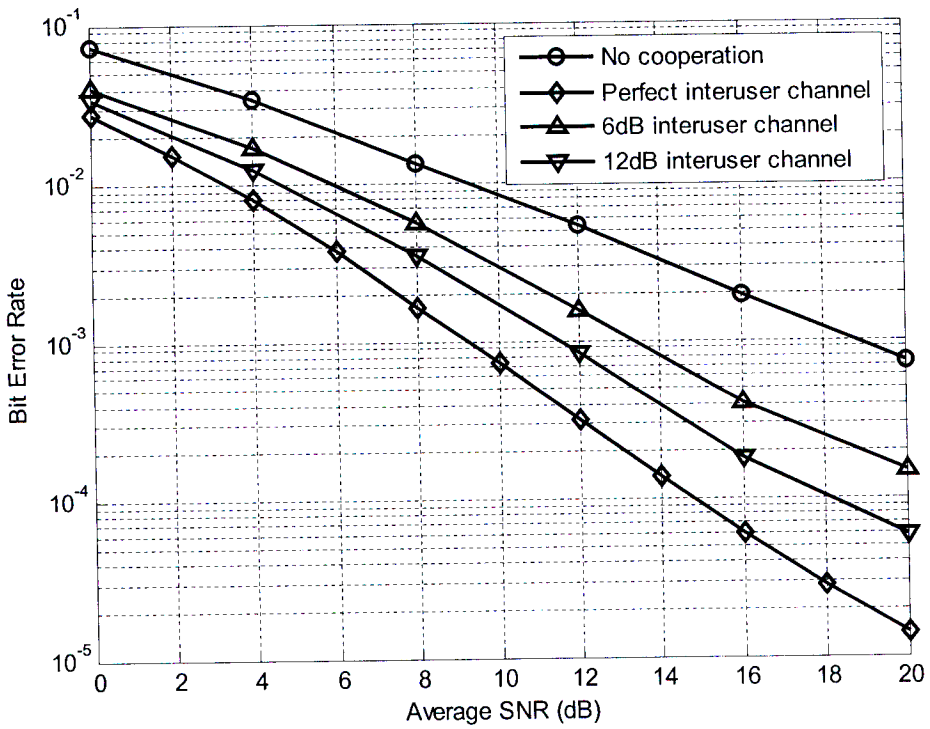

Figure 2.15: Turbo coded cooperation in slow fading, users have equal uplink channel.

2.5 Conclusion

This chapter reviews the coded cooperation framework beginning with the coded cooperation using RCPC codes. A description of its operation and implementation is given. Simulation results of this scheme are then presented to illustrate the performance improvement of cooperation over the comparative non-cooperation mode under various conditions of the inter-user channel. Turbo codes are investigated in great details. This investigation starts with the description of the turbo code encoder and decoder structures. An explanation of iterative decoding is given prior to the complete description of MAP decoding algorithm. Some parameters affecting turbo codes performance are briefly discussed. Finally an extension to coded cooperation framework is presented. This extension which involves turbo codes in coded cooperation is termed turbo coded cooperation. The results obtained in this chapter confirm that, coded cooperation using either RCPC codes of Turbo Codes under different inter-user SNR values yields impressive performance gain over the corresponding non-cooperative system, namely, cooperation is beneficial to both users.

CHAPTER 3

REPEAT-PUNCTURED TURBO CODED COOPERATION

In this chapter, two extensions to the coded cooperation using turbo codes that improve performance under different inter-user channel qualities are proposed. The first scheme uses a modified structure of turbo codes termed repeat-punctured turbo codes (RPTCs) and we refer to it as repeat-punctured turbo coded cooperation. In RPTCs, interleaver of size larger than the data frame length, is possible due to the use of a repeater-puncturer combination. Increasing the interleaver length improves the distance spectrum of turbo codes.

The second extension to the turbo coded cooperation uses a modification structure of RPTCs, which is called dual repeat-punctured turbo codes (DRPTCs) and is referred to as dual repeat-punctured turbo coded cooperation. DRPTCs use a double combination of repeater-puncturer for both RSC encoders. We demonstrate via computer simulations that dual repeat-punctured turbo coded cooperation outperforms coded cooperation with the conventional turbo codes and also improves performance over non-cooperative dual repeat-punctured turbo coded systems with the same complexity.

Firstly, a review of RPTCs is presented. Secondly, its applications in coded cooperation with a system model and an implementation of RPTCs in coded cooperation are described. We

also give a description of dual repeat-punctured turbo coded cooperation in a similar manner to coded cooperation using RPTCs. Simulations results are presented to show performance improvement over turbo coded cooperation.

3.1 Review of Repeat-Punctured Turbo Codes

The effect of increasing interleaver size or frame length on the performance of turbo codes is discussed in Chapter 2. It is shown that turbo codes performance gradually improves as the interleaver length increases. This is primarily due to the interleaver gain [27, 41]. As the interleaver length increases, the distance spectrum of the code becomes thinner and the performance of turbo code is dominated down to smaller SNRs by the error floor [42, 43]. In other terms, the free distance (the minimal Hamming distance amongst encoded sequences) asymptote dominates the turbo code performance for low SNR values to achieve near-capacity performance. A pseudo-random interleaver [24] is used to randomize information bits in order to minimize the chance of both RSC encoders generating low-weight codewords. This reduction in low-weight parity sequences leads to spectral thinning. Figure 3.1 [42] illustrates the effect of increasing the interleaver size on distance spectrum. N_d in Figure 3.1 refers to the multiplicity of codewords of weight d.

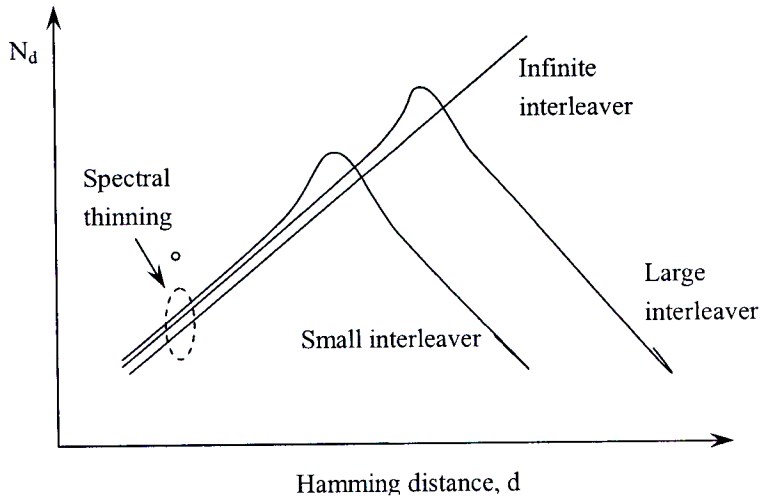

Figure 3.1: Spectral thinning illustration.

In the conventional turbo codes, the interleaver length is identical to the frame length. Hence, increasing the interleaver size requires increasing the information frame length by the same factor. The increase in frame length comes at a cost, since the larger the frame length or interleaver size, the longer the delays. Thus, TCs with large interleavers cannot be used in real time applications requiring low transmission delays such as voice applications [70], [71].

Kim *et al.* [44] proposed a turbo code structure employing a repeater and a puncturer in order to allow the use of interleavers of sizes larger than the information frame length. This proposed TC scheme with modified structure is termed repeat-punctured turbo codes.

3.1.1 Repeat-Punctured Turbo Code Encoder

A block diagram of a repeat-punctured turbo code encoding structure is shown in Figure 3.2 [44].

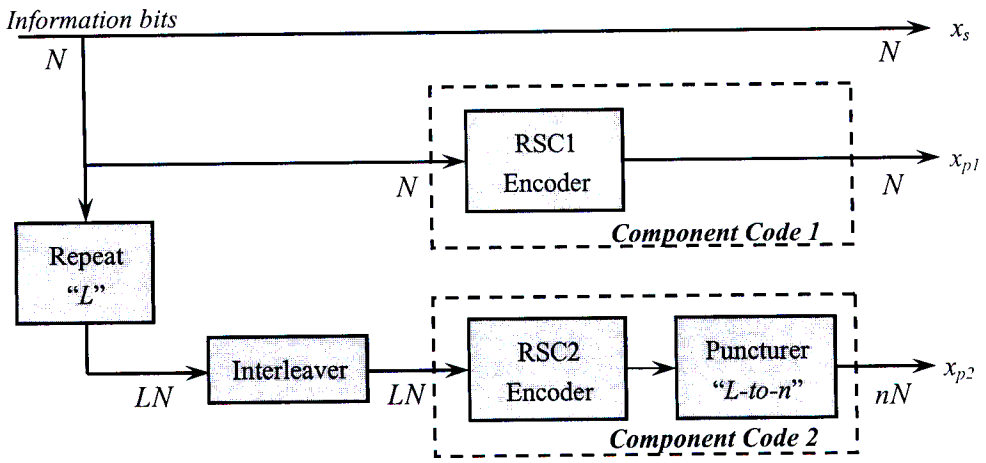

Figure 3.2: Structure of RPTC encoder.

The encoder structure of RPTC consists of two RSC encoders concatenated in parallel reminiscent of TC encoder structure. The encoder of the first component code is similar to the corresponding TC encoder. However, the second component code is not identical to the one generated by the second RSC encoder of TC. In the conventional TC, the information

sequence of length N is directly permuted by an interleaver, whereas in RPTC, this information sequence is first repeated L times prior to permuting by an interleaver of size LN. A puncturer is cascaded in series with the second RSC encoder, so as to recover the loss in code rate due to the repeater. The output of the second RSC encoder of length LN is punctured to a sequence of length nN. Various code rates could be obtained depending on the values of the L and n parameters. For example, when $n = 1$, the overall code rate is $1/3$ similar to the overall code rate of TC.

RPTCs may be useful in generating low rate TCs from which codes with various rates may be designed with suitable puncturing [44]. This can be done by decreasing the number of bits/symbols punctured and/or increasing the repetition factor L. In [41, 45], construction of low rate TCs consists of decreasing the code rate of the component RSC encoders or increasing the number of parallel RSC encoders.

3.1.2 Repeat-Punctured Turbo Code Decoder

A block diagram of the repeat-punctured turbo code decoder structure is shown in Figure 3.3. The two constituent decoders are *soft-in soft-out* (SISO) similar to TC decoders.

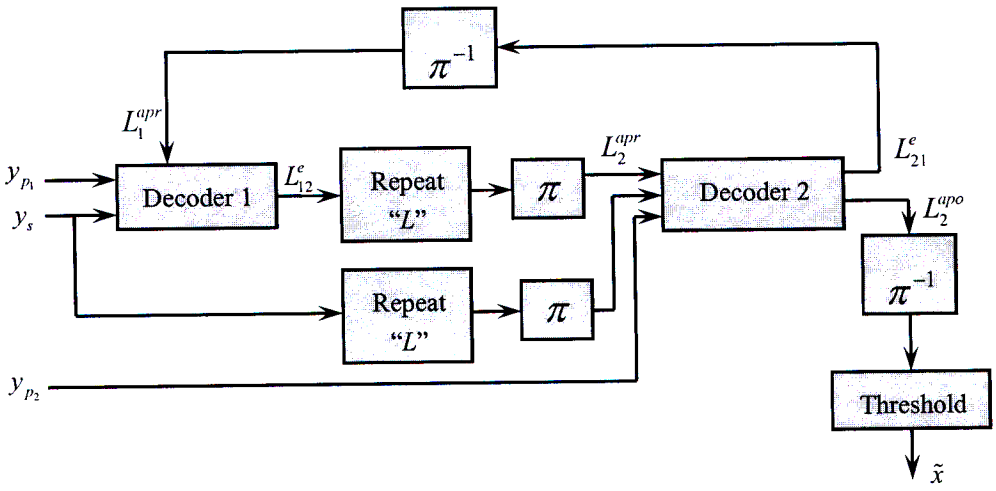

Figure 3.3: Structure of the RPTC decoder.

The two MAP decoders are linked by a repeater and an interleaver of length LN in a similar fashion to the encoder structure. This decoder operates iteratively as described in Chapter 2 for turbo code decoder, with the only difference that RPTC decoder combines repeaters and interleavers so as to match the structure of RPTC encoder.

3.2 Coded Cooperation with Repeat-Punctured Turbo Codes

3.2.1 System and Channel Models

The system model of RPTC in coded cooperation is similar to the one documented in [16, 19] and is repeated here. A communication system between two users and a destination is considered. This can be extended to a scenario with more than two users. The channels between the users (inter-user) and from the users to the destination (uplink) are independent of each other and under slow Rayleigh fading. Assuming that the inter-user channels are reciprocal and considering the case in which the destination or receiver maintains Channel State Information (CSI) and employs coherent detection (so that in the analysis, only the fading coefficient magnitudes are taken into account). The system uses BPSK modulation, with all users having the same transmit power. For BPSK modulation, the baseband-equivalent discrete-time signal transmitted by user $i \in \{1,2\}$ and received by user $j \in \{0,1,2\}$ with $j \neq i$, where $j = 0$ denotes the destination, is defined as

$$r_{i,j}(n) = \alpha_{i,j}(n)\sqrt{E_{b,i}}b_i(n) + z_j(n), \tag{3.1}$$

where $E_{b,i}$ is the transmitted energy per symbol for user i, $b_i(n) \in (-1,+1)$ is the BPSK modulated signal at time n, $\alpha_{i,j}(n)$ is the fading coefficients magnitude between users i and j, and $z_j(n)$ represent the effects of AWGN and other forms of interference, and is modeled as zero-mean, mutually independent, white Gaussian random variables with variance $\sigma^2 = N_0/2$ per dimension. The fading coefficients $\alpha_{i,j}(n)$ are modeled as independent sample of Rayleigh-distributed random variable characterized by mean-square value

$\Omega_{i,j} = E_{\alpha_{i,j}}[\alpha_{i,j}^2(n)]$, where the expectation operator with respect to random variable x is denoted by $E_x[.]$. For slow fading, the fading coefficients remain constant over the entire transmission of each source block, that is $\alpha_{i,j}(n) = \alpha_{i,j}$. In the case of reciprocal channels, $\alpha_{i,j}(n) = \alpha_{j,i}(n)$ for slow fading [5, 6, 11, and 14].

The instantaneous received SNR per bit for the channel between users i and j is defined as

$$\gamma_{i,j}(n) = \frac{\alpha_{i,j}^2(n) E_{b,i}}{N_0}. \tag{3.2}$$

For $\alpha_{i,j}(n)$ Rayleigh distributed, the instantaneous received SNR per bit $\gamma_{i,j}(n)$ has an exponential distribution with mean

$$\Gamma_{i,j} = E_{\alpha_{i,j}}[\gamma_{i,j}(n)] = \Omega_{i,j} \frac{E_{b,i}}{N_0}. \tag{3.3}$$

It is assumed that the channel statistics are not changing with time, that is, $\Omega_{i,j}$ and $\Gamma_{i,j}$ are constant over n for a given channel. The quality of the channel is quantified by its corresponding average received SNR obtained in (3.3). N_0 is the variance of the noise.

3.2.2 Implementation of Coded Cooperation with RPTC

The implementation of repeat-punctured turbo coded cooperation [74] is shown in Figure 3.4. The codeword of the first frame is obtained using the first RSC encoder. The user repeats the information sequence of length N (in our case $L = 2$, i.e., the information sequence is repeated twice) before being permuted by an interleaver of size $2N$, encodes it and punctures the RSC encoder output sequence to N, so as to maintain the code rate to $1/3$. This is done upon successful decoding of the partner. This scheme has a fixed cooperation percentage of 33%. The destination combines both the first (systematic + first parity bits) and second frames prior to repeat-punctured turbo decoding.

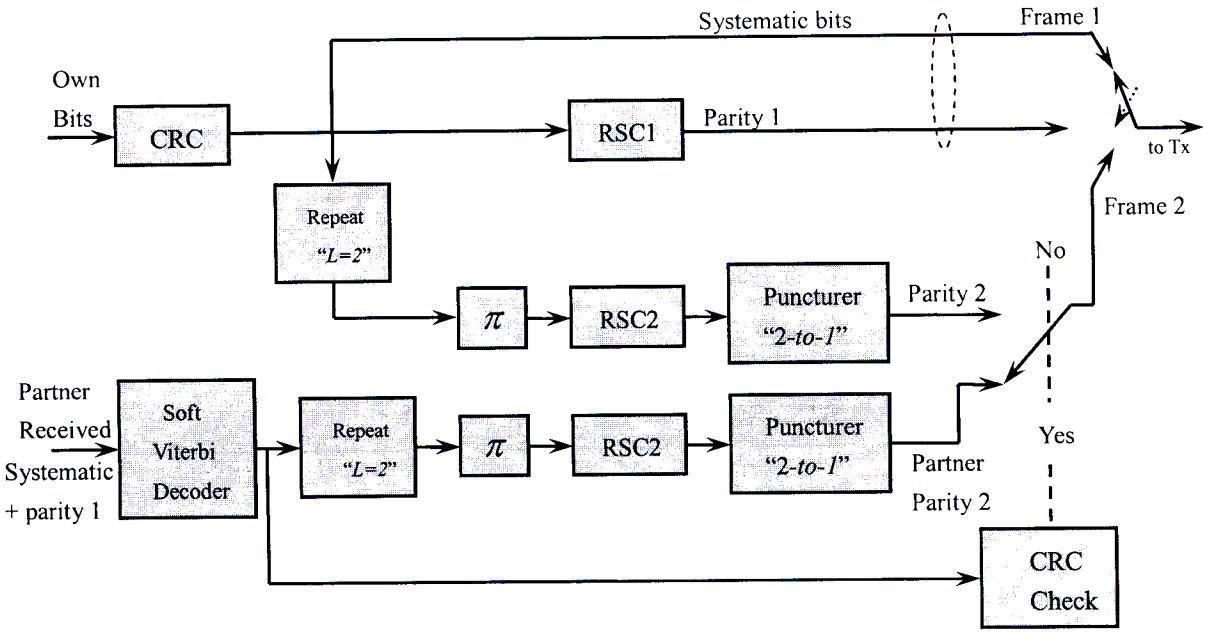

Figure 3.4: Repeat-punctured turbo encoding in a coded cooperation.

