Coding for Insertion/Deletion Channels

by

Feng Wang

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

Approved July 2012 by the
Graduate Supervisory Committee:

Tolga M. Duman, Chair
Cihan Tepedelenlioglu
Martin Reisslein
Junshan Zhang

ARIZONA STATE UNIVERSITY

August 2012

ABSTRACT

Insertion and deletion errors represent an important category of channel impairments. Despite their importance and much work over the years, channels with such impairments are far from being fully understood as they proved to be difficult to analyze.

In this dissertation, a promising coding scheme is investigated over independent and identically distributed (i.i.d.) insertion/deletion channels, i.e., interleaved concatenation of an outer low-density parity-check (LDPC) code with error-correction capabilities and an inner marker code for synchronization purposes. Marker code structures which offer the highest achievable rates are found with standard bit-level synchronization is performed. Then, to exploit the correlations in the likelihoods corresponding to different transmitted bits, a novel symbol-level synchronization algorithm that works on groups of consecutive bits is introduced. Extrinsic information transfer (EXIT) charts are also utilized to analyze the convergence behavior of the receiver, and to design LDPC codes with degree distributions matched to these channels.

The next focus is on segmented deletion channels. It is first shown that such channels are information stable, and hence their channel capacity exists. Several upper and lower bounds are then introduced in an attempt to understand the channel capacity behavior. The asymptotic behavior of the channel capacity is also quantified when the average bit deletion rate is small. Further, maximum-a-posteriori (MAP) based synchronization algorithms are developed and specific LDPC codes are designed to match the channel characteristics.

Finally, in addition to binary substitution errors, coding schemes and the corresponding detection algorithms are also studied for several other mod-

els with synchronization errors, including inter-symbol interference (ISI) channels, channels with multiple transmit/receive elements and multi-user communication systems.

To my dear family.

ACKNOWLEDGEMENTS

It would have been impossible for me to finish my dissertation without the guidance of my committee members, help from friends, and support from my family.

First of all, I would like to express my deepest gratitude to my advisor, Dr. Tolga Duman, for his excellent guidance, patience, and providing me with a great atmosphere for doing research. I am inspired by his insight, and have learned a lot from him. I would also like to thank Dr. Cihan Tepedelenlioglu, Dr. Junshan Zhang, Dr. Martin Reisslein, who not only have served as my committee members, but also have provided me with valuable suggestions to ensure a better quality of this dissertation.

Special thanks to Dr. Dario Fertonani and Dr. Defne Aktas, who have been perfect collaborators and helped me with many technical problems. It was a great pleasure working with them. Many thanks to my former and current colleagues: Dr. Kai Tu, Dr. Uttam Bhat, Jatinder Bajwa Singh, Mojtaba Rahmati, Shahrouz Sharifi, Ahmad El-Moslimany and Ahmad Salim, for the pleasing and inspiring discussions during the past four years. It would have been much less fun without them.

Finally, I would like to dedicate this dissertation to my parents, who have supported me since always, and my beloved wife, Shanshan Wang, who has always been there cheering me up and standing by my side through the good times and bad.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

INTRODUCTION

In this introductory chapter, our objective here is to give a brief overview of the topic and contents of this dissertation. Specifically, we first describe a channel model used to study synchronization errors in communication systems, namely, insertion/deletion channels, in Section 1.1. We then present our contributions and outline of the dissertation in Section 1.2.

1.1 Insertion/Deletion Channels

We consider an important category of channel impairments, i.e., synchronization errors. Such errors are usually caused by the mismatch between the clocks of the transmitters and the receivers or imperfect timing-alignment in a recording system, e.g., in the read/write process of bit-patterned media recording systems [1]. As a result of these type of errors, transmitted symbols can be deleted and random symbols may be inserted into the received data stream, whose positions are unknown to the transmitter and the receiver. The resulting channels are referred to as insertion/deletion channels.