3.2.3 Simulation Results

We present the BER performance of repeat-punctured turbo coded cooperation under slow fading. The results were obtained via computer simulations. The RPTC used for simulations is an 8-state turbo code with generator polynomials $G=(1,15/13)_8$ and 1/2 rate RSC encoders. The source block code length is $K=128$. The overall code rate is 1/3. As mentioned before, the level of cooperation is 33% and it is assumed that the uplink channels are statistically similar, that is, the two channels experience the same fading process. Figure 3.5 compares the performance of turbo coded cooperation and coded cooperation using RPTCs under various qualities of the inter-user channel.

The improvement of coded cooperation using RPTCs for various qualities of the inter-user channel (6dB and 12dB) over the non-cooperative RPTC is dramatic. More importantly, we note that RPTC cooperation outperforms turbo coded cooperation by almost 2dB for higher uplink SNRs, both for 6dB and 12dB inter-user channels.

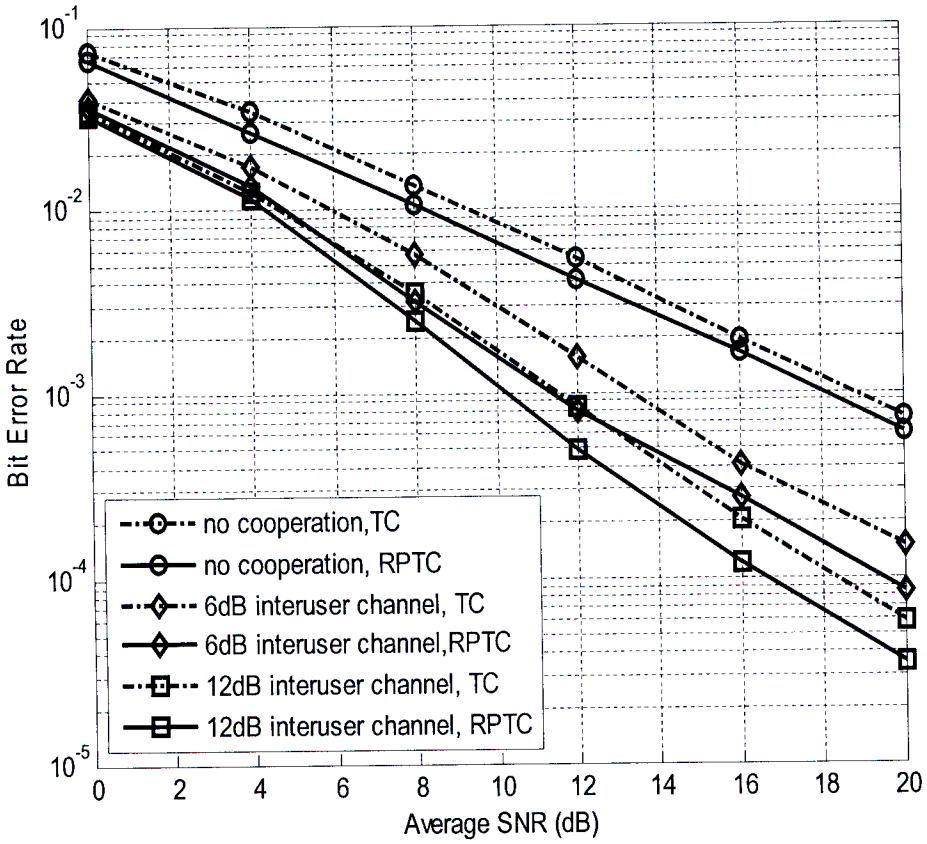

Figure 3.5: BER Comparison of turbo coded and repeat-punctured turbo coded cooperation in slow fading.

3.3 Dual Repeat-Punctured Turbo Coded Cooperation

Motivated by [44], the work in [75] proposed a dual Repeat-Punctured Turbo Codes (DRPTC) in coded cooperation. The system and channel models are identical to the ones described above and therefore will be omitted here. We only present the implementation of this code in coded cooperation and evaluate its BER performance in slow fading.

3.3.1 Implementation of Dual RPTCs in Coded Cooperation

The implementation of coded cooperation using DRPTCs is described here. The modified RPTC consists of two parallel concatenated RSC encoders. The encoding structure of this code is different from the conventional TC encoding and RPTC encoding. In the conventional TCs, the information sequence of length N is directly permuted by an interleaver, whereas in the case of DRPTC, this information sequence of length N is first repeated L times and then permuted by an interleaver of size LN similar to [44], but for both encoders. We use a repeater coefficient of $L=2$. The block diagrams of DRPTC encoding and decoding structures are shown in Figures 3.6 and 3.7. In Figure 3.7, the faded versions of the systematic bits, first and second parity sequence are denoted by y_s, y_{p1} and y_{p2} respectively.

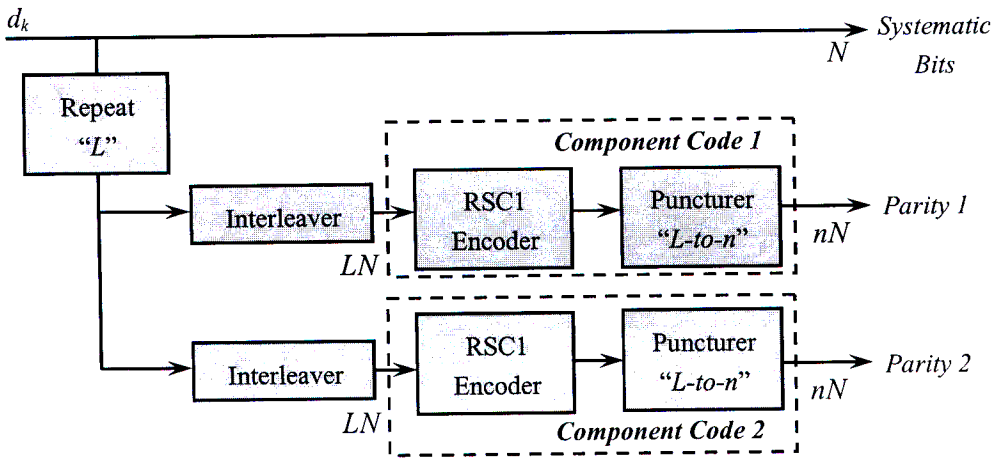

Figure 3.6: Dual RPTC encoding structure.

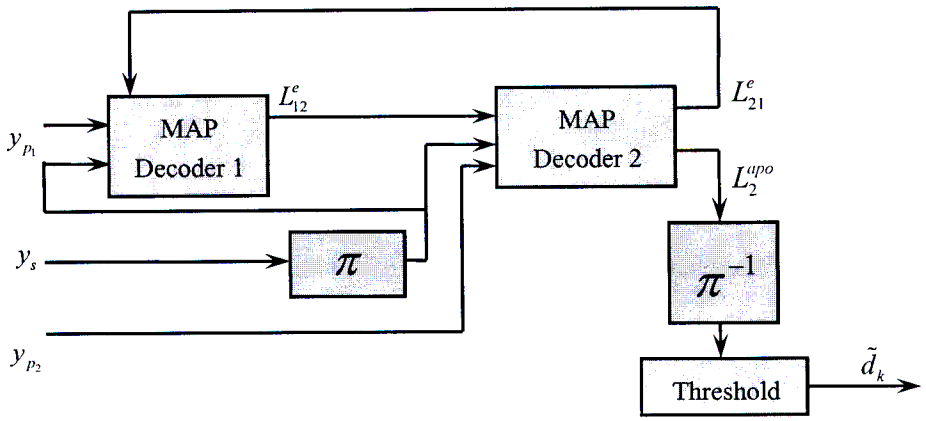

Figure 3.7: Structure of the DRPTC decoder.

The block diagram illustrating the implementation of coded cooperation dual repeat-punctured turbo codes is shown in Figure 3.8. The first parity sequence in the first frame is obtained via an RSC encoder as described in Figure 3.6. The user first repeats twice the source bits of length N prior to permuting through an interleaver π_1 of size $2N$, punctures to nN bits (where $n=1$), and transmits together with the systematic bits to the destination and the partner. The partner then receives the first frame and attempts a soft Viterbi decoding. The partner transmits the second parity sequence (obtained via repeating, interleaving π_2, encoding and puncturing) for the other user upon successful decoding. If the decoding is not a success, the other user will send its second parity sequence itself via interleaver π_3. At the destination, the combination of the first and second frames offers the possibility of DRPTC decoding.

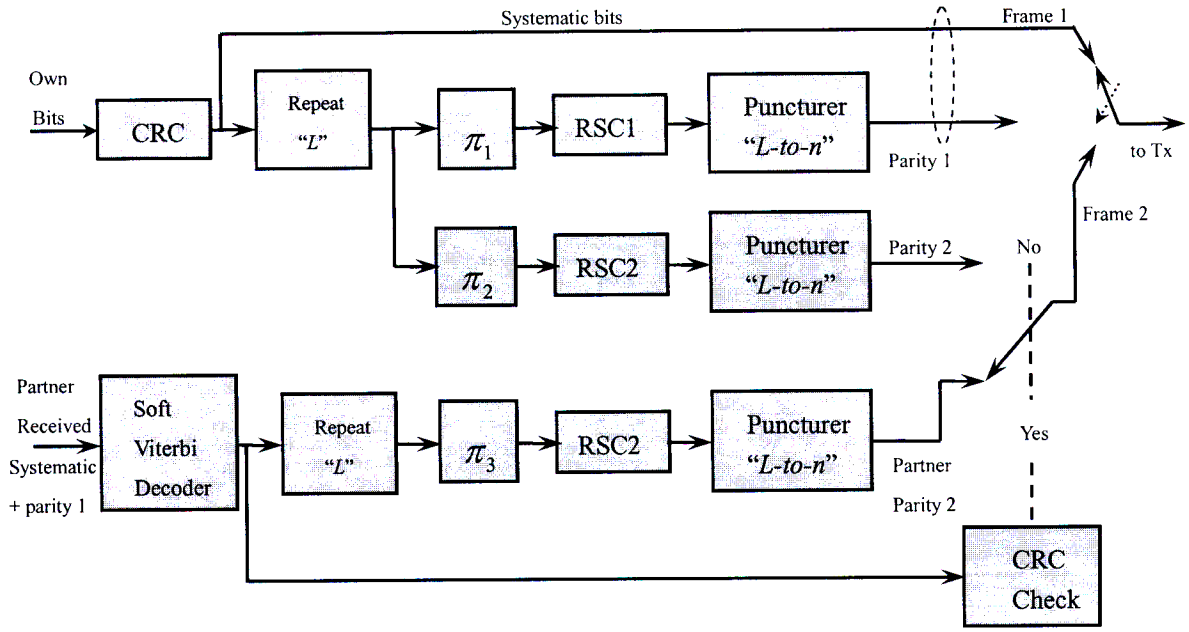

Figure 3.8: Dual RPTC encoding in coded cooperation.

3.3.2 Simulation Results

An 8-state dual repeat-punctured turbo code with generator polynomials $G=(1,15/13)_8$ and 1/2 rate RSC encoders is used. The source block code length is $K=128$. The overall code rate is 1/3. The level of cooperation is 33% and we assume that the uplink channels are statistically similar.

Figure 3.9 compares the BER performance of the turbo coded cooperation, repeat-punctured turbo coded cooperation and the dual repeat-punctured turbo coded cooperation in slow fading. The overall rate for this scheme is 1/3. For good clarity, we omit the performance of the 6dB inter-user SNR for these schemes here. We observe that DRPTC in coded cooperation performs better than the TC and RPTC in coded cooperation by almost 4dB and 2dB respectively for higher uplink SNR, under the 12dB inter-user channel. For the

non-cooperative mode, the new scheme outperforms the existing TC and DRPTC cooperation schemes by 1dB and 2dB respectively.

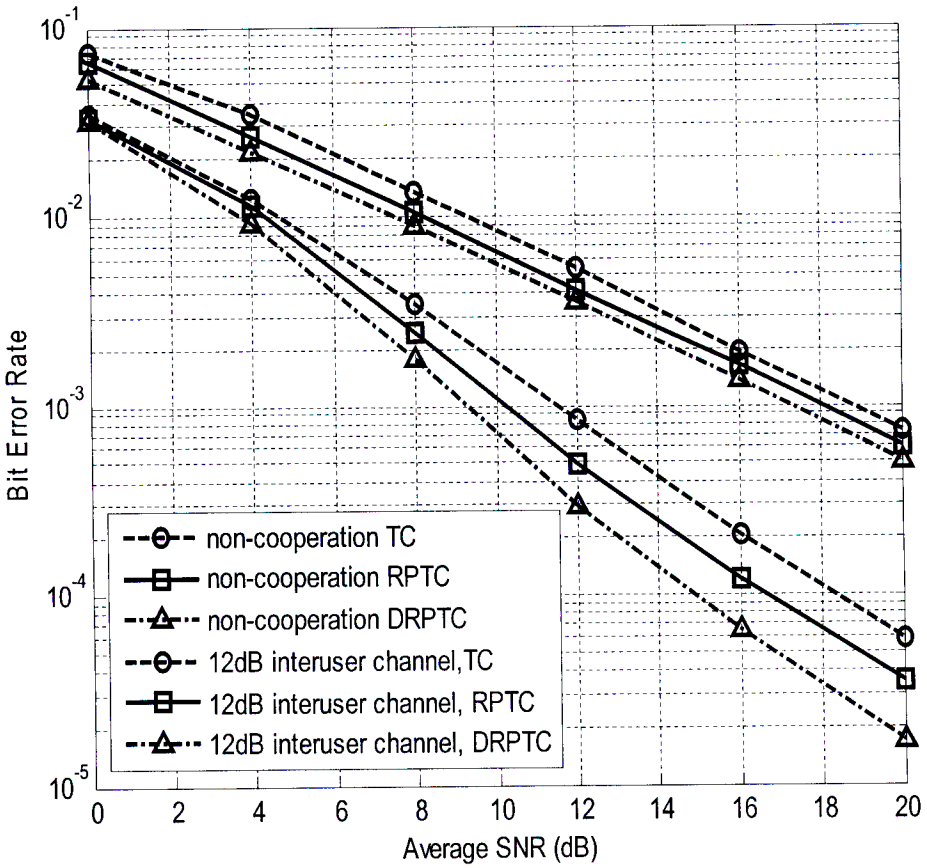

Figure 3.9: Comparison of turbo coded, repeat-punctured turbo coded and dual repeat-punctured turbo coded cooperation schemes under slow Rayleigh fading.

3.4 Conclusion

In this chapter, two new coded cooperation schemes using repeat-punctured turbo codes and dual repeat-punctured turbo codes are presented. Both schemes are presented separately in almost a similar manner. The system and channel models of these schemes are first introduced, followed by their codes implementation in coded cooperation framework. Finally, simulation results of both schemes are discussed. It is shown that, the dual RPTC cooperation framework achieves approximately 4dB and 2dB gains for a 12dB inter-user SNR at a BER of 6×10^{-5}, over TC and RTPC cooperation respectively. These gains represent significant improvements. However, these gains come at the cost of some processing delay introduced in both proposed schemes by the repetition and puncturing structure. DRPTCs outperforms regular TCs and RPTCs because they introduce more randomness due to the second interleaver, thus reducing the codewords of low weight.The proposed schemes are more complex than the conventional turbo coded cooperation due to the additional repetition-puncturing structure.

CHAPTER 4

PERFORMANCE ANALYSIS OF RPTC AND DRPTC COOPERATION

In this chapter, the theoretical bounds for bit error probability via numerical analysis for repeat-punctured turbo coded and dual repeat-punctured turbo coded cooperation over slow fading channels are presented. This is done by first deriving the pairwise error probabilities using the tools and techniques developed in [46] and [47] in Section 4.1. In Section 4.2, the weight enumerators are derived using the transfer bounding techniques [48] to obtain bounds on bit error rate. Finally, the upper union bounds for the end-to-end bit error probability are determined using the techniques developed by Malkamaki and Leib [49]. Comparisons between theoretical bounds and computer simulations of TC, RPTC and DRPTC cooperation show that the analytical bounds obtained are accurate.

4.1 Pairwise Error Probability

In this section the pairwise error probability (PEP) as exactly developed in [15, 16, 18, and 19] is presented. The PEP for a coded system is the probability of detecting an erroneous codeword $\boldsymbol{e}$ when codeword $\mathbf{c}$ was transmitted. The PEP is written as [47, (12.13)],

$$P(\mathbf{c} \to \mathbf{e} \mid \gamma) = Q\left(\sqrt{2\sum_{n\in\eta}\gamma(n)}\right), \tag{4.1}$$

where $Q(x)$ denotes the Gaussian Q -function [50, (2-1-97)] and $\gamma(n)$ is the instantaneous received SNR per bit for code bit n. The set η is defined as the set of all n for which $c(n) \neq e(n)$, hence the cardinality (the number of elements in a set) of η, $|\eta| = d$, where d is the Hamming distance between $\mathbf{c}$ and $\boldsymbol{e}$. Assumption of a linear code is made, where the transmitted codeword $\mathbf{c}$ is always chosen as the all-zero codeword only for error analysis purposes. Therefore, the PEP does not depend on the particular codewords $\mathbf{c}$ and $\boldsymbol{e}$, but only on the Hamming distance d. In this case, the conditional PEP will denoted by $P(d \mid \gamma)$.

It is assumed that the uplink channel coefficients remain constant over the codeword, that is, $\alpha_{i,0}(n) = \alpha_{i,0}$ and $\gamma_{i,0}(n) = \gamma_{i,0}$ are constant for $n = 1,...,N$ user i's uplink channel for quasi-static fading. For two-user cooperation, four cases arise as shown in Figure 4.1 [15, 19].

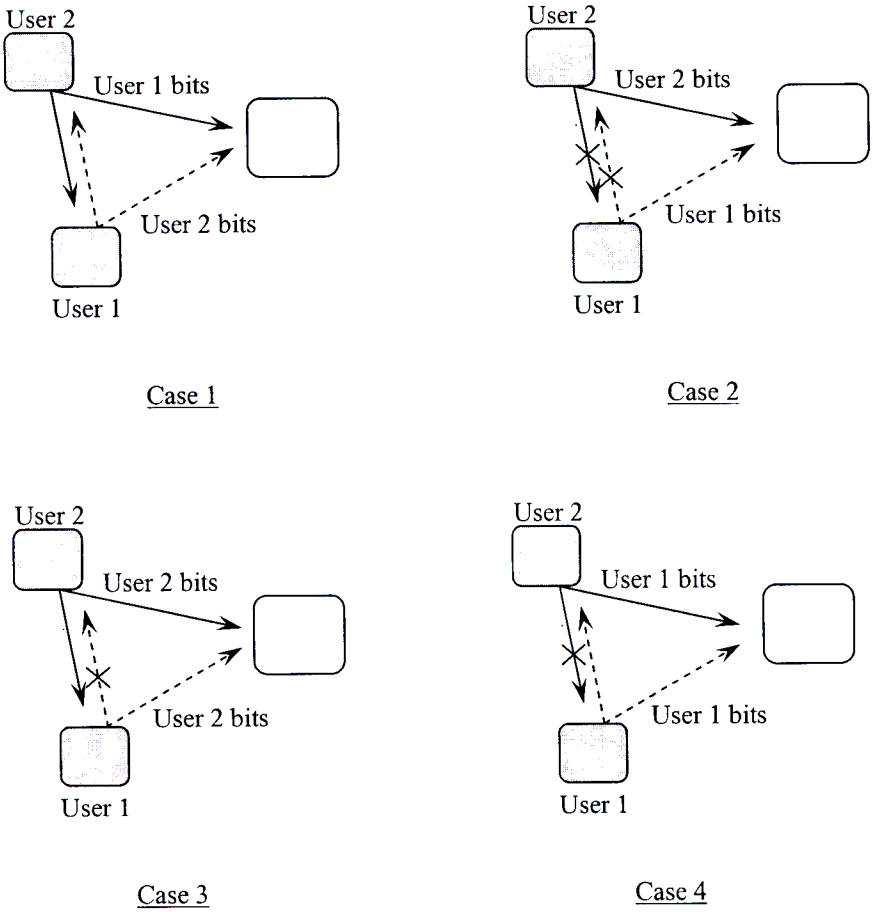

Figure 4.1: Four cooperative cases for a two-user cooperation.

4.1.1 Case 1: Both Users Successfully Decode Each Other

For the full cooperation case or equivalently Case 1, when both users successfully decode each other data in the first frame, each user's coded bits are segmented between the two user channels. User's 1 codeword is considered here, so (4.1) can be written as

$$P(d \mid \gamma_{1,0}, \gamma_{2,0}) = Q\left(\sqrt{2d_1\gamma_{1,0} + 2d_2\gamma_{2,0}}\right) \tag{4.2}$$

where d_1 and d_2 represent the portions of the error event bits transmitted via user 1's and user 2's channel respectively, such that $d = d_1 + d_2$. It should be noted that d_1 and d_2 are

independent of $\gamma_{1,0}$ and $\gamma_{2,0}$.

The unconditional PEP can be obtained by averaging (4.2) over fading distributions as

$$P(d)=\int_0^{\infty}\int_0^{\infty}P(d\,|\,\gamma_{1,0},\gamma_{2,0})p(\gamma_{1,0})p(\gamma_{2,0})d\gamma_{1,0}d\gamma_{2,0}, \tag{4.3}$$

where $p(x)$ is the probability density function of random variable x. An exact solution to (4.3) can be obtained using the techniques developed by Simon and Alouini [47]. Using the following alternative representation for the Gaussian Q-function [46], and then applying to performance analysis in fading channels in [47] will lead to the exact solution to (4.3)

$$Q(x)=\frac{1}{\pi}\int_0^{\pi/2}\exp\left(-\frac{x^2}{2\sin^2\theta}\right)d\theta, \tag{4.4}$$

Using (4.4) in (4.2) and (4.3) yields

$$P(d)=\frac{1}{\pi}\int_0^{\pi/2}\left[\int_0^{\infty}\exp\left(-\frac{d_1\gamma_{1,0}}{\sin^2\theta}\right)p(\gamma_{1,0})d\gamma_{1,0}\right]\left[\int_0^{\infty}\exp\left(-\frac{d_2\gamma_{2,0}}{\sin^2\theta}\right)p(\gamma_{2,0})d\gamma_{2,0}\right]d\theta. \tag{4.5}$$

The two inner integrals in (4.5) are similar to the moment- generating functions for the two densities $p(\gamma_{1,0})$ and $p(\gamma_{2,0})$ [51, (3.5-1)-(3.5-3)],

$$M_x(s)=\int_0^{\infty}e^{sx}p(x)dx, \tag{4.6}$$

where $M_x(s)$ is the moment-generating function of random variable x. Thus (4.5) can be written as

$$P(d)=\frac{1}{\pi}\int_0^{\pi/2}M_{\gamma_{1,0}}\left(-\frac{d_1}{\sin^2\theta}\right)M_{\gamma_{2,0}}\left(-\frac{d_2}{\sin^2\theta}\right)d\theta. \tag{4.7}$$

The moment-generating function is identical to the Laplace transform, the only difference being a change of sign in the exponent. Therefore, various techniques for evaluating moment-generating functions and Laplace transforms can be used to find the solution to the integrals of this form. The moment-generating function for the instantaneous SNR γ is [52, (17)] in the case of Rayleigh fading

$$M_{\gamma}(-s)=\frac{1}{1+s\Gamma}, \qquad s>0. \tag{4.8}$$

Using (4.8) in (4.7) results in

$$P(d)=\frac{1}{\pi}\int_0^{\pi/2}\left(1+\frac{d_1\Gamma_{1,0}}{\sin^2\theta}\right)^{-1}\left(1+\frac{d_2\Gamma_{2,0}}{\sin^2\theta}\right)^{-1}d\theta, \tag{4.9}$$

where $\Gamma_{i,0}$ is the mean exponential distribution defined in (3.3).

An exact expression for the unconditional PEP is given by (4.9) and is well evaluated using numerical integration techniques. Furthermore, using [47, (5A.58)-(5A.60)], a closed-expression can be obtained for (4.9). The resulting closed-expression does not give extra information insight into coded cooperation.

The following upper bound from (4.9) can be obtained. This is done by remarking that the integrand is maximized for $\sin^2\theta=1$, so that

$$P(d)\le\frac{1}{2}\left(\frac{1}{1+d_1\Gamma_{1,0}}\right)\left(\frac{1}{1+d_2\Gamma_{2,0}}\right) \tag{4.10}$$

For large SNR values, the PEP is inversely proportional to the product of the average SNR of the uplink channels, i.e.

$$P(d)\le\frac{1}{2}\left(\frac{1}{d_1d_2\Gamma_{1,0}\Gamma_{2,0}}\right). \tag{4.11}$$

Full diversity of order of two is achieved for this case (Case 1), if both d_1 and d_2 are non-zero. At high SNRs, this case also achieves full diversity.

4.1.2 Case 2: Neither User Successfully Decodes Each Other

For Case 2 or non-cooperative transmission, all the code bits for either user 1 or user 2 are transmitted through the same channel, i.e. $d_1=d$ and $d_2=0$. In slow fading, the conditional PEP is obtained from (4.2) to give

$$P(d\,|\,\gamma)=Q\left(\sqrt{2d\gamma}\right), \tag{4.12}$$

and the unconditional PEP becomes

$$P(d) = \frac{1}{\pi}\int_0^{\pi/2}\left(1+\frac{d\Gamma}{\sin^2\theta}\right)^{-1} d\theta$$

$$\leq \frac{1}{2}\left(\frac{1}{1+d\Gamma}\right). \tag{4.13}$$

Eventually, this case will not provide diversity.

4.1.3 Case 3: Only One User Successfully Decodes the Other User

For Case 3, where user 2 does not successfully decode user 1, but user 1 successfully decodes user 2, the same additional parity bits are sent by both users for user 2 in the second time interval. These bits are combined at the destination to give the following conditional PEP obtained from (4.2)

$$P(d \mid \gamma_{1,0}, \gamma_{2,0}) = Q\left(\sqrt{2d_1\gamma_{1,0} + 2d_2(\gamma_{1,0}+\gamma_{2,0})}\right)$$

$$= Q\left(\sqrt{2d_1\gamma_{1,0} + 2d\gamma_{2,0}}\right) \tag{4.14}$$

and the unconditional PEP obtained by averaging (4.14) over the fading distributions is

$$P(d) = \frac{1}{\pi}\int_0^{\pi/2}\left(1+\frac{d_1\Gamma_{1,0}}{\sin^2\theta}\right)^{-1}\left(1+\frac{d\Gamma_{2,0}}{\sin^2\theta}\right)^{-1} d\theta$$

$$\leq \frac{1}{2}\left(\frac{1}{1+d_1\Gamma_{1,0}}\right)\left(\frac{1}{1+d\Gamma_{2,0}}\right). \tag{4.15}$$

At high SNRs, this case also achieves full diversity for user 2. The last possible case is when user 1 does not successfully decode user 2, but user 2 does decode user 1. This is similar to case 3, but with the roles reversed. In this case, user 1 achieves full diversity at high SNRs.

4.2 Weight Distribution

The weight distribution or weight enumerating function $A(d)$ is essential in deriving the union bounds of a given code. In (4.11) and (4.15), the PEP is a function of the partitions of the Hamming distance, d_1 and d_2. For turbo codes, obtaining the weight distribution is

exceedingly complicated for a given interleaver, since the redundant bits generated by the second encoder will depend on the weight of the input word and more importantly on how the bits have been interleaved. Theoretically speaking, an exhaustive enumeration over all possible cases is the best solution. But this turns out to be impractical for long interleavers. An average weight distribution was proposed in [27, 48, 68], where the average is over all interleavers of a given length. Turbo codes can be viewed as the concatenation of multiple code fragments x_k^s, x_{1k}^p and x_{2k}^p as shown in Figure 4.2. Figure 4.2 shows that the code fragments x_{1k}^p and x_{2k}^p generated by the constituent encoders depend on the input weight and on how the bits within the fragments have been interleaved by I_1 and I_2 respectively. The derivation of the weight enumerating function is shown here as presented in [53].

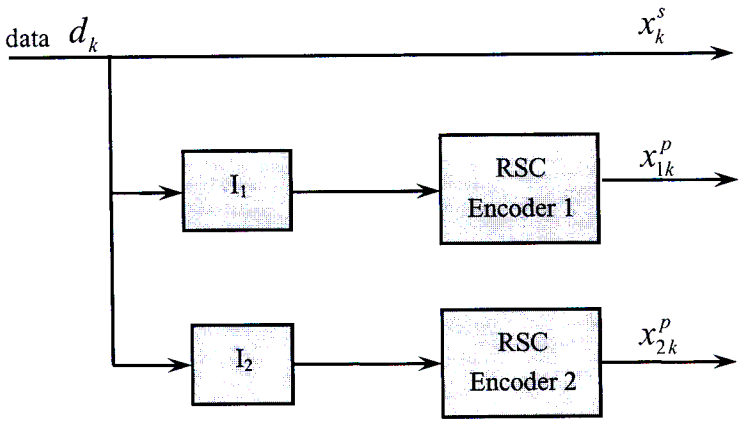

Figure 4.2: Concatenation of multiple code fragments.

The union upper bound for the BER for ML decoding of an (n,k) block code over a memoryless channel is given as shown in (4.16) provided that the interleavers I_1 and I_2 are known.

$$P_{word}(I_1,I_2) \leq \sum_{d=d_{\min}}^{n} A(d/I_1,I_2)P(d) \tag{4.16}$$

where $A(d/I_1,I_2)$ is the number of codewords of Hamming weight d and $P(d)$ is the PEP having weight d. For a given pair of interleavers (I_1,I_2), $A(d/I_1,I_2)$ is constructed

by encoding all valid input sequences and determining its output weight. The word error probability averaged over all interleaver pairs is

$$\begin{aligned}\bar{P}_{word} &= \sum_{I_1}\sum_{I_2} P(I_1,I_2)P_{word}(I_1,I_2) \\ &= \sum_{I_1}\sum_{I_2} P(I_1,I_2)\sum_{d=d_{\min}}^{n} A(d/I_1,I_2)P(d) \\ &= \sum_{d=d_{\min}}^{n} P(d)\sum_{I_1}\sum_{I_2} P(I_1,I_2)A(d/I_1,I_2) \\ &= \sum_{d=d_{\min}}^{n} P(d)\bar{A}(d),\end{aligned} \tag{4.17}$$

where $P(I_1,I_2)$ is the probability of some given interleavers and $\bar{A}(d)$ is the average weight enumerating function. $\bar{A}(d)$ is formed by averaging over all interleaver pairs. Assuming that the interleaver pairs are uniform and independent from all possible permutations of N elements, the average enumerating distribution can be written as

$$\bar{A}(d) = \sum_{I_1}\sum_{I_2} P(I_1)P(I_2)A(d/I_1,I_2). \tag{4.18}$$

In addition, $A(d/I_1,I_2)$ can be written as

$$A(d/I_1,I_2) = \sum_{d}\binom{N}{i}P(d/i,I_1,I_2), \tag{4.19}$$

where $P(d/i,I_1,I_2)$ is the conditional probability obtained from an input of weight i and interleaver pair (I_1,I_2) to produce a codeword of weight d. Moreover, because turbo codes are a parallel concatenation of three code fragments, the weight of the codeword can be calculated as the sum of the weights of the three code fragments, i.e. $d = i + d_1 + d_2$, where d_1 and d_2 represent the Hamming weights of the input and first parity sequences x_{1k}^{p} and x_{2k}^{p} respectively. Using these information, (4.19) can be rewritten as

$$A(d/I_1,I_2) = \underbrace{\sum_{i}\sum_{d_1}\sum_{d_2}}_{d=i+d_1+d_2}\binom{N}{i}P(i,d_1,d_2/i,I_1,I_2). \tag{4.20}$$

Since the interleavers are assumed to be independent, $P(i,d_1,d_2/i,I_1,I_2)$ can be rewritten as $P(i/i)P(d_1/i,I_1)P(d_2/i,I_2)$ which substituting in (4.18) gives

$$\bar{A}(d)=\sum_{I_1}\sum_{I_2}P(I_1)P(I_2)\underbrace{\sum_{i}\sum_{d_1}\sum_{d_2}}_{d=i+d_1+d_2}\binom{N}{i}P(i/i)P(d_1/i,I_1)P(d_2/i,I_2)$$

$$=\underbrace{\sum_{i}\sum_{d_1}\sum_{d_2}}_{d=i+d_1+d_2}\binom{N}{i}\sum_{I_1}P(I_1)P(d_1/i,I_1)\sum_{I_2}P(I_2)P(d_2/i,I_2)$$

$$=\underbrace{\sum_{i}\sum_{d_1}\sum_{d_2}}_{d=i+d_1+d_2}\binom{N}{i}P(d_1/i)P(d_2/i) \tag{4.21}$$

where $P(d_1/i)$ and $P(d_2/i)$ are obtained using Bayes' Theorem and denote the conditional probability of producing a codeword fragment of weight d given an input sequence of weight i randomly selected.

In the sequel, the derivation of the conditional probabilities using transfer function bounding techniques as described in [48] is shown. First of all, the state diagram of a $(1,15/13)_8$ code fragment as illustrated in Figure 4.3 is described.

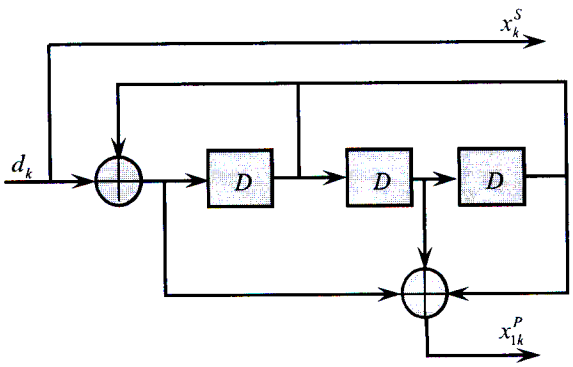

Figure 4.3: RSC $(1, 15/13)_8$ encoder.