Many practical systems exhibit insertion and deletion errors. For instance, in the context of wireless communications, with the increasing demands for mobile data and voice services, e.g., high-definition (HD) video streaming, video calls on smartphones, the next generation wireless systems are required to provide high-speed yet reliable and secure communication links in various environments, e.g., urban, rural, indoor and outdoor. According to the International Mobile Telecommunications Advanced (IMT-Advanced) requirements on the fourth generation of wireless cellular standards (4G), downlink

speeds of 100 Mbit/s for high mobility users (vehicles) and 1 Gbit/s for low mobility users (pedestrians) are specified. At this level of data rates, perfect synchronization becomes more and more difficult to achieve, and therefore, synchronization errors may occur. A wireless communication system with a variable transmission rate is also a good example of a channel where synchronization errors are common [2]. That is, during the interval when the sampling rate changes, the receiver may lose synchronization for several symbol periods which may lead to a combination of possible insertions and deletions.

Magnetic and optical recording systems are other examples which frequently suffer from synchronization errors. Variations in the rotation speed of the hard disks may cause read and write errors which, for example, were the motivation for considering insertions and deletions in [3] and [4]. Another important example is the recently developed bit-patterned media recording technology [1] aiming to achieve ultra-high recording densities and to replace the conventional film media. In this technology, the recording media is pre-patterned into small "islands". For each island, a time window is assigned in which the writing process can be completed successfully. This brings about the write synchronization problem as a new design issue compared to the conventional media recording. As a result, written-in errors may occur due to several reasons; for instance, they may be caused by the imperfect synchronization of the write head to the bit positions, fluctuations of the bit positions and the switching field, and so on. Another problem due to mis-synchronization is that when writing to an island occurs outside the specific time window, either the previous bit is overwritten or the current bit is skipped. The writing process can therefore be viewed as a recording channel with insertion and deletion errors. Since it is hard to adjust the write head to each island perfectly,

the insertion and deletion errors cannot be avoided, thus good error correcting coding schemes are needed. Also, we note that for these channels, inter-symbol interference (ISI) and additive white Gaussian noise (AWGN) are also present along with the insertions and deletions.

Channels that experience insertion and deletion errors may also be present in various other scenarios, including transmission over a serial line (the clock speed of the transmitter may not be accurately known which leads to an unknown time of arrival for each bit), and as typos on letters and documents, i.e., when there is a missing letter or extra letters, which are the motivations of studying non-standard communication channels in [5].

Despite of the broad applications and the importance of this channel model, insertion/deletion channels are still far from being fully understood. For instance, tight upper and lower bounds on their channel capacity are only available for deletion channels with deletion rate close to zero. In this dissertation, our main objective is to contribute towards an understanding of these channels, specifically, by considering design suitable practical coding schemes. Our main contributions are summarized in the next section, and an outline of the dissertation is given.

1.2 Outline of the Dissertation and Contributions

We review existing results on insertion/deletion channels in Chapter 2. We first describe different channel models proposed in the literature. Then, we review the existing results on the channel capacity and coding schemes for insertion/deletion channels. Furthermore, we give a very brief introduction to channel coding. Specifically, we focus on low density parity check (LDPC) codes and the corresponding iterative decoding algorithm. We also present some

basic ideas of multiple-input multiple-output (MIMO) systems and multi-user communication systems, since synchronization errors can be considered in the context of these systems as well.

In Chapter 3, we consider binary channels impaired by independent and identically distributed (i.i.d.) insertion, deletion, and substitution errors, whose positions are unknown to either the transmitter or the receiver. As in [6], we consider the interleaved serial concatenation of an outer error-correcting code with an inner marker code. To limit the decoding latency, we assume that the required marker code-based synchronization is performed only once per received packet, i.e., iterations with the outer decoder are not allowed. Our first contribution consists of the evaluation of highest rates at which reliable communications (in the Shannon sense [7]) is possible, for a given channel and a given marker code. An approximate solution of this problem was proposed in [6], where the authors characterized the capacity of a proper time-varying binary symmetric channel (BSC) and conjectured that it gives an accurate approximation of the actual achievable rate. In our work, we consider the exact solution to the problem based on mutual information arguments, i.e., we numerically evaluate the information rates, and show how to exploit these achievable rate analyses to optimize the marker code structure. As the marker code-based synchronization algorithm, we first consider the standard maximum-a-posteriori (MAP) detector working at the bit level [8, 9]. Then, we introduce the MAP detection at the symbol level, defining a symbol as a group of m consecutive bits, and demonstrate how this approach can improve the achievable rates without changing the transmitter structure.