The state transition diagram and the state transition table for the $(1,15/13)_8$ code fragment are shown in Table 4.1 and Figure 4.4 below. The transition between states is labeled by the input information bit and the corresponding output bits (the first bit represents the encoded bit and the second one is the systematic bit). In the state transition table, S_i^n where $i = \{0,1,2\}$ represent the current state of the encoder and S_i^{n+1} represent the next state of the encoder. O_1 represents the encoded output bit and O_2 represents the systematic bit.

Table 4. 1: State transition table of the $(13, 15)_8$ code fragment.

S_0^n	S_1^n	S_2^n	*Input*	S_0^{n+1}	S_1^{n+1}	S_2^{n+1}	O_1	O_2
0	0	0	0	0	0	0	0	0
			1	0	0	1	1	1
0	0	1	0	0	1	1	1	0
			1	0	1	0	0	1
0	1	0	0	1	0	0	1	0
			1	1	0	1	0	1
0	1	1	0	1	1	1	0	0
			1	1	1	0	1	1
1	0	0	0	0	0	1	0	0
			1	0	0	0	1	1
1	0	1	0	0	1	0	1	0
			1	0	1	1	0	1
1	1	0	0	1	0	1	1	0
			1	1	0	0	0	1
1	1	1	0	1	1	0	0	0
			1	1	1	1	1	1

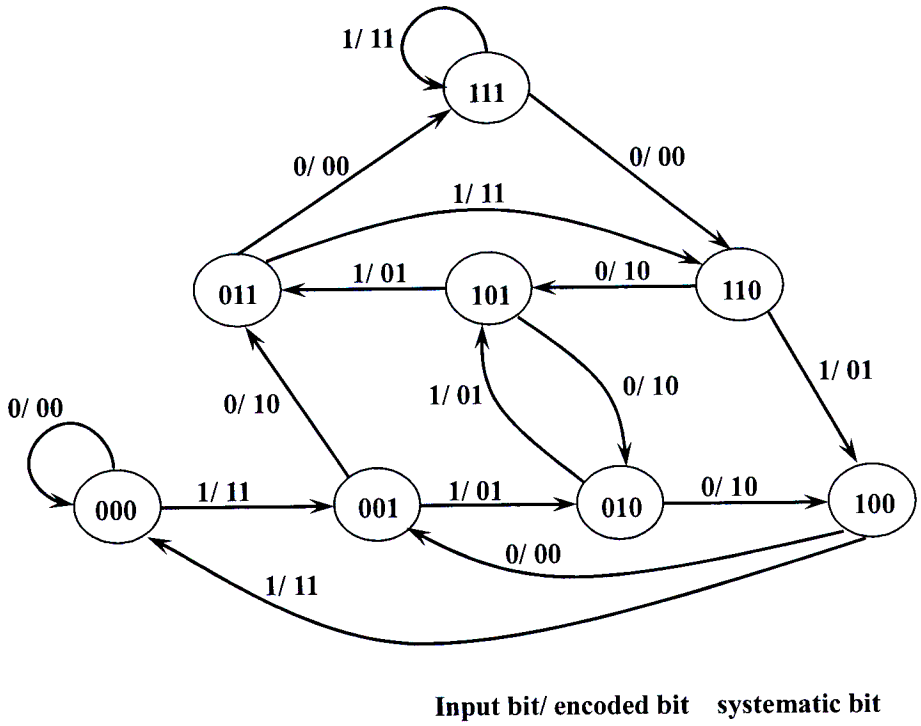

Figure 4.4: State transition diagram of $(13, 15)_8$ code fragment.

Each edge label as shown in Figure 4.4 can conveniently be substituted with a monomial $L^l I^i D^d$ which describes the input/output relationship for a particular transition, where l is always equal to 1 for a rate $1/m$, and i and d depending on whether the corresponding input and output bits are 0 or 1, can take the values 0 or 1 respectively.

For example:

State 000 → State 000, input/encoded output is 0/0, $L^l I^i D^d = LI^0 D^0 = L$.

State 000 → State 001, input/encoded output is 1/1, $L^l I^i D^d = LI^1 D^1 = LID$.

State 000 → State 010, no connection, $L^l I^i D^d = 0$.

State 000 → State 011, no connection, $L^l I^i D^d = 0$.

A state transition matrix $A(L, I, D)$ shown below is obtained from the information state transition diagram in Figure 4.4

$$A(L,I,D)=\begin{pmatrix} L & LID & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & LI & LD & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & LD & LI & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & LID & L \\ LID & L & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & LD & LI & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & LI & LD & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & L & LID \end{pmatrix}. \tag{4.22}$$

The rows represent the previous states of the encoder, while the columns represent the next states of the encoder.

For a given code fragment, defined by a state diagram with 2^m states, $t(l,i,d)$ represent the number of paths of length l, input weight i, and output weight d, starting and ending in the zero state, with the condition that the trellis is terminated by m bits and these termination bits do not have length nor input weight [48]. The corresponding generating function or transfer function is then defined as

$$T(L,I,D)=\sum_{l\geq 0}\sum_{i\geq 0}\sum_{d\geq 0}L^l I^i D^d t(l,i,d). \tag{4.23}$$

As reported in [54, (4.7)], $T(L,I,D)$ is the $(0^m,0^m)$ entry in the matrix

$$(I+A(L,I,D)+A(L,I,D)^2+A(L,I,D)^3+\ldots)A(1,1,D)^m. \tag{4.24}$$

The factor $A(1,1,D)^m$ does care for the termination bits in the trellis. Since $I+A+A^2+A^3+\ldots=(I-A)^{-1}$, we obtain from (4.24)

$$T(L,I,D)=\left[(I-A(L,I,D))^{-1}.A(1,1,D)^m\right]_{0^m,0^m} \tag{4.25}$$

An approximate expression of (4.25) can be obtained by omitting the termination factor $A(1,1,D)^m$, so as to get the transfer function

$$T(L,I,D)=\left[(I-A(L,I,D))^{-1}\right]_{0^m,0^m}. \tag{4.26}$$

Two methods can be used to determine the $\left[I-A(L,I,D))^{-1}\right]_{0^m,0^m}$. The first method described in

[55] is described in Appendix A. The other method uses the symbol operation in MATLAB and yields the same transfer function as found in Appendix A. The transfer function for $(15/13)_8$ code fragment is determined in Appendix A using one of the two methods given above. The other method is omitted here In Appendix A, the term *tf_0_0* represents the transfer function $T_{15/13}(L,I,D)$.

By multiplying both sides of (B.1) in appendix B by the denominator of the right-hand side, and taking the coefficient $t(l,i,d)$ of both sides of the resulting equation, the following recursion formula is obtained

$$\begin{aligned} t(l,i,d) = {} & t(l-1,i,d)+t(l-1,i-1,d-1)-t(l-6,i-5,d-5) \\ & -t(l-6,i-1,d-1)+t(l-6,i-5,d-1)+t(l-6,i-1,d-5) \\ & -t(l-7,i-5,d-1)+t(l-7,i-7,d-3)-2t(l-7,i-5,d-5) \\ & -t(l-7,i-1,d-5)+t(l-7,i-3,d-7)-t(l-7,i-6,d-2) \\ & -t(l-7,i-2,d-6)+2t(l-7,i-3,d-3)-2t(l-7,i-2,d-2) \\ & +t(l-7,i,d-4)+t(l-7,i-4,d)+2t(l-7,i-4,d-4) \\ & -t(l-8,i-8,d-4)-t(l-8,i-4,d-8)-t(l-8,i-4,d) \\ & +2t(l-8,i-6,d-6)+2t(l-8,i-2,d-6)+2t(l-8,i-6,d-2) \\ & +2t(l-8,i-2,d-2)-t(l-8,i,d-4)-4t(l-8,i-4,d-4) \\ & +\delta(l,i,d)-\delta(l-1,i-1,d-1)-\delta(l-2,i-1,d-1)-\delta(l-3,i-1,d-1) \end{aligned} \tag{4.27}$$

where $\delta(l,i,d)=1$ if $l=i=d=0$ and $\delta(l,i,d)=0$ otherwise.

$P(d/i)$ can be computed as follows

$$P(d/i)=\frac{t(N,i,d)}{\sum_{d'} t(N,i,d')}=\frac{t(N,i,d)}{\binom{N}{i}}. \tag{4.28}$$

Figure 4.5 shows the plot of the conditional probability $P(d/i)$ for the $(15/13)_8$ code fragment, for block length $N=128$ and a binomial probability distribution for 128 trials with probability 1/2, for reference. We note that the code fragment $(15/13)_8$ only admits for even input weights i even output weights d and admits for odd input weights i odd output

weights d. We also observe that the conditional probability $P(d/i)$ tends to the reference binomial distribution for moderate to high values of the input weight i, whereas for low values of the input weight i, $P(d/i)$ has more or less a skewed distribution. The fact that $P(d/i)$ gets closer to the reference binomial distribution for large values of i, is an indication of the random distribution of output weights d.

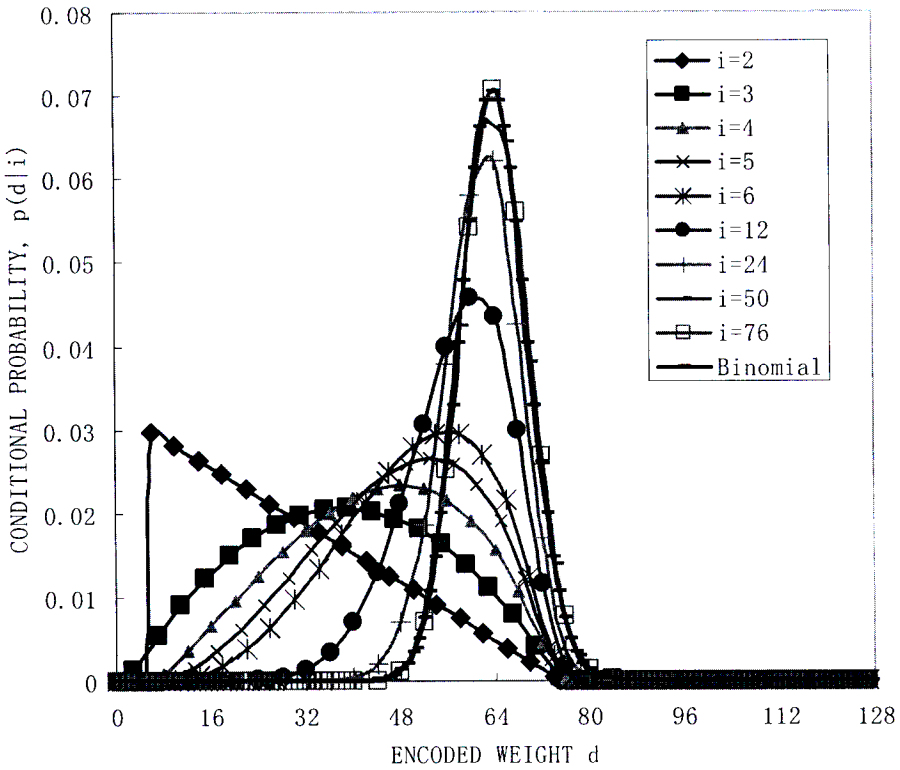

Figure 4.5: Conditional probability distribution *p(d|i)* for output weight given input weight, with block length *N=128*.

4.3 Bit Error Rate Analysis

The PEP and weight distribution obtained in the previous sections will be used in the computation of the end-to-end bit error probabilities. The following BER analysis has been

derived in [15]-[19] and is repeated here. Firstly, the calculation of the probability of operating in the cooperative mode, which can be parameterized with $\Psi = \{1,2,3,4\}$, where $\Psi = 1$ corresponds to Case 1 described above (full cooperation case), $\Psi = 2$ is the no-cooperation case, $\Psi = 3$ and $\Psi = 4$ correspond to Case 3 and Case 4 respectively, is imperative. Reciprocal inter-user channels are assumed.

For the full cooperation case, the conditional inter-user block error probability is given by

$$\begin{aligned} P(\Psi = 1 \mid \gamma) &= (1 - P_{block}^{(1)}(\gamma))(1 - P_{block}^{(2)}(\gamma)) \\ &\geq (1 - P_E^{(1)}(\gamma))^B (1 - P_E^{(2)}(\gamma))^B \\ &\geq (1 - BP_E^{(1)}(\gamma))(1 - BP_E^{(2)}(\gamma)), \end{aligned} \tag{4.29}$$

where $P_{block}(\gamma)$ is the block error rate probability bounded by [49, (12)], [56, (11)]

$$\begin{aligned} P_{block}(\gamma) &\leq 1 - (1 - P_E(\gamma))^B \\ &\leq B \cdot P_E(\gamma) \end{aligned} \tag{4.30}$$

In Equation (4.30), B is the number of trellis branches in the codeword and $P_E(\gamma)$ is the error even probability conditioned on γ. For a rate $1/n$ convolutional code, B equals the uncoded block length N. $P_E(\gamma)$ is bounded by [57, (4.4.8)]

$$P_E(\gamma) \leq \sum_{d=d_{\min}}^{\infty} a(d) P(d \mid \gamma), \tag{4.31}$$

where $d_{\min}$ is the minimum Hamming distance of the code and $a(d)$ is the weight enumerating function or the number of error events with Hamming distance d.

The conditional inter-user block error probability for Cases 2, 3 and 4 are given below

$$\begin{aligned} P(\Psi = 2 \mid \gamma) &= P_{block}^{(1)}(\gamma) P_{block}^{(2)}(\gamma) \\ &\geq BP_E^{(1)}(\gamma).BP_E^{(2)}(\gamma) \\ &\geq B^2 P_E^{(1)}(\gamma) P_E^{(2)}(\gamma). \end{aligned} \tag{4.32}$$

$$P(\Psi=3\mid\gamma)=(1-P_{block}^{(1)}(\gamma))P_{block}^{(2)}(\gamma)$$
$$\geq(1-P_{E}^{(1)}(\gamma))^{B}.BP_{E}^{(2)}(\gamma)$$
$$\geq B(1-BP_{E}^{(1)}(\gamma))P_{E}^{(2)}(\gamma). \quad (4.33)$$

$$P(\Psi=4\mid\gamma)=P_{block}^{(1)}(\gamma)(1-P_{block}^{(2)}(\gamma))$$
$$\geq BP_{E}^{(1)}(\gamma).(1-P_{E}^{(2)}(\gamma))^{B}$$
$$\geq BP_{E}^{(1)}(\gamma).(1-P_{E}^{(2)}(\gamma))^{B}. \quad (4.34)$$

The end-to-end bit error rate probabilities are computed using the unconditional inter-user block error probability $P(\Psi)$, which is given by

$$P(\Psi)=\int_{\gamma}P(\Psi\mid\gamma)p(\gamma)d\gamma. \quad (4.35)$$

Using (4.30) yields loose bounds in the case of slow fading as presented in [49]. The limit-before-average technique [49] can be used in (4.30) to give tight bounds on the block error probability. We only discuss the full cooperation (Case 1) here, as it is similar for the remaining 3 cases. Applying the limit-before-averaging techniques in (4.29), the tight upper bound obtained is

$$P(\Psi=1)\leq\int_{\gamma}\left(1-\min\left[1,\sum a_1(d)P(d\mid\gamma)\right]\right)^{B}$$
$$\times\left(1-\min\left[1,\sum a_2(d)P(d\mid\gamma)\right]\right)^{B}p(\gamma)d\gamma. \quad (4.36)$$

The overall end-to-end BER is obtained by averaging the unconditional BER over the four cooperative cases,

$$P_b=\sum_{i=1}^{4}P_b(\Psi)P(\Psi=i), \quad (4.37)$$

where $P_b(\Psi)$ is the BER. The conditional BER is bounded by [57, (4.4.8)]

$$P_b(\gamma,\Psi)\leq\sum_{d=d_{\min}}^{\infty}a(d)P(d\mid\gamma), \quad (4.38)$$

where $a(d)$ is given by

$$a(d) = \underbrace{\sum_{i}\sum_{d_1}\sum_{d_2}}_{d=i+d_1+d_2} \frac{i}{N}\binom{N}{i} P(d_1/i)P(d_2/i). \tag{4.39}$$

The unconditional BER is obtained by averaging (4.38). As mentioned above, tight bounds on the unconditional BER are obtained using the limit-before-average method [49]. The unconditional BER is thus given by

$$P_b(\Psi) = \int_{\gamma} \min\left[\frac{1}{2}, P_b(\gamma,\Psi)\right] p(\gamma)d\gamma. \tag{4.40}$$

In (4.39) the conditional probability for the both encoders for the DRPTC is given by

$$P(d/i) = \frac{t(LN,Li,d)}{\sum_{d'} t(LN,Li,d')} = \frac{t(LN,Li,d)}{\binom{LN}{Li}}, \tag{4.41}$$

where $L=2$ is the repeater coefficient. In RPTC, only the conditional probability of the second encoder is similar to (4.41) , whereas the first encoder conditional probability is identical to (4.28).

4.4 Numerical Results

In this part of the dissertation, the plots of the upper bounds on the unconditional BER P_b using the limit-before-average method [49] and the simple average technique are compared. The probabilities of occurrence of the four cooperative cases under three different inter-user channels SNR are then presented. The BER theoretical bounds for RPTC and DRPTC implementations of coded cooperation with the simulation results are compared.

The convergence of the union bound on the bit error probability of turbo coded cooperation (non-cooperation case) using the simple average method and the one obtained from the limit-before-average technique is shown in Figure 4.6. An observation is made that the

limit-before-average union bound on the bit error probability converges quite fast, whereas the union bound based on the simple average method calculates as the sum of the average PEPs does not. The summation in (4.38) where the upper limit corresponds to $d_{\max}$ is truncated such that only the error events having the total distance $d = d_1 + \ldots + d_L \le d_{\max}$ are considered. A rate 1/3 Turbo code with $(1,15/13)_8$ generator polynomials is used, and source block length $K = N = 128$ bits.

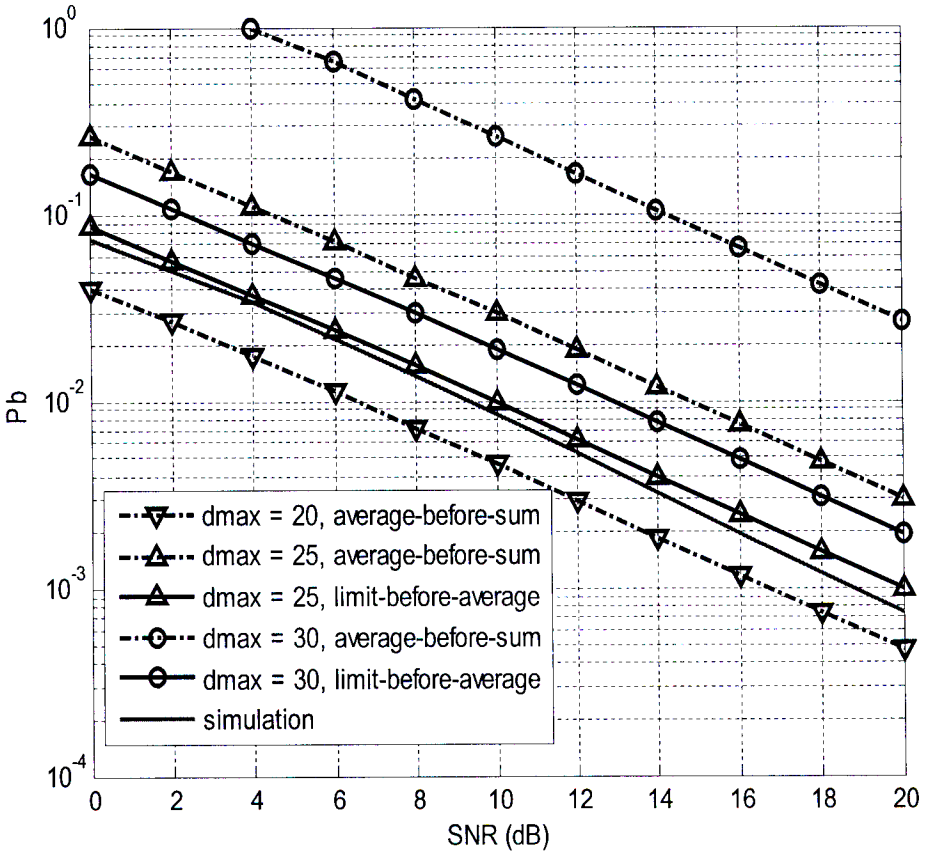

Figure 4.6: Union bounds on the bit error probability P_b for different values of $d_{\max}$.

Table 4.3 lists the inter-user block error rate (BLER) and cooperative case probabilities to DRPTC with coded cooperation for reciprocal inter-user channels. We should note that the -15dB inter-user SNR is approximately equivalent to the non-cooperative case. As expected,

the probability of occurrence of Case 1 in the 12dB inter-user channel is almost equal to 1, since it performs as good as in the perfect inter-user channel most of the time, whereas Case 2 almost never occurs and the probabilities of occurrence of Case 3 and Case 4 approach 0. In the 6dB inter-user channel, the probability of occurrence of Case 1 decreases while the probabilities of Cases 2, 3 and 4 increase. This makes sense, as the inter-user channel worsens. Finally for the non cooperative case, the probability of occurrence of Case 2 is equal to 1, whereas Cases 1, 3 and 4 do not occur in theory. Table 4.2 lists the inter-user block error rate (BLER) and cooperative case probabilities to RPTC in coded cooperation for reciprocal inter-user channels.

Figures 4.7 and 4.8 show the theoretical BER bounds versus simulation results for repeat-punctured turbo coded cooperation and dual repeat-punctured turbo coded cooperation in quasi-static fading with reciprocal inter-user channels, respectively. Combining [48] which uses a short recursion formula to compute transfer function coefficients and [49] yields tight bounds on the BER as shown in Figures 4.7 and 4.8.

It is noted that, the analytical bounds almost match the simulations results in both schemes under various inter-user channel qualities. This shows that the method used to derive the bounds yield tight bounds.

Table 4. 2: Inter-user BLER and cooperative case probabilities corresponding to RPTC in coded cooperation.

Inter-user SNR	Case Probabilities $\Psi=1$	$\Psi=2$	$\Psi=3$	$\Psi=4$
12dB	0.966	0.0311	0.0022	0.0022
6dB	0.874	0.1182	0.0080	0.0080
-15dB	~0	1	~0	~0

Table 4. 3: Inter-user BLER and cooperative case probabilities corresponding to DRPTC in coded cooperation.

Inter-user SNR	Case Probabilities $\Psi=1$	$\Psi=2$	$\Psi=3$	$\Psi=4$
12dB	0.979	0.0192	0.0018	0.0018
6dB	0.922	0.0743	0.0069	0.0069
-15dB	~0	1	~0	~0

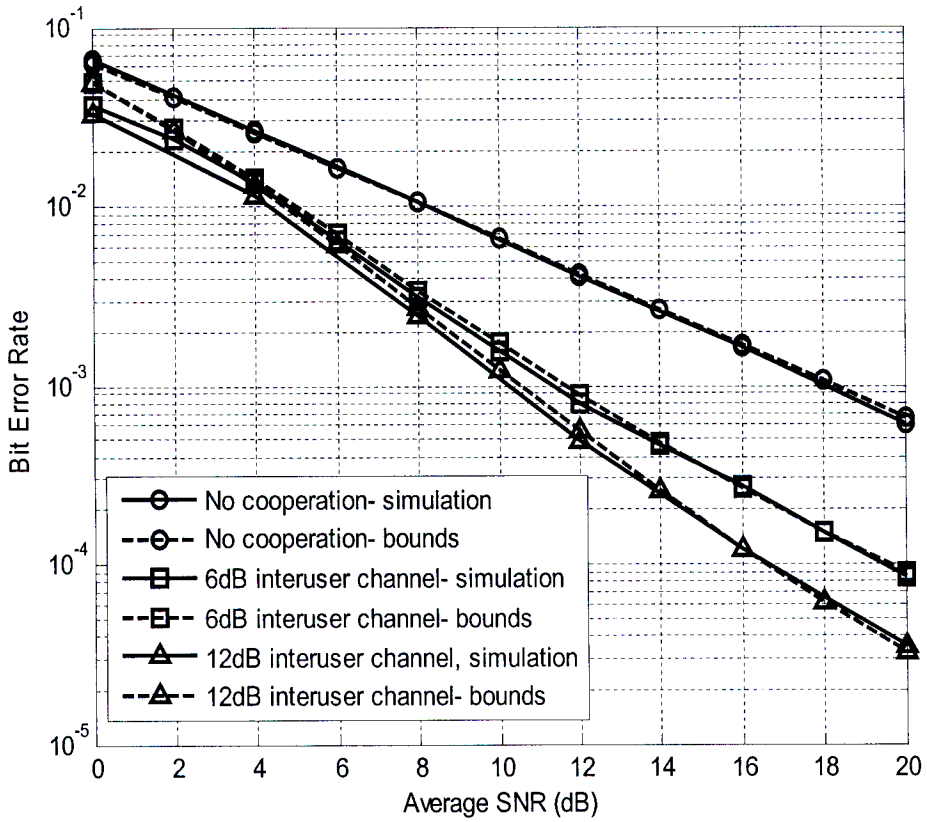

Figure 4.7: BER comparison between theoretical bounds and simulation of RPTC cooperation.

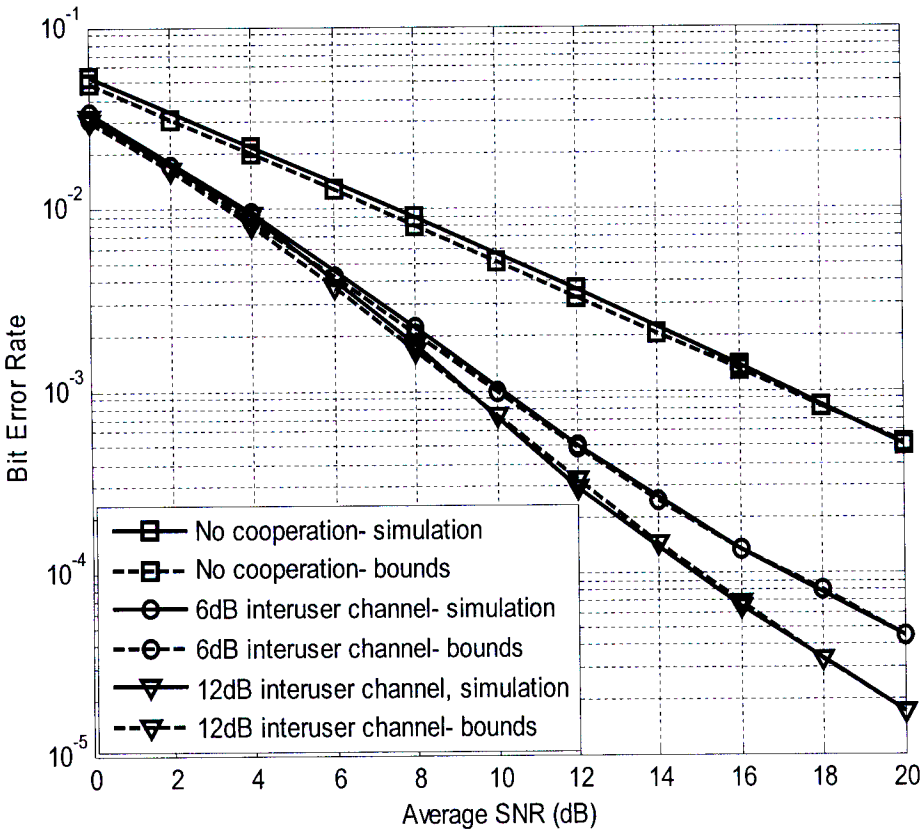

Figure 4.8: BER comparison between theoretical bounds and simulation of DRPTC cooperation.

4.5 Conclusion

This chapter presents the BER analysis of the two schemes presented in Chapter 3, that is, repeat-punctured turbo coded cooperation and dual repeat-punctured turbo coded cooperation over quasi-static fading channels. In Section 4.1, the conditional and unconditional PEPs for the four cooperative cases are derived. The conditional probability is later used for the computation of the end-to-end BER, since the conditional PEP yields loose BER bounds.

In Section 4.2, we derive the weight enumerating function which is essential in the computation of the union bounds of a given code. An algorithm that employs a short recursion formula to compute the transfer function coefficients is used in the derivation of weight enumerators.

The BER analysis is presented in Section 4.3. The unconditional PEP and weight enumerators derived in the previous sections are combined together to yield the end-to-end probabilities. Prior to combining unconditional PEP and weight enumerators, we calculate the probability of operating in the cooperative mode for the four cooperative cases.

Finally, numerical results are discussed in Section 4.4. A comparison of the theoretical BER bounds versus computer simulations of RPTC and DRPTC in coded cooperation is presented. This comparison shows that the simulated results confirm the analytical bounds.

CHAPTER 5

TURBO CODED COOPERATION WITH FORCED SYMBOL METHOD

In this chapter, an improvement of turbo coded cooperation scheme under the 6dB and 12dB inter-user channels SNR, which uses the Forced Symbol Method (FSM) [58, 59], is presented. In the first section of this chapter, we briefly introduce this method which improves the error rate of turbo codes and present previous works related to lowering the error floor of turbo codes. In Section 5.2, the Forced Symbol Method is described and some simulation results of turbo codes with FSM are presented. In Section 5.3, we introduce the new scheme which is the FSM with turbo coded cooperation. Two cases are presented here via computer simulations: the first one is the implementation of FSM at the destination while the second case is the implementation of FSM at each user. The latter case shows an improvement over the conventional turbo coded cooperation discussed in Chapter 2, especially under the 6dB inter-user channel. Finally, conclusions are drawn in the last section.

5.1 Introduction

It is known that the error floor of turbo codes mainly depends on the distance properties of the code. One of the methods to reduce the error floor of turbo codes is to increase the

minimum distance or free distance of the code which is determined by the interleaver. The design of interleavers which produce high free distances is a complicated task. Previous works have been done in order to lower the error floor using various methods. We briefly present some of these works here. In [24, 60-62], a serial concatenation of turbo code and high rate algebraic outer code was introduced to reduce the error floor. The use of Reed-Solomon (RS) outer code was proposed in [60], whereas the use of a Bose, Chaudhuri, and Hocquenghem (BCH) outer code was proposed by Andersen [61]. Narayanan *et al.* [62] proposed a method that employs a BCH outer code that protects a few positions in the information packet having the tendency to be in errors as oppose to Andersen's method in which the BCH outer code is used to protect the entire packet.

Another method for lowering the error floor was proposed by Narayanan and Stuber [63] that uses the List Viterbi Algorithm (LVA). The encoder structure comprises a concatenation of an outer CRC code and an inner turbo code. The CRC serves two purposes here: error detection and early stopping criterion for turbo decoding. A method for reducing the error floor based on distance spectrum analysis was presented in [67]. This method gets rid of the contribution of lowest distances to the error performance at the cost of the reduction of the code rate. Independent methods related to the Forced Symbol Method have been developed by Pishro [64] and Varnica [65] and by Papagiannis [66] for Low-Density Parity-Check (LDPC) codes and serial concatenated convolutional codes (SCCC) respectively. A new method has been proposed in [58,59] to lower the error rate without modifying the encoder structure, namely the interleaver structure. This method termed Forced Symbol Method also improves the error performance in the water fall region.

5.2 The Forced Symbol Method

5.2.1 Description of the FSM

The FSM as presented in [58, 59] is described. As mentioned above, the FSM improves the

error floor as well as waterfall region of turbo codes without modifying the encoder structure. This method required the use of an error detection and in this case, CRC code is used. The use of CRC code leads to a small reduction in the code rate and SNR, which can be negligible if the frame length is long enough. In general, the use of an n-bit CRC in an N - bits information length, reduces the code rates by a factor of $\frac{N-n}{N}$. We use a "genie" CRC here for error detection, that is, the decoded codeword is compared to the transmitted sequence and there is no reduction in the code rate. In [59], it is noted that the 16-bit CRC should provide essentially the same performance as the genie for improvements up to four order of magnitudes.

The FSM involves the following steps as described in [59]. First, the received sequence is decoded normally, that is, using a standard decoding algorithm. Secondly, in the case of error detection, find a few positions in the decoded sequence, that are most likely to have bit errors, with the order of the positions based on their increasing reliability values. One at a time and in the increasing order of reliability values, each potential bit/symbol position that is found to have a bit/symbol error respectively is flipped to the opposite value corresponding to the opposite value obtained after the first decoding. The newly received codeword with forced bit/symbol is decoded once again. The decoding on newly received codeword obtained after each bit flipping is repeated until all candidate bit positions have been tested. The candidate bit positions can vary but should be less than the frame length. Another stopping criterion of the repeated decoding is based upon successful decoding. The iterative decoding in this method is allowed to use the new information, that is, the positions of least reliable bits obtained from the first decoding, to make better decisions. The use of this new information and the error detection allow for an improvement of the reliability of the soft output values, hence an improvement of the error performance. It should be noted that the FSM does not require the determination of all bits errors in the decoded frame, only a few probable bit errors. The pseudo-code of the FSM as presented in [59] is repeated in Appendix B.

5.2.2 Simulations Results

The FSM is applied to single-binary turbo codes 8-state with generator polynomials $(1,15/13)_8$ and code rate 1/3. The information frame length is 1024 bits and the log-MAP decoding algorithm is performed with a maximum of 8 iterations. The transmission is assumed to be over an AWGN channel. Figure 5.1 shows the Frame Error Rate (FER) performance with the FSM of turbo codes with the parameters as listed above. We show here the performance improvement in the waterfall region. We denote by $FSM(L)$, the error performance obtained by testing the least reliable L symbols in the decoded sequence. The simulation results show that $FSM(1)$ outperforms $FSM(0)$ (standard decoding). Similarly $FSM(16)$ outperforms $FSM(4)$ and $FSM(0)$ by 0.15dB and 0.5dB respectively.