In addition to the achievable rate analysis with a single pass decoder, we also investigate the outer LDPC code design when multiple-pass decoding is

used, i.e., synchronization is performed multiple times for each received packet by iterating between the decoder for the outer code and the synchronization module. The goal of our design is to find "good" outer LDPC codes concatenated with marker codes for transmission over the insertion/deletion channels which offer better decoding performance compared to the ones optimized for AWGN channels by varying the variable/check node degree distributions. The motivation is explained as follows: since the detection extrinsic information transfer (EXIT) chart [10] is not flat for the channel model under consideration, LDPC codes designed for AWGN channels are no longer optimal when iterative decoding is performed [11]. Many optimization schemes for LDPC codes over different types of channels have been reported in the literature, e.g., density evolution techniques [12], EXIT chart-based designs [10, 13, 14]. Here, we utilize the EXIT charts to analyze the impact of insertions and deletions on the convergence behavior of the receiver and find good variable/check node degree distributions.

In Chapter 4, we aim at the development of both information theoretic results and practical coding schemes for the segmented deletion channel. Compared to channels with i.i.d. deletions, where each bit is independently deleted with an equal probability, the segmentation assumption imposes certain constraints, i.e., in a block of bits of a certain length, only a limited number of deletions are allowed to occur. In particular, we consider the elementary segmented deletion channel, i.e., no more than one deletion per segment is allowed. We first show that the segmented deletion channels fall within the framework of memoryless synchronization error channels (with non-binary inputs), and by a proper application of Dobrushin's results [15], we argue that the channel is information stable. Then, we explore several upper and lower bounds on

the capacity of these channels by providing the transmitter and the receiver with genie-aided information, i.e., about which segment has a deletion error. We also show that when the average bit deletion rate is small, asymptotic behavior of the capacity can be characterized by utilizing the methodology developed in [16] to the new model. As a result, a good approximation of the channel capacity for small deletion probabilities is obtained. Also, we illustrate that the derived upper and lower bounds behave similarly for some range of deletion probabilities, while a wide-range of deletion probabilities exist where improvement of the results is clearly possible.

In addition to the capacity characterization of the segmented deletion channel, we also consider a practical concatenated coding approach, for which as in Chapter 3, concatenation of an outer LDPC code for error-correcting purposes with an inner marker code, which provides re-synchronization capabilities, is explored. Despite the similar encoding procedure (with the i.i.d. insertion/deletion channel case), there are significant differences at the receiver. In particular, the soft-output synchronization algorithm is no longer optimal. Therefore, we introduce bit-level and symbol-level MAP detection algorithms which incorporate the segmentation assumption for improved results. Our approach is motivated by the fact that if we allow for the use of powerful codes with strong error-correcting capabilities, a much higher code rate (compared to the ones reported in [17]) can be achieved with a low probability of error (when we drop the requirement of no errors).

We deal with the case of insertion/deletion channels with ISI in Chapter 5. Following a similar encoding/decoding procedure to the ones in Chapters 3 and 4, we design a MAP detection algorithm at the bit level based on a modification of the trellis diagram used in [18]. The derived channel detection

algorithm jointly achieves equalization for the ISI channel and synchronization for the insertion/deletion channel. Furthermore, as an alternative to the joint MAP detection, we introduce several low-complexity solutions. We investigate the separate channel detection scheme, i.e., the concatenation of an equalizer for the ISI channel and a MAP detector for synchronization purposes. Then, we discuss other simplified solutions to the problem by employing the well-known M- [19] and T- algorithms [20] as simplifications of the full complexity forward backward algorithm [8, 9]. We also design a soft-input soft-output stack decoding algorithm [21–23] which utilizes a code-tree instead of the trellis representation. We show that these three approaches greatly reduce the decoding complexity, especially for channels with long memory or for high insertion/deletion rates, at the expense of reduced decoding performance.

Chapter 6 is devoted to multi-element/multi-user communication systems with deletion errors. We describe two new channel models suitable in certain applications. The first one, namely the MIMO deletion channel, models the scenarios where multiple transmitters and receivers suffering from synchronization errors are employed, e.g., multi-track bit patterned media recording systems. The second one, which is referred to as Gaussian multiple access channel (MAC) with deletion errors, models a scenario where multiple transmitters communicate to the same receiver simultaneously, while each of them independently suffering from deletion errors. Some potential applications of this model can be found in the context of asynchronous wireless sensor networks [24]. For these two channel models, we consider a coding scheme based on a serial concatenation of an LDPC code, a marker code and a layered space-time code (for the MIMO deletion channel case), and design suitable detectors and iterative decoding schemes operating at the bit level which jointly achieve

synchronization for the deletion channel and detection for the MIMO and Gaussian multiple access channels. Utilizing the proposed detector together with the outer LDPC code, we demonstrate that reliable transmission over these channels is feasible.