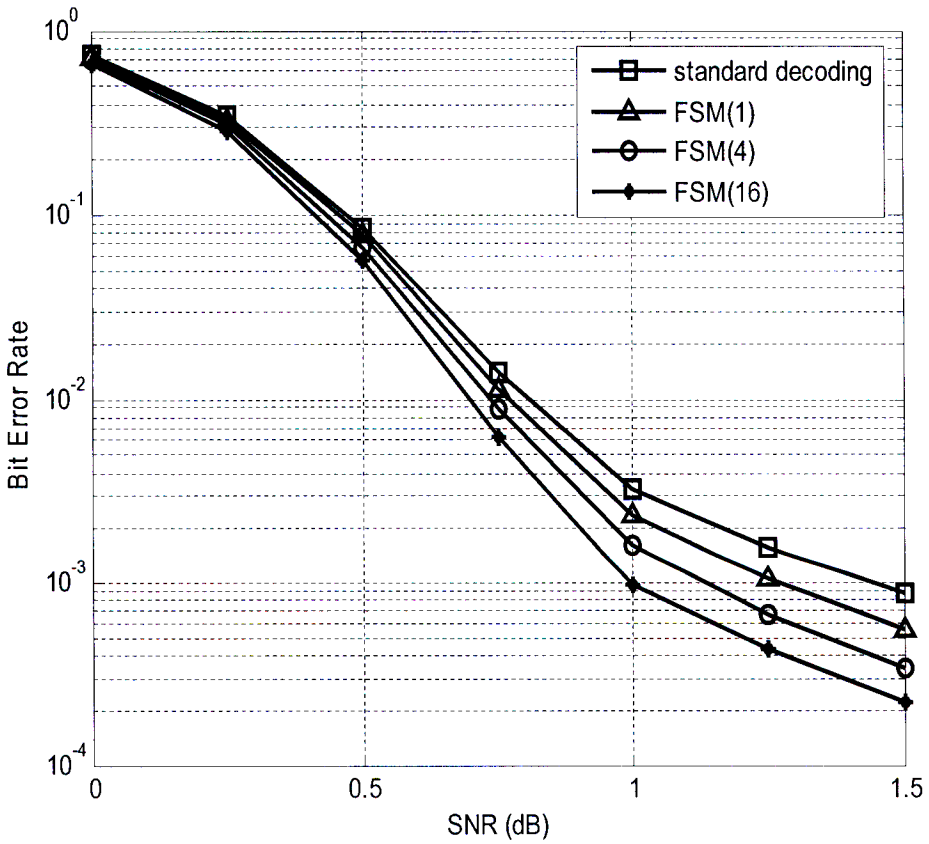

Figure 5.1: FER performance for an 8-state, single binary turbo code over AWGN channel with FSM.

5.3 Turbo Coded Cooperation with Forced Symbol Method

5.3.1 FSM Applied at the Destination

Since turbo decoding is employed at the destination, the Forced Symbol Method is applied as described in [58, 59]. The results on Figure 5.2 show that there is a very little (negligible) improvement (about 0.1dB gain at 10^{-4}) under various inter-user SNR channel. This is predictable, since the inter-user channel under various qualities (various SNRs) is not improved and remains the same as in the case of the conventional turbo coded cooperation. Figure 5.2 show the comparison of the FER performance between the conventional turbo coded cooperation and the forced symbol turbo coded cooperation.

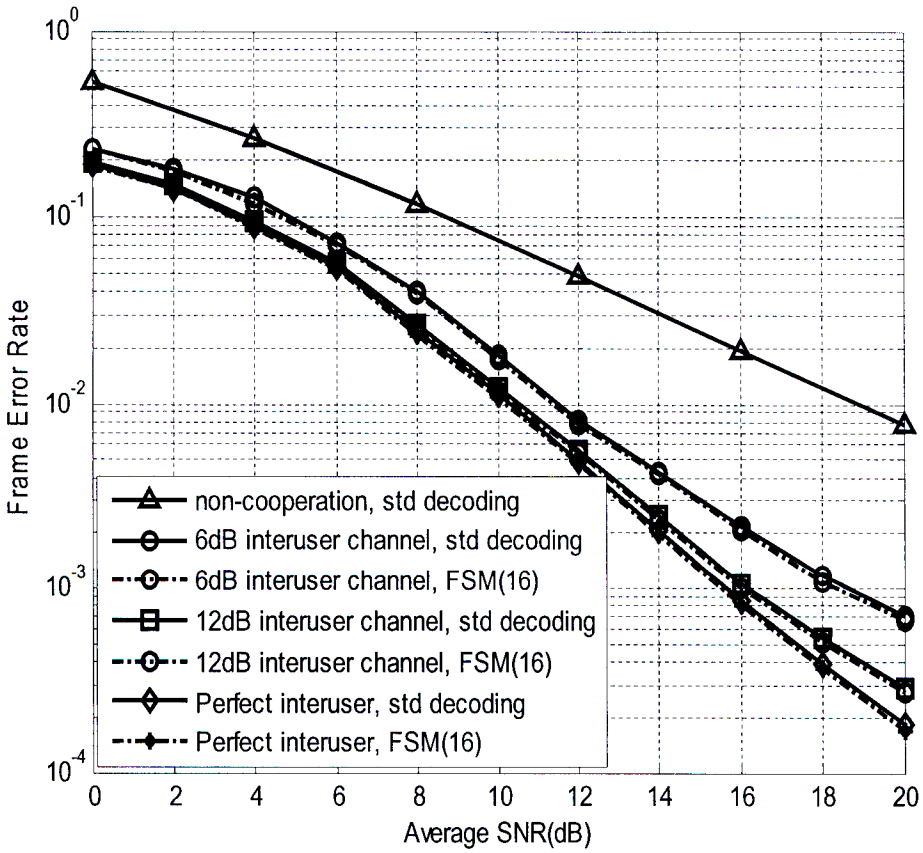

Figure 5.2: FER performance comparison between TC cooperation with standard decoding and TC cooperation with FSM.

5.3.2 FSM Applied at Each User

Instead of using the FSM at the destination, we apply the method at each user in order to enhance the inter-user channel quality. By doing so, the perfect inter-user channel SNR is considered as the bound for this scheme, in other words any inter-user channel SNR (6dB or 12dB) cannot outperform the perfect inter-user channel.

The FSM applied here is almost similar to the one applied at the destination. The algorithm used for the FSM is similar to the one described in Appendix B, except that, in this case, the Viterbi decoding algorithm for convolutional encoding is employed instead of the iterative decoding that was used in the case of turbo encoding. By applying the FSM at each user, we correct some bit errors, hence improve the quality of the inter-user channel. This means that, a 6dB inter-user channel with FSM can perform as good as or better than a 12dB inter-user channel without applying FSM.

Figure 5.3 shows the FER performance of the turbo coded cooperation with FSM at each user and the conventional turbo coded cooperation under the 6dB inter-user channel. The turbo coded cooperation with $FSM(16)$ outperforms the conventional turbo coded cooperation scheme by almost 3dB at high SNR values. This is an impressive improvement over the conventional turbo coded cooperation scheme.

Figure 5.4 shows the FER performance of turbo coded cooperation with FSM at each user under the 6dB and 12 dB inter-user channels versus the conventional turbo coded cooperation scheme under the 6dB, 12dB and perfect inter-user channel SNRs. It is noted that the 6dB inter-user SNR with $FSM(16)$ performs as good as the 12dB inter-user for low to moderate SNR values and outperforms the 12dB inter-user channel for high SNRs by approximately 1dB. The turbo coded cooperation with $FSM(16)$ under the 12dB inter-user SNR performs as good as (they overlap) the perfect inter-user channel which confirms our predictions. We also note that the turbo coded cooperation with $FSM(16)$ under the perfect inter-user SNR

is the bound for this new scheme.

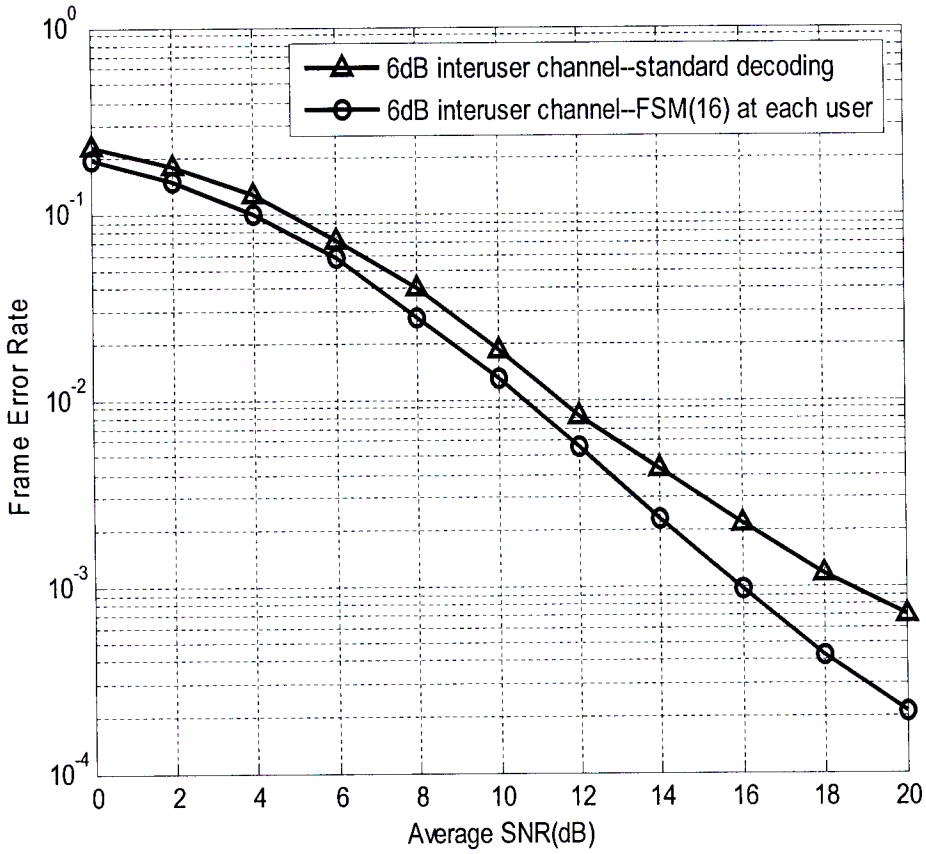

Figure 5.3: FER performance comparison between TC cooperation with standard decoding and FSM under 6dB inter-user channel.

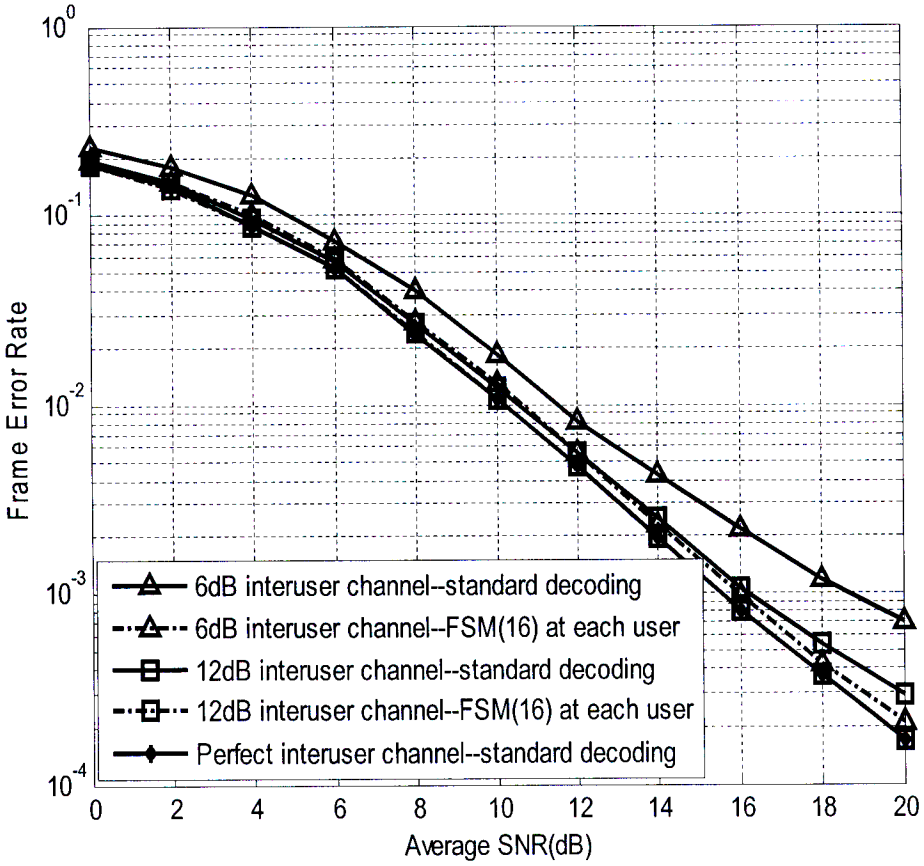

Figure 5.4: FER performance comparison of TC cooperation with standard decoding and FSM under various inter-user SNRs.

5.4 Conclusion

In this Chapter, a modified turbo coded cooperation scheme is presented. This scheme combines turbo coded cooperation with the FSM. The proposed scheme presents some improvement over the conventional turbo coded cooperation when applied at the users. This works by flipping the bit that might seem to be in error and decode once again. By doing so, the inter-user channel improves, hence the end-to-end BER improves as well. The proposed scheme presents some processing delay due to the additional FSM algorithm at the users or base station, which will require more decoding whenever a bit is in error.

CHAPTER 6

CONCLUSIONS AND FUTURE WORK

6.1 Conclusions

Cooperative diversity allows multiple single-antenna systems to cooperate and share their antenna creating virtual MIMO systems, and hence reaping the benefits of diversity. Early works on user cooperation have been proposed by Sendonaris *et al.* [5] [6], Laneman *et al.* [11]. It should be noted that the genesis of cooperation goes back to the work of Cover and Gamal [7]. Coded cooperation [15-19] have been introduced recently and combined channel coding with coded cooperation. The work in this dissertation is based on coded cooperation.

In Chapter 1, a brief history of cooperative communication is given. Cooperative signaling techniques to date are described.

Chapter 2 describes coded cooperation framework in detail. Coded cooperation integrates user cooperation with channel coding. Various coding methods can be implemented in coded cooperation. Implementation of coded cooperation using RCPC codes [15-19] is first discussed here, followed by coded cooperation with turbo codes [23]. An overview of turbo codes is presented with various simulation results, prior to describing turbo coded cooperation.

Two new protocols termed repeat-punctured turbo coded cooperation and dual repeat-punctured turbo coded cooperation are designed and presented. These two schemes are the extensions of coded cooperation using turbo codes. The first scheme uses the repeat-punctured turbo codes, which was introduced in [44] and yield some gain over turbo codes at moderate to high SNRs. DRPTC based on RPTC, in coded cooperation is then introduced. The performance of the three schemes altogether in quasi-static fading channel, under various inter-user channels SNR in Chapter 3 is compared via computer simulations. DRPTC outperforms both TC and RPTC in coded cooperation with significant gains at low BER. It can therefore be concluded that RPTC and DRPTC are promising error correcting techniques that can be applied in coded cooperation since they yield impressive results over existing conventional turbo coded cooperation scheme with the same rate .

An analysis of the bounds for bit error probability of the coded cooperation using RPTCs and DRPTCs is made. The analysis begins with the derivation of the PEP for the four cooperative cases, since we assume a two-user cooperation. It is later found that the unconditional PEP leads to very loose bounds. Thus, we consider the conditional PEP. The second part of this analysis consisted of the derivation of the weight enumerating functions, which a short recursion formula and yield tight bounds for bit error probability. Finally, we combine the PEP and weight enumerating function to derive the end-to-end bit error probabilities and illustrate the comparison between the theoretical bounds and simulation results obtained in Chapter 3. It is found that the analytical bounds developed for the two new schemes are confirmed by simulations results, since they yield almost the same performance. The analysis based on transfer function yield good results and can therefore be applied in any turbo coded related cooperation scheme.

6.2 Future Work

Many open problems related to the work in this dissertation still remain to be investigated. The work performed in Chapter 3 on Repeat-Punctured Turbo Coded and Dual-Repeat

Punctured Turbo Coded cooperation has been done over quasi-static or slow Rayleigh fading channels. Future research work could extend the work in Chapter 3 to fast fading channels or multi-path fading and analyze the performance.

In chapter 5, turbo coded cooperation using the FSM under various inter-user channel SNRs is implemented and simulations results show a considerable improvement over the turbo coded cooperation using standard decoding with corresponding inter-user channel SNRs. Future work should extend to the derivation of the theoretical bounds and complexity analysis of this scheme.

Coded cooperation integrates channel coding with cooperation. Various channel coding schemes can be implemented in coded cooperation. In this dissertation, RPTCs and DRPTCs in cooperation are implemented. Previous work [16] [23] have integrated turbo codes with cooperation and yield impressive results. Other channel coding schemes such as Low-Density Parity-Check (LDPC) codes can also be implemented and investigated in coded cooperation framework. LDPC codes, like turbo codes are part of a broad family of error-control coding techniques termed compound codes. LDPC codes which perform as good as turbo codes in AWGN channels, present some advantages over turbo codes such as:

- Lower complexity of the iterative decoding.
- The inexistence of low weight codewords.

The channel model used throughout this research work is the Rayleigh fading. Other channel models such as Rician fading channels mostly encountered in satellite channels and land mobile channels, where a Line of Sight (LOS) is possible, and Nagakami fading channels remain to be fully investigated.

APPENDICES

APPENDIX A

Derivation of $\left[I - A(L,I,D))^{-1}\right]_{0^m,0^m}$ Using the Adjoint Matrix Method and the Transfer Function.

Using the following notation,

$$A^{-1} = \frac{1}{|A|}(C_{ij}^{'})^T = \frac{1}{|A|}\begin{pmatrix} C_{11} & C_{21} & \cdots & C_{j1} \\ C_{12} & \ddots & \vdots & C_{j2} \\ \vdots & \cdots & \ddots & \vdots \\ C_{1i} & C_{2i} & \cdots & C_{ji} \end{pmatrix}, \tag{A.1}$$

where $(C_{ij}^{'})^T$ is equal to $adj(A)$ the adjoint of A. Therefore, the first entry to A^{-1} can be computed as follows

$$\left[A^{-1}\right]_{0^m,0^m} = \frac{C_{11}}{|A|}. \tag{A.2}$$

Hence

$$\left[(I - A(L,I,D))^{-1}\right]_{0^m,0^m} = \left[\begin{pmatrix} 1-L & -LID & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -LI & -LD & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & -LD & -LI & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & -LID & -L \\ -LID & -L & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & -LD & -LI & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -LI & -LD & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -L & 1-LID \end{pmatrix}^{-1}\right]_{0^m,0^m}$$

$$= \frac{C_{11}}{|I - A(L,I,D)|}, \tag{A.3}$$

where

$$C_{11} = \begin{vmatrix} 1 & -LI & -LD & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -LD & -LI & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & -LID & -L \\ -L & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & -LD & -LI & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -LI & -LD & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -L & 1-LID \end{vmatrix}.$$

We use MATLAB to compute (A.3), since it is computationally difficult to get determinant of a 7×7 matrix (C_{11}) or an 8×8 matrix ($|I - A(L,I,D)|$).

tf_0_0 =

L^2*(L^4*D^2*I^6-L^3*D*I^5-2*L^4*I^4*D^4-L^4*I^4+2*L^3*I^3*D^3+L^2*D^3*I^3+L*I^3*D^3+I^2*L^4*D^6+2*L^4*I^2*D^2-L*I^2*D^2-I^2*D^2-L^3*D^5*I-L^2*I*D-L*I*D-D^4*L^4)/(4*I^4*L^7*D^4-2*I^2*L^7*D^2+L*I*D+L^4*I^4*D^4+I^4*L^7*D^8-2*I^6*L^7*D^6+L^5*I^5*D^5+L^7*D^4*I^8-I*D-L^4*I^2*D^2-L^3*I*D+L^3*I^3*D^3+L^2*I^2*D^2-L^6*D^4+L^7*D^4-2*I^4*L^6*D^4+I^2*L^6*D^6+2*I^2*L^6*D^2+L^6*D^2*I^6-I^4*L^6-2*I^2*L^7*D^6-2*L^5*D^3*I^3+I^4*L^7-2*L^7*D^2*I^6+L^5*I*D) (A.4)

APPENDIX B

Pseudo-Code for the Forced Symbol Method

Let $\mathbf{u}$ and $\mathbf{y}$ be the information frame and the received codeword, respectively. For simplicity assume the use of antipodal signaling that maps the bit b to $(-1)^b$. Let E be a real number greater than $d_{\min}$. To force the turbo decoder to decide for the bit b_i at position i in u, it is sufficient to insert the impulse $I_i = (-1)^{b_i} E$ at position i in $\mathbf{y}$, in other words to flip the bit in the corresponding received codeword. Let Ω be the set of positions in the decoded frame, $\hat{\mathbf{u}}$, that are most likely to have bits errors. Let L be the number of elements in Ω. The positions of interest are recorded in Ω according to their increasing reliabilities (i.e., position $\Omega(i)$ is more likely to have an error than position $\Omega(i+1)$). In the sequel, j is the position of the bit forced to change or flip.

- Set the noise variance σ^2, if it is required for iterative decoding;
- Choose E ($E > d_{\min}$);
- iterative decoding of $y \Rightarrow \hat{u}$;
- If (error detected)

  || determine Ω;

  || set $i = 0$;

  || while ((error detected) and ($i < L$))

  - set $j = \Omega(i)$;
  - set $z = y(j)$;
  - set $y(j) = -(-1)^{\hat{u}(j)} E$;
  - iterative decoding of $\mathbf{y}$;

- set $y(j) = z$;
- set $i = i + 1$;

end while

end if

APPENDIX C

Derivation of the Turbo Code Decoder Equations (MAP algorithm)

A modified version of the BCJR algorithm for performing symbol-by-symbol MAP decoding is first discussed. It is then shown how this algorithm is incorporated into an iterative decoder employing two BCJR-MAP decoders. The following definitions shall be required:

- E1 is a notation for encoder 1.
- E2 is a notation for encoder 2.
- D1 is a notation for decoder 1.
- D2 is a notation for decoder 2.
- m is the constituent encoder memory.
- S is the set of all 2^m constituents encoder states.
- $\mathbf{x}^{\mathbf{s}} = (x_1^s, x_2^s, \ldots, x_N^s) = (u_1, u_2, \ldots, u_N)$ is the encoder input word.
- $\mathbf{x}^{\mathbf{p}} = (x_1^p, x_2^p, \ldots, x_N^p)$ is the parity word generated by a constituent encoder.
- $\mathbf{y_k} = (y_k^s, y_k^p)$ is a noisy (AWGN version) of (x_k^s, x_k^p).
- $\mathbf{y_a^b} = (y_a, y_{a+1}, \ldots, y_b)$.
- $\mathbf{y} = y_1^N = (y_1, y_2, \ldots, y_N)$ is the noisy received codeword.

The modified BCJR algorithm

In the symbol-by-symbol MAP decoder, the decoder decides $u_k = +1$ if $P(u_k = +1 \mid \mathbf{y}) > P(u_k = -1 \mid \mathbf{y})$, and it decides $u_k = -1$ otherwise. More succinctly, the decision $\hat{u}_k$ is given by

$$\hat{u}_k = sign(L(u_k)) \tag{C.1}$$

where $L(u_k)$ is the log *a-posteriori probability* (LAPP) ratio defined as

$$L(u_k) = \log\left(\frac{P(u_k = +1 \mid \mathbf{y})}{P(u_k = -1 \mid \mathbf{y})}\right). \tag{C.2}$$

Incorporating the code's trellis, this may be written as

$$L(u_k) = \log\left(\frac{\sum_{S^+} p(s_{k-1} = s', s_k = s, y)/p(\mathbf{y})}{\sum_{S^-} p(s_{k-1} = s', s_k = s, y)/p(\mathbf{y})}\right), \tag{C.3}$$

where $s_k \in S$ is the space of the encoder at time k, S^+ is the set of ordered pairs (s', s) corresponding to all state transitions $(s_{k-1} = s') \rightarrow (s_k = s)$ caused by data input $u_k = +1$, and S^- is similarly defined for $u_k = -1$.

Observe we may cancel $p(\mathbf{y})$ in (C.3) which means we require only an algorithm for computing $p(s', s, y) = p(s_{k-1} = s, s_k = s, y)$. The BCJR algorithm [33] for doing this is

$$p(s', s, y) = \alpha_{k-1}(s').\gamma_k(s', s).\beta_k(s) \tag{C.4}$$

where $\alpha_k(s) = p(s_k = s, y_1^k)$ is computed recursively as

$$\alpha_k(s) = \sum_{s' \in S} \alpha_{k-1}(s')\gamma_k(s', s), \tag{C.5}$$

with initial conditions

$$\alpha_0(0) = 1 \quad \text{and} \quad \alpha_0(s \neq 0) = 0. \tag{C.6}$$

(These conditions state that the encoder is expected to start in state 0.) The probability $\gamma_k(s', s)$ in (C.5) is defined as

$$\gamma_k(s', s) = p(s_k = s, y_k \mid s_{k-1} = s'), \tag{C.7}$$

and will be discussed further below. The probabilities $\beta_k(s) = p(y_{k+1}^N \mid s_k = s)$ in (C.4) are computed in a "backward" recursion as

$$\beta_{k-1}(s') = \sum_{s \in S} \beta_k(s)\gamma_k(s', s), \tag{C.8}$$

with boundary conditions

$$\beta_N(0)=1 \quad \text{and} \quad \beta_N(s\neq 0)=0 \quad \text{for the first encoder.} \tag{C.9}$$

$$\beta_N(s)=\frac{1}{2^m} \quad \text{for the second encoder.}$$

(The encoder is expected to end in state 0 after N input bits, implying that the last m input bits, called *termination bits*, are also elected.) Figure C.1 shows how $\alpha_k(0)$ and $\beta_k(0)$ are computed recursively in a trellis.

Unfortunately, cancelling the divisor $p(\mathbf{y})$ in (C.3) leads to a numerically unstable algorithm. We can include division by $p(\mathbf{y})/p(y_k)$ in the BCJR algorithm[1] by defining

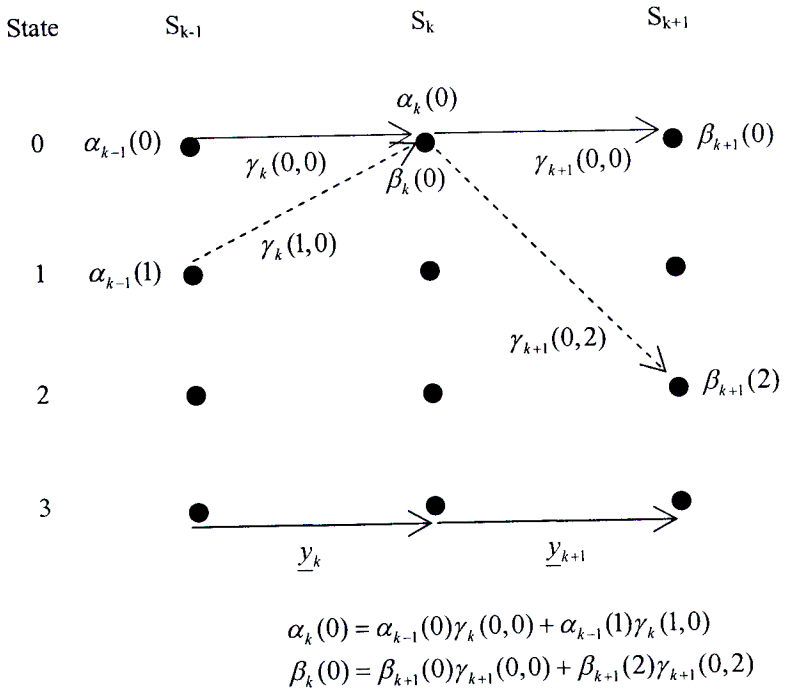

Figure C.1: Recursive calculation of $\alpha_k(0)$ and $\beta_k(0)$.

[1] Unfortunately, dividing by simply $p(y)$ to obtain $p(s',s\,|\,y)$ also leads to an unstable algorithm. Obtaining $p(s',s\,|\,y)p(y_k)$ instead of the APP $p(s',s\,|\,y)$ presents no problem since an APP ratio is computed so that the unwanted factor $p(u_k)$ cancels. See equation (C.14) below.

modified probabilities

$$\tilde{\alpha}_k(s) = \alpha_k(s) / p(y_1^k)$$

and

$$\tilde{\beta}_k(s) = \beta_k(s) / p(y_{k+1}^N \mid y_1^k) .$$

Dividing (C.4) by $p(y)/p(y_k) = p(y_1^{k-1})p(y_{k+1}^N \mid y_1^k)$, we obtain

$$p(s', s \mid y)p(y_k) = \tilde{\alpha}_{k-1}(s').\gamma_k(s', s).\tilde{\beta}_k(s) . \tag{C.10}$$

Note that since $p(y_1^k) = \sum_{s \in S} \alpha_k(s)$, the values $\tilde{\alpha}_k(s)$ may be computed from $\{\alpha_k(s) : s \in S\}$ via

$$\tilde{\alpha}_k(s) = \alpha_k(s) / \sum_{s \in S} \alpha_k(s) . \tag{C.11}$$

But since we would like to avoid storing both $\{\alpha_k(s)\}$ and $\{\tilde{\alpha}_k(s)\}$, we can use (C.5) in (C.11) to obtain a recursion involving only $\{\tilde{\alpha}_k(s)\}$,

$$\begin{aligned} \tilde{\alpha}_k(s) &= \frac{\sum_{s'} \alpha_{k-1}(s')\gamma_k(s', s)}{\sum_s \sum_{s'} \tilde{\alpha}_{k-1}(s')\gamma_k(s', s)} \\ &= \frac{\sum_{s'} \tilde{\alpha}_{k-1}(s')\gamma_k(s', s)}{\sum_s \sum_{s'} \tilde{\alpha}_{k-1}(s')\gamma_k(s', s)} , \end{aligned} \tag{C.12}$$

where the second equality follows by dividing the numerator and the denominator by $p(y_1^{k-1})$.

The recursion for $\tilde{\beta}_k(s)$ can be obtained by noticing that

$$\begin{aligned} p(y_k^N \mid y_1^{k-1}) &= p(y_1^k) \frac{p(y_{k+1}^N \mid y_1^k)}{p(y_1^{k-1})} \\ &= \sum_s \sum_{s'} \alpha_{k-1}(s')\gamma_k(s', s) . \frac{p(y_{k+1}^N \mid y_1^k)}{p(y_1^{k-1})} \\ &= \sum_s \sum_{s'} \tilde{\alpha}_{k-1}(s')\gamma_k(s', s) . p(y_{k+1}^N \mid y_1^k) \end{aligned}$$

so that dividing (C.8) by this equation yields

$$\tilde{\beta}_{k-1}(s') = \frac{\sum_s \tilde{\beta}_k(s)\gamma_k(s',s)}{\sum_s \sum_{s'} \tilde{\alpha}_{k-1}(s')\gamma_k(s',s)}. \tag{C.13}$$

In summary, the modified BCJR-MAP algorithm involves computing the LAPP ration $L(u_k)$ by combining (C.3) and (C.10) to obtain

$$L(u_k) = \log\left(\frac{\sum_{S^+} \tilde{\alpha}_{k-1}(s').\gamma(s',s).\tilde{\beta}_k(s)}{\sum_{S^-} \tilde{\alpha}_{k-1}(s').\gamma(s',s).\tilde{\beta}_k(s)}\right) \tag{C.14}$$

where the $\tilde{\alpha}$'s and $\tilde{\beta}$'s are computed recursively via (C.12) and (C.13), respectively. Clearly the $\{\tilde{\alpha}_k(s)\}$ and $\{\tilde{\beta}_k(s)\}$ share the same boundary conditions as their counterparts as given in (C.6) and (C.9). Computation of the probabilities $\gamma_k(s',s)$ will be discussed shortly.

Iterative MAP decoding

From Bayes' rule, the LAPP ratio for an arbitrary MAP decoder can be written as

$$L(u_k) = \log\left(\frac{P(\mathbf{y} \mid u_k = +1)}{P(\mathbf{y} \mid u_k = -1)}\right) + \log\left(\frac{P(u_k = +1)}{P(u_k = -1)}\right)$$

with the second term representing a priori information. Since $P(u_k = +1) = P(u_k = -1)$ typically, the a priori term is usually zero for the conventional decoders. However, for iterative decoders, D1 receives extrinsic or soft information for each u_k from D2. Similarly, D2 receives extrinsic information from D1 and the decoding iteration proceeds as $\text{D1} \rightarrow \text{D2} \rightarrow \text{D1} \rightarrow \text{D2} \rightarrow \ldots$, with the previous decoder passing soft information along to the next decoder at each half-iteration except for the first. The idea behind extrinsic information is that D2 provides soft information to D1 for each u_k, using only information not available to D1 (i.e., E2 parity); D1 does likewise for D2.

An iterative decoder using component BCJR-MAP decoders is shown in Figure 2.8. Observe how permuters and de-permuters are involved in arranging systematic, parity, and extrinsic

information in the proper sequence for each decoder.

We now show how the extrinsic information is extracted from the modified-BCJR version of the LAPP ratio embodied in (C.14). We first observe that $\gamma_k(s',s)$ may be written as (cf. equation C.7)

$$\begin{aligned}\gamma_k(s',s) &= P(s\,|\,s')p(y_k\,|\,s',s)\\ &= P(u_k)p(y_k\,|\,u_k)\end{aligned}$$

where the event u_k corresponds to the event $s' \to s$. Defining

$$L^e(u_k) = \log\left(\frac{P(u_k=+1)}{P(u_k=-1)}\right),$$

observe that we may write

$$\begin{aligned}P(u_k) &= \left(\frac{\exp[-L^e(u_k)/2]}{1+\exp[-L^e(u_k)]}\right).\exp[u_k L^e(u_k)/2]\\ &= A_k \exp[u_k L^e(u_k)/2] \qquad \text{(C.15)}\end{aligned}$$

where the first equality follows since it equals

$$\left(\frac{\sqrt{P_-/P_+}}{1+P_-/P_+}\right)\sqrt{P_+/P_-} = P_+ \quad \text{when} \quad u_k=+1 \quad \text{and}$$

$$\left(\frac{\sqrt{P_-/P_+}}{1+P_-/P_+}\right)\sqrt{P_-/P_+} = P_- \quad \text{when} \quad u_k=-1,$$

where we have defined $P_+ = P(u_k=+1)$ and $P_- = P(u_k=-1)$ for convenience. As for $p(y_k\,|\,u_k)$, we may write (recall $y_k=(y_k^s,y_k^p)$ and $x_k=(x_k^s,x_k^p)=(u_k,x_k^p)$)

$$\begin{aligned}p(y_k\,|\,u_k) &\propto \exp\left[-\frac{(y_k^s-u_k)^2}{2\sigma^2}-\frac{(y_k^p-x_k^p)^2}{2\sigma^2}\right]\\ &= \exp\left[-\frac{{y_k^s}^2+{u_k}^2+{y_k^p}^2+{x_k^p}^2}{2\sigma^2}\right].\exp\left[\frac{u_k y_k^s + x_k^p y_k^p}{\sigma^2}\right]\\ &= B.\exp\left[\frac{u_k y_k^s + x_k^p y_k^p}{\sigma^2}\right]\end{aligned}$$

so that

$$\gamma_k(s',s) \propto A_k B_k \exp[u_k L^e(u_k)/2].\exp\left[\frac{u_k y_k^s + x_k^p y_k^p}{\sigma^2}\right]. \qquad \text{(C.16)}$$

Now since $\gamma_k(s',s)$ appears in the numerator (where $u_k = +1$) and the denominator (where $u_k = -1$) of (C.14), the factor $A_k B_k$ will cancel as it is independent of u_k. Also, since we assume BPSK transmission over the channel, $\frac{E_c}{N_0/2} = \frac{1}{\sigma^2}$ so that $\sigma^2 = N_0/2E_c$ where $E_c = rE_b$ is the energy per channel bit. From (C.16), we then have

$$\gamma_k(s',s) \sim \exp\left[\frac{1}{2}u_k(L^e(u_k) + L_c y_k^s) + \frac{1}{2}L_c y_k^p x_k^p\right]$$

$$= \exp\left[\frac{1}{2}u_k(L^e(u_k) + L_c y_k^s)\right].\gamma_k^e(s',s) \qquad \text{(C.17)}$$

where $L_c = \frac{4E_c}{N_0}$ and where $\gamma_k^e(s',s) = \exp\left[\frac{1}{2}L_c y_k^p x_k^p\right]$.