In the last chapter of the dissertation, we summarize our results and present our contributions. We also provide several possible future research directions in the context of insertion/deletion channels.

Chapter 2

REVIEW OF EXISTING RESULTS ON INSERTION/DELETION CHANNELS

In this chapter, our objective is to review some essential material related to the insertion/deletion channels in order to provide the necessary background for the rest of the dissertation and to put our contributions in a proper context. Specifically, we begin with an introduction of insertion/deletion channel models. Then, in Sections 2.2 and 2.3, we review some important existing results on their channel capacity and practical coding schemes, respectively, followed by a general discussion of channel coding techniques in Section 2.4. In the last section, we provide a brief introduction to MIMO channels and multi-user communication systems which will be used in the later chapters of the dissertation.

2.1 Insertion/Deletion Channel Models

Many models for insertion/deletion channels are proposed and studied in the literature. In [9], a binary synchronization error channel is characterized by three parameters: P_i, P_d and P_s, which indicate the probability of insertion, deletion and substitution events as a result of the channel impairments, respectively. Insertion, deletion and substitution errors are described by the state diagram shown in Fig. 2.1. In this model, the number of consecutive insertions between any two bits can be arbitrary large. In practice, it may be more suitable to limit the maximum number if consecutive insertions to a value I because P_i is relatively small in a practical system, and therefore, the probability of having more than I consecutive insertions between two consecutive bits may be negligible. Also, each transmitted bit independently gets deleted

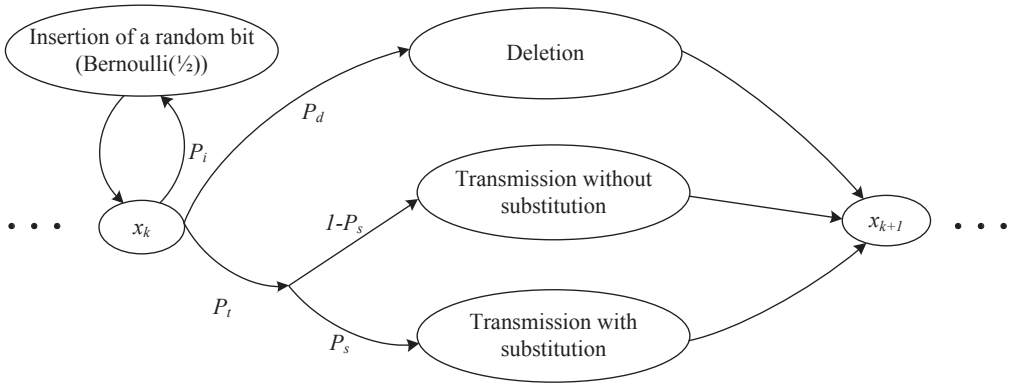

Figure 2.1: Mackay's insertion/deletion channel model.

with probability P_d. Therefore, a transmission event occurs with probability of $P_t = 1 - P_i - P_d$. The factor P_s indicates the probability of x_k suffering from a substitution error provided that a transmission event happens.

Zigangirov considers a very similar channel model in [25] where the only difference is in the definition of the insertion probability: the probability of no insertion is q_1, and number of insertions being i has a probability of $q_1 p_1^i$. Since the summation of all possible event probabilities equals one, i.e., $\sum_{i=0}^{\infty} q_1 p_1^i = 1$, we have $q_1 + p_1 = 1$, which indicates that p_1 is the probability of having at least one insertion. As for the deletion errors, it is assumed that each transmitted bit[1] is deleted with probability p_2, and the probability of no deletion is q_2 so that $q_2 + p_2 = 1$.

Another important model is introduced in [22] by Gallager. Specifically, each input bit independently gets deleted (with probability P_d), or gets replaced by two uniformly distributed random bits (with probability P_i), or is correctly received (with probability P_c), or incorrectly received[2] (with probability P_e), where $P_i + P_d + P_c + P_e = 1$. An important difference from the

[1]The inserted bits cannot be deleted according to the model.
[2]This is due to the presence of substitution errors.

previous two insertion/deletion channel models is that for each transmitted bit, it cannot experience an insertion event and a deletion event together.

The sticky channel, whose capacity is studied in [26], is a particular type of insertion channel. In this channel, the transmitted symbols can be independently repeated several times at the receiver where the number of duplications is random and follows some fixed distribution. Typing on a keyboard maybe an example of a sticky channel. When a certain key is held too long, the letter typed may be printed multiple times although only one occurrence should be present at the output.

In [17], a different class of synchronization errors are considered, i.e., binary insertion/deletion channels with the additional segmentation assumption. According to the segmentation assumption, several consecutively transmitted bits are considered as one block or segment and the number of insertions/deletions within each segment is limited to a certain number. Motivation of studying this channel model is that the segmentation assumption appears naturally in many practical systems. For instance, consider a bit-patterned media recording system where cycle slips are caused due to the mis-alignment between the write-head and pre-patterned magnetic islands [1]. In this case, when a bit is skipped or written multiple times on one island, the next deletion/insertion event will appear only after some number of bits. Another point in using the segmented deletion channel model is that this model can incorporate various types of deletion errors as special cases. When we choose the length of a segment to be one, the channel becomes an i.i.d. deletion channel. If the segmentation assumption requires that all the bits from the segment get deleted whenever an error occurs, the channel becomes a bursty deletion channel, which can model a variable transmission rate wireless communication

system [27], when the receiver lose synchronization for several symbol periods during the interval of changing sampling rate. More precisely, in this model, for a binary input and binary output channel, the transmitted bit sequence is implicitly partitioned into N consecutive disjoint blocks $\{\mathbf{X}_n\}_{n=1}^{N}$ each with the same length of b bits. Note that there is no explicit partitioning at the transmitter side, however, the receiver is aware of the restriction. During the transmission, a total number of at most d_0 insertions and deletions are allowed to happen for each $\mathbf{X}_n$, resulting in a received vector of varying lengths. For instance, if we utilize the insertion model in [22], the length of the received vector corresponding to $\mathbf{X}_n$ takes values in $\{b-d_0,\ldots,b,\ldots,b+d_0\}$, and the positions of insertion/deletion errors are uniformly chosen within the segment. In addition to the synchronization errors, substitution errors can also be incorporated [9, 22], i.e., every undeleted user bit maybe incorrectly received with a certain probability. We also note that there is no special marker between the bits of different segments, hence the receiver does not know the segment boundaries.

In [17], the authors focus on a particular case, namely, the elementary segmented deletion channel, i.e., the segment $\mathbf{X}_n$ is received intact with probability $1-P_d$ while only one bit is deleted with probability P_d. Also, the deletion events for each segment are independent. A simple example is given as follows. Assume that the binary sequence 00101101 is transmitted over a segmented deletion channel with $b=4$, it is possible that the third and fifth bits are deleted, leading to a received sequence of 000101. However, receiving 001001 is impossible as in this case two bits from the second segment would need to be deleted, which is not allowed.

In addition to the above described channel models, it should be noted that the insertion/deletion errors can also be incorporated with other channel models and impairments, such as in the presence of AWGN, binary erasure channels (BECs), channels with ISI, MIMO channels, multiple access channels, etc. Some of these models will be considered further in the subsequent chapters.

2.2 Upper/Lower Capacity Bounds on Insertion/Deletion Channels

In this section, we review some existing results on the capacity of insertion/deletion channels. Since the positions of insertions/deletions are unknown to the transmitter and receiver, study of insertion/deletion channels from an information theoretic or a practical coding point of view is remarkably difficult. For i.i.d. insertion/deletion channels, the information stability and the Shannon's capacity theorem are proved in [15]. However, a finite-letter expression of the channel capacity does not exist and only upper/lower bounds on this quantity are available in the literature. Furthermore, for most models, these bounds are not tight for almost any range of insertion/deletion probabilities [28–30].

Performance of channels with insertions, deletions and flipping errors by adding a pseudorandom sequence to the convolutionally coded bits to combat the synchronization errors is investigated by Gallager in [22]. A lower bound on their capacity is then derived which is given by

$$C \geq 1 + P_i \log_2 P_i + P_d \log_2 P_d + P_e \log_2 P_e + P_c \log_2 P_c, \tag{2.1}$$

where P_i, P_d, P_e and P_c are defined in Section 2.1 when explaining Gallager's model. We stress that this bound is only applicable to Gallager's insertion/deletion channel model and is loose for large P_i, P_d values.

Combinatorial upper and lower bounds on the capacity of binary insertion and deletion channels when the number of errors in a codeword block is asymptotically a fraction of the block size are given in [31]. The resulting upper and lower bounds are given by

$$1-p\log_2\left(e^2\left(\frac{3}{2p}+\frac{15}{16}\right)\left(\frac{3}{2p}+\frac{47}{16}\right)\right) \leq C \leq 1-(1+p)\log_2(1+p)+p\log_2(2p), \tag{2.2}$$

where p is the fraction of insertion/deletion errors when the block length goes to infinity. These bounds are on the zero-error capacity and they do not apply for the Shannon capacity of these channels.

For the deletion channel, Diggavi and Grossglauser [32] consider an achievable rate which serves as a lower bound on the capacity of deletion channels. It is shown that with i.i.d. codebooks, the achievable rate of deletion channels differs from that of erasure channels by at most $H(P_d) - P_d \log_2 \frac{K}{K-1}$ for $P_d < 1 - K^{-1}$, where K is the alphabet size and $H(\cdot)$ is the binary entropy function. This difference can be further narrowed by considering Markovian codebooks. Recent works by Drinea and Mizenmacher [33, 34] improve the bounds in [32], and extend their results to channels with duplication errors as well. The best reported upper bounds for i.i.d. binary deletion channels [29] are obtained by providing the transmitter and receiver with genie-aided information to make the channel memoryless and using the Blahut-Arimoto algorithm [35, 36] in a suitable way. Furthermore, in [30], the authors extend their work to compute several upper and lower bounds on the capacity of channels with insertion, deletion and substitution errors as well.

In two recent papers [16, 37], asymptotic behavior of the capacity for the binary i.i.d. deletion channel is characterized for small deletion rates. These papers use different methodologies but reach similar conclusions, i.e.,

in [37], it is shown that the capacity lower bound $C \geq 1 - H(P_d)$ is tight as P_d approaches zero, while in [16], under the same condition, the channel capacity is quantified as a series expansion of the deletion rate P_d, which for any $\epsilon > 0$ is given as $C = 1 + P_d \log_2 P_d - (\log_2 2e - \sum_{l=1}^{\infty} 2^{-l-1} l \log_2 l) P_d + O(P_d^{3/2-\epsilon})$. As an extension work of [16], in [38], the coefficient for P_d^2 is also explicitly given. The authors in [39] specify a memoryless synchronization error channel by a stochastic transition probability matrix, and obtain analytical lower bounds on the capacity for channels with deletions or duplications only, some of which are expected to be tight for small deletion or duplication probabilities.

2.3 Practical Coding Schemes of Insertion/Deletion Channels

The difficulty in the study of i.i.d. insertion/deletion channels is also confirmed by the lack of channel codes able to provide reliable communications at rates close to the capacity lower bounds [28]. An early approach for identifying synchronization errors is the use of markers referring to the insertion of known bits at pre-specified positions into the transmitted stream [40], so that synchronization can be re-gained by locating the markers in the received sequence. Apart from these, there are three basic approaches toward designing codes for channels with synchronization errors: algebraic/number theoretic code design, convolutional (trellis-based) codes, channel codes based on code concatenation.

A comprehensive survey of works on single-deletion-correcting codes is provided in [41] primarily focusing on the algebraic/number theoretic approaches. It becomes evident that even the restriction on the number of possible insertions/deletions to be only one over a codeword does not offer simple solutions. Codes with multiple insertion and/or deletion correcting capabil-

ities are described in [42]. The use of Reed-Muller $(1, m)$ code is discussed in [43], where in addition to substitution errors, the channel model permits either the repetition or the deletion of a single bit. Non-binary codes over deletion channels, e.g., Reed-Solomon (RS) codes, are studied in [44], where for codeword lengths $l \leq 36$, RS codes capable of correcting up to $l - 3$ deletions are provided. Further, algebraic designs employing cyclic codes are reported in [45, 46].

As an example of the second approach, i.e., convolutional coding over insertion/deletion channels, we can cite pruned convolutional codes in [47]. [48] considers convolutional codes with a long buffer and bit reversal before transmission. In [49], new states in the trellis of a convolutional code are added to accommodate synchronization errors and a special code construction algorithm is introduced which maximizes the minimum Levenshtein distance [50] between different codewords. Idea of adopting parallel Viterbi decoders to correct insertion, deletion and flipping errors is investigated in [51], where the key is to ensure that the decoding procedure starts from the correctly synchronized decoder.

To date, coding schemes with the most promising performance reported over i.i.d. insertion/deletion channels are based on code concatenation. The key idea is to concatenate, through an interleaver, an outer code with good error-correction capabilities with an inner code whose aim is to help the receiver detect synchronization errors due to the presence of insertions/deletions. For example, [52] considers concatenation of Reed-Solomon codes as outer codes with an inner code designed using a brute force approach. The paper shows that these codes are asymptotically good for channels with insertions and deletions in the sense that the code rate remains positive. Concatenation

of outer LDPC codes and inner watermark codes are investigated in [9], while concatenation of LDPC codes and marker codes are studied in [5,6]. Synchronization can be achieved by means of the forward-backward algorithm presented in [8,9] with the help of information carried by the marker/watermark codes. A sub-optimal decoding strategy for concatenation of an LDPC code, a Varshamov-Tenengolts (VT) code and a marker code is considered in [53].

In addition to the i.i.d. insertion/deletion channels, of interest in this dissertation are the segmented deletion channels as well. For these channels, there is very little work on suitable channel coding schemes over these channels, and most of the code designs for i.i.d. insertion/deletion channels cannot be directly applied, e.g., those in [41,44,47,54]. The only existing coding approach for this channel is given in [17], where the proposed codes can correct all the insertions and deletions with no errors when only a single insertion/deletion error per segment is allowed. The key idea is to encode the data sequence so that each segment is a codeword from a 1-deletion/insertion correcting code. Other constraints are also enforced on the codewords which allow for a simple left-to-right, segment by segment decoding. As an example, a codebook containing 12 codewords is found for an elementary segmented deletion channel with a segment size of $b = 8$, resulting in an overall code rate of $R = 0.448$. Higher code rates can be achieved for larger b. Although some extensions have also been studied offering increased code rates, these coding algorithms require some check bits and check sums to be known at the receiver side leading to the need of a perfect side-information channel [17].

It is shown that when the transmission rate is less than the channel capacity, reliable communication is possible even for channels with synchronization errors [15, 55]. In light of this conclusion, it is also desired to design "good" codes which can correct possible insertions/deletions as well as other types of errors.

Theoretically speaking, a randomly chosen code with length n approaching infinity is a good code with a high probability, as known from the usual random coding argument in information theory[3]. However, the corresponding encoding and decoding complexity grows exponentially in n, and thus it becomes infeasible in practice. This problem is solved by the invention of turbo codes and the re-discovery of LDPC codes [56, 57] in the 1990's, both of which offer performance within a fraction of a decibel (dB) of the Shannon limit over many channels of practical interests and have a practical encoding and decoding algorithms. Both of two codes are inspired by the idea of random coding, and offer excellent performance when very large block sizes are used, e.g., $n > 10000$. In the following sub-section, we will introduce some basic concepts and corresponding decoding algorithms for LDPC codes, since in the rest of the dissertation we will heavily refer to them.

2.4.1 Low Density Parity Check Codes

The LDPC codes were first introduced by Gallager in his doctoral dissertation and then were re-discovered by Mackay and Neal [57]. Considering only the binary LDPC codes, where all operations are carried out in the binary field

[3]This claim is somewhat weaker for channels with synchronization errors as we do not know the optimal input distributions in general.

$GF(2)$, a regular LDPC code is a linear block code defined by an $m \times n$ parity check matrix $\mathbf{H}$ with w_c ones in every column and w_r ones in every row. The term "low density" means that $\mathbf{H}$ is sparse, i.e., $w_c \ll m$ and $w_r \ll n$. As a generalization, for irregular LDPC codes, the row weights and/or column weights of $\mathbf{H}$ are not constants. Usually irregular LDPC codes offer a better error correcting performance than the regular ones with the same rate [57].

An LDPC code (or any linear block code) can be represented by a Tanner graph. The Tanner graph is a bipartite graph in which the nodes can be partitioned into two classes, and no edge connects two nodes from the same class. In the first class of nodes, there is one node for each of the n bits in the codeword, which are often referred to as the variable nodes. In the second class of nodes, there is one node for each of the m parity checks equations, which are often referred to as the check nodes. An edge connects the i-th variable node c_i to the j-th check node f_j if and only if (iff) the bit is included in the parity check, i.e., the ij-th entry in $\mathbf{H}$ equals 1. As an example, the Tanner graph of the linear block code with the following parity check matrix

$$\mathbf{H} = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \end{bmatrix},$$

is shown in Fig. 2.2.

2.4.1.1 Message Passing Algorithm

In the very first paper on LDPC codes [56], Gallager provides two decoding algorithms (one of which is only applicable to BSC) that is near optimal.

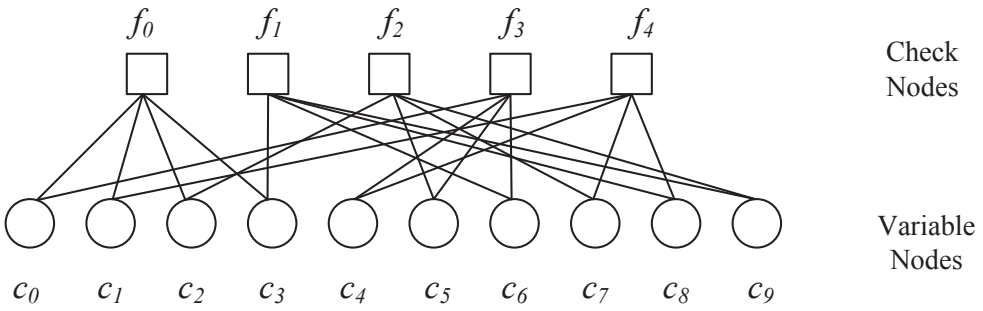

Figure 2.2: Tanner Graph for the example LDPC code.

Later on, other researchers [57] have independently discovered a graph-based iterative decoding algorithm which offers better decoding results. This algorithm comes with different names under various circumstances, including: the sum-product algorithm (SPA), the belief-propagation algorithm (BPA), and the message passing algorithm (MPA).

The objective of SPA is to find a valid codeword $\mathbf{x}$ with $\mathbf{Hx} = \mathbf{0}$. The main idea is that since multiple coded bits are involved in one check equation, the sum of them should equal to one, therefore, an updated information on one particular bit can be generated by utilizing the a-priori information from other bits belonging to the same check equation. As one bit is usually involved in several check equations, soft information can be iteratively exchanged between the variable and check nodes which converge to a final decision on all the coded bits. Before describing the detailed decoding algorithm, we start with some notation:

- p_i: $p_i = P(c_i = 1|y_i)$, where y_i is the channel output corresponding to the bit c_i.
- Q_i: output a-posteriori probability for the variable node c_i,
- q_{ij}: information from the ith variable node to the jth check node, where

$q_{ij}(x)$ equals the probability that $c_i = x$ given y_i and all the extrinsic information passed to c_i from all the check nodes except except f_j,

- r_{ji}: information from the jth check node to the ith variable node, where $r_{ji}(x)$ equals the probability that parity check equation f_j is satisfied, given $c_i = x$ and the other information passed to f_j (except c_i),
- R_j: the set of column indices of the ones in the j-th row,
- $R_{j\backslash i}$: the set of column indices of the ones in the j-th row excluding c_i,
- C_i: the set of row indices of the ones in the i-th column,
- $C_{i\backslash j}$: the set of row indices of the ones in the i-th column excluding f_j.

We the above notation, the message passing algorithm can be written into five steps. The detailed derivation is omitted and can be found in [57].

1. Initialize
$$q_{ij}(0) = 1 - p_i,$$
$$q_{ij}(1) = p_i.$$
2. For the first half of one iteration, at the check node f_j, the message passed to c_i is calculated as (one example is given in the left sub-figure of Fig. 2.3)
$$r_{ji}(0) = \frac{1}{2} + \frac{1}{2} \prod_{i' \in R_{j\backslash i}} (