Combining (C.17) with (C.14) we obtain

$$L(u_k) = \log\left(\frac{\sum_{S^+}\tilde{\alpha}_{k-1}(s').\gamma_k^e(s',s).\tilde{\beta}_k(s).C_k}{\sum_{S^-}\tilde{\alpha}_{k-1}(s').\gamma_k^e(s',s).\tilde{\beta}_k(s).C_k}\right)$$

$$= L_c y_k^s + L^e(u_k) + \log\left(\frac{\sum_{S^+}\tilde{\alpha}_{k-1}(s').\gamma_k^e(s',s).\tilde{\beta}_k(s)}{\sum_{S^-}\tilde{\alpha}_{k-1}(s').\gamma_k^e(s',s).\tilde{\beta}_k(s)}\right). \qquad \text{(C.18)}$$

where $C_k = \exp\left[\frac{1}{2}u_k(L^e(u_k) + L_c y_k^s)\right]$. The second equality follows since $C_k(u_k = +1)$ and $C_k(u_k = -1)$ can be factored out of the summations in the numerator and denominator, respectively. The first term in (C.18) is sometimes called the *channel value*, the second term represents any *a priori* information about u_k provided by a previous decoder, and the third term represents *extrinsic information* that can be passed on to a subsequent decoder. Thus, for example, on any given iteration, D1 computes

$$L_1(u_k) = L_c y_k^s + L_{21}^e(u_k) + L_{12}(u_k)$$

where $L_{21}^e(u_k)$ is the extrinsic information passed from D2 to D1, and $L_{12}^e(u_k)$ which is to be used as an extrinsic information from D1 to D2.

REFERENCES

[1] G. J. Foschini, "Layered space-time architecture for wireless communication in a fading environment when using multi-element antennas," *Bell Labs Technical Journal*, 1(2): pp. 41-59, 1996.

[2] I. E. Teletar, "Capacity of multiple-antenna Gaussian channels," *European Transactions on Telecommunications*, 10: pp. 585-595, November 1999.

[3] C. E. Shannon, "A mathematical theory of communications," *Bell System Technical Journal*, vol. 27, pp. 379-423 and 623-656, 1948.

[4] V. Tarokh, N. Seshadri and A. R. Calderbank, "Space-time block codes for high data rate wireless communication performance criterion and code construction," *IEEE Transactions on Information Theory*, vol. 44, pp. 744-765, 1998.

[5] A. Sendonaris, E. Erkip, B. Aazhang, "User cooperation diversity-Part I: System description," *IEEE Transactions on Communications*, vol. 51, no. 11, pp. 1927-1938, November 2003.

[6] A. Sendonaris, E. Erkip, B. Aazhang, "User cooperation diversity-Part II: Implementation aspects and performance analysis," *IEEE Transactions on Communications*, vol. 51, no. 11, pp. 1927-1938, November 2003.

[7] T. M. Cover and A. E. Gamal, "Capacity theorems for the relay channel," *IEEE Transactions on Information Theory*, vol. 25, no. 5, pp. 572-584, September 1979.

[8] A. Host-Madsen, "On the capacity of wireless relaying," in *Proc. IEEE Vehicular Technology Conference (VTC)*, vol. 3, Vancouver, B.C., Canada, pp. 1333-1337, September 2002.

[9] A. Norastinia, T. E. Hunter and A. Hedayat, "Cooperative communication in wireless

networks," *IEEE Communications Magazine*, vol. 42, no. 10, pp. 68-73, October 2004.

[10] A. Sendonaris, E. Erkip, and B. Aazhang, "Increasing uplink capacity via user cooperation diversity," in *Proc. IEEE International Symposium on Information Theory (ISIT)*, Cambridge, MA, p. 156, August 1998.

[11] J. N. Laneman, G. W. Wornell and D. N. C. Tse, "An efficient protocol for realizing cooperative diversity in wireless networks," *Proc. IEEE International Symposium on Information Theory (ISIT)*, Washington, DC, p. 294, June 2001.

[12] J. N. Laneman, D. N. C. Tse, G. W. Wornell, "Cooperative diversity in wireless networks: Efficient protocols and outage behaviour," *IEEE Transactions on Information Theory*, vol. 50, no. 12, pp. 3062-3080, December 2004.

[13] A. Sendonaris, *Advanced techniques for next-generation wireless systems*, PhD thesis, Rice University, May 1999.

[14] J. N. Laneman and G. W. Wornell, "Energy-efficient antenna sharing and relaying for wireless networks," in *Proc. IEEE Wireless Communication and Networking Conference (WCNC)*, Chicago, IL, pp. 7-12, September 2000.

[15] T. E. Hunter and A. Nosratinia, "Cooperative diversity through coding," in *Proc. IEEE International Symposium on Information Theory (ISIT)*, Lausanne, Switzerland, p. 220, July 2002.

[16] T. E. Hunter, *Coded Cooperation: A new framework for user cooperation in wireless networks*, PhD thesis, University of Texas at Dallas, May 2004.

[17] T. E. Hunter and A. Nosratinia, "Coded cooperation under slow fading, fast fading and power control," in *Proc. Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, CA, pp. 118-122, November 2002.

[18] T. E. Hunter and A. Nosratinia, "Performance analysis of coded cooperation diversity," in *Proc. IEEE International Conference on Communications (ICC)*, Anchorage, AK, May 2003.

[19] T. E. Hunter and A. Nosratinia, "Diversity through coded cooperation," IEEE Transactions on Wireless Communications, vol. 5, no. 2, pp. 283-289, February 2006.

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

[21] J. Hagenauer, "Rate-compatible punctured convolutional codes (RCPC codes) and their

applications," *IEEE Transactions on Communications*, vol. 36, no. 4, pp. 389-400, April 1988.

[22] S. M. Alamouti, "A simple transmit diversity technique for wireless communications," IEEE *Journal on Selected Areas in Communications*, vol. 16, no. 8, pp. 1451-1458, October 1998.

[23] M. Janani, A. Hedayat, T. Hunter and A. Norastinia, "Coded cooperation in wireless communications: space-time transmission and iterative decoding," *IEEE Transactions on Signal Processing*, vol. 52, no. 2, pp.362-371, February 2004.

[24] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting and decoding: Turbo codes," in *Proc. IEEE International Conference on Communications*, Geneva, Switzerland, pp. 1064-1070, May 1993.

[25] C. Berrou and A. Glavieux, "Near optimum error correcting coding and decoding: Turbo codes," *IEEE Transactions on Communications*, vol. 44, no. 10, pp. 1261-1271, October 1996.

[26] C. E. Shannon, "A mathematical theory of communications," *Bell System Technical Journal*, vol. 27, pp. 379-423 and 623-659, 1948.

[27] S. Benedetto and G. Montorsi, "Unveiling turbo codes: Some results on parallel concatenated coding schemes," *IEEE Transactions on Information Theory*, vol. 42, no. 2, pp. 409-428, March 1996.

[28] P. Robertson, "Illuminating the structure of parallel concatenated recursive systematic (turbo) codes," in *Proc. IEEE GLOBECOM*, San Francisco, CA, pp. 1298-1303, 1994.

[29] A. S. Barbulescu and S. S. Pietrobon, "Interleaver design of turbo codes," IEE Electronics Letters, vol. 30, no. 25, p. 2107, December 1994.

[30] A. S. Barbulescu and S. S. Pietrobon, "On terminating the trellis of turbo codes in the same state," *IEE Electronics Letters*, vol. 31, no. 1, pp. 22-23, January 1995.

[31] W. J. Blackert, E. K. Hall, and S. G. Wilson, "Turbo codes termination and interleaver conditions," IEE Electronics Letters, vol. 31, no. 24, pp. 2082-2084, November 1995.

[32] O. Joerssen and H. Meyr, "Terminating the trellis of turbo codes," IEE Electronics Letters, vol. 30, no. 16, pp. 1285-1286, August 1994.

[33] L. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for

minimizing symbol error rate," *IEEE Transactions on Information Theory*, vol. 20, no. 2, pp. 284-287, March 1974.

[34] J. Hagenauer and P. Hoeher, "A Viterbi algorithm with soft-decision outputs and its applications," *IEEE GLOBECOM'89*, Dallas, TX, pp. 1680-1686, 1989.

[35] W. E. Ryan, "A turbo code tutorial," New Mexico State University, Las Cruces, NM, unpublished paper, http://www.ece.arizona.edu/~wryan/.

[36] L. Hanzo, J. P. Woodard, and P. Robertson, "Turbo decoding and detection for wireless applications," in *Proceedings of the IEEE*, vol. 95, no. 6, pp. 1178-1200, June 2007.

[37] W. Koch and A. Baier, "Optimum and sub-optimum detection of coded data disturbed by time-varying inter-symbols interference," *IEEE GLOBECOM*, pp. 1679-1684, December 1990.

[38] J. A. Erfanian, S. Pasupathy, and G. Gulak, "Reduced complexity symbol detectors with parallel structures for ISI channels," *IEEE Transactions on Communications*, vol. 42, no. 2/3/4, pp. 1661-1670, February/March/April 1994.

[39] P. Robertson, E. Villebrun, and P. Hoeher, "A comparison of optimal and sub-optimal MAP decoding algorithms operating in the log domain," in *Proc. International Conference on Communications*, pp. 1009-1013, June 1995.

[40] J. Hagenauer, E. Offer, and L. Papke, "Iterative decoding of binary block and convolutional codes," *IEEE Transactions on Information Theory*, vol. 42, no. 2, pp. 429-445, March 1996.

[41] B. Vucetic and J. Yuan, *Turbo codes: Principles and Applications*, Boston: Kluwer Academic Publishers, 2000.

[42] L.C. Perez, J. Seghers, and D. J. Costello Jr., "A distance spectrum interpretation of turbo codes," *IEEE Transactions on Information Theory*, vol. 42, no. 6, pp. 1698-1709, November 1996.

[43] J. Seghers, "On the free distance of turbo codes and related product codes," Diploma Project, Swiss Federal Institute of Technology Zurich, 1995.

[44] Y. Kim, J. Cho, and K. Cheun, "Improving the performance of turbo codes by repetition and puncturing," *in Proc. JCCI*, Chungmu, Korea, 2004.

[45] D. N. Rowitch and L. B. Milstein, "On the performance of hybrid FEC/ARQ systems

using rate compatible punctured turbo (RCPT) codes," *IEEE Transactions on Communications*, vol. 48, no. 6, pp. 948-959, June 2000.

[46] J. W. Craig, "A new, simple and exact result for calculating the probability of error for two dimensional signal constellations," in *Proc. IEEE MILCOM*, McLean, VA, pp. 571-575, October 1991.

[47] M. K. Simon and M. S. Alouini, *Digital Communication over Fading Channels: A unified Approach to Performance Analysis*. New York: John Wiley and Sons, 2000.

[48] D. Divsalar, S. Dolinar, and F. Pollara, "Transfer function bounds on the performance of turbo codes", *Telecom and Data Acquisition Progress Report 42-122*, JPL, August 1995.

[49] E. Malkamaki and H. Leib, "Evaluating the performance of convolutional codes over block fading channels," *IEEE Transactions on Information Theory*, vol. 45, no. 5, pp. 1643-1646, July 1999.

[50] J. K. Proakis, *Digital Communications*, 3rd edition. New York: McGraw-Hill, 1995.

[51] H, Stark and J. W. Woods, *Probability, Random Processes, and Estimation Theory for Engineers, 1sst Edition*, Englewood Cliffs, NJ: Prentice Hall, 1986.

[52] I. S. Gradshteyn and I. M. Ryzhik, *Table of Integrals, Series, and Products*. San Diego, CA: Academic Press, 1994.

[53] T. M. N. Ngatched. "Performance of turbo –Coded DS-CDMA systems in fading and burst channels," Masters in Engineering Dissertation, University of Natal, 2001.

[54] R. P. Stanley, *Enumerative Combinatorics*, Monterey, California: Wadsworth & Brooks/Cole, 1986.

[55] J. B. Fraleigh and R. A. Beauregard, *Linear Algebra, 3rd edition*, Addison Wesley, pp. 269-270, 1995.

[56] S. Kallel and C. Leung, "Efficient ARQ schemes with multiple copy decoding," *IEEE Transactions on Communications*, vol. 40, pp. 642-650, March 1992.

[57] A. Viterbi and J. K. Omura, *Principles of Digital Communications Coding*, New York: McGraw-Hill, 1979.

[58] Y. O-C-Mouhamedou, S. Crozier, K. Gracie, P. Guinand, and P. Kabal, "A method for lowering turbo code error flare using correction impulses and repeated decoding," in Proc. *4th International Symposium on Turbo Codes*, April 2006.

[59] Y. O-C-Mouhamedou and S. Crozier, "Improving the error rate performance of turbo codes using the forced symbol method," *IEEE Communications Letters*, vol. 11, no. 7, pp. 616-618, July 2007.

[60] D. J. Costello, Jr. and G. Meyerhans, "Concatenated turbo codes," in Proc. *IEEE International Symposium on Information Theory and its Applications (ISITA'96)*, pp. 571-574, September 1996.

[61] J. D. Andersen, "Turbo codes extended with outer bch code," *Electronic Letters*, vol. 32, pp. 2059-2060, October 1996.

[62] K. R. Narayanan and G. L. Stuber, "Selective serial concatenation of turbo codes," *IEEE Communications Letters*, vol. 1, pp. 136-139, September 1997.

[63] K. R. Narayanan and G. L. Stuber, "List decoding of turbo codes," *IEEE Transactions on Communications*, vol. 46, pp. 754-762, June 1998.

[64] H. Pishro-Nik and Fekri, "Improved decoding algorithms for low-density parity-check codes," in *Proc. 3rd International Symposium on Turo Codes*, pp. 117-120, September 2003.

[65] N. Varnica and M. Fossorier, "Belief-propagation with information correction" Improved near mamimum-likelihood decoding of low-density parity-check codes," in *Proc. IEEE International Symposium on Information Theory (ISIT'04)*, p. 343, June 27- July 2 2004.

[66] E. Papagiannis, M. Ambroze, and M. Tomlinson, "Analysis of non convergent blocks at low and moderate SNR in SCCC turbo schemes," in *Proc. ESA*, Septemeber 2003.

[67] M. Oberg and P. H. Siegel, "Application of distance spectrum analysis to turbo code performance improvement," in *Proc. 35th Annual Allerton Conference on Communications, Control and Computing*, pp. 701-710, September 1997.

[68] E. K. Hall and G. Wilson, "Design and analysis of turbo codes on Rayleigh fading channels," *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 2, pp. 160-174, February 1998.

[69] A. J. Viterbi, "Convolutional codes and their performance in communication systems," IEEE Transactions on Communication Technology, vol. COM-19, no.15, pp. 751-772, October 1971.

[70] 3rd Generation Partnership Project, *3GPP TS 25.212: Multiplexing and Channel coding (FDD)*.

[71] 3rd Generation Partnership Project 2, *Physical Layer Standard for CDMA2000 Spread Spectrum Systems*.

[72] A. J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Transactions on Information Theory*, vol. 13, pp. 260-269, April 1967.

[73] G. D. Forney, "The Viterbi algorithm," Proceedings *of the IEEE*, vol. 61, pp. 218-278, March 1973.

[74] J.M. Mouatcho Moualeu, H. Xu, and F. Takawira, "Cooperative diversity using repeat-punctured turbo codes," Submitted in the *SAIEE journal*.

[75] J.M. Mouatcho Moualeu, H. Xu, and F. Takawira, "Double repeat-punctured turbo coded cooperative diversity scheme," accepted for presentation at the *IEEE Wireless Communications and Networking Conference (WCNC) 2008*, 31March-3April 2008, Las Vegas, Nevada, USA.

[76] J. H. Winters, "Switched diversity with feedback for DPSK mobile radio systems," *IEEE Transactions on Vehicular Technology*, vol. 32, no. 1, pp. 134-150, February 1983.

[77] A. Wittneben, "A new bandwidth efficient transmit diversity antennas modulation diversity scheme for linear digital modulation," in *Proc. International Conference on Communications (ICC'93)*, pp. 1630-1634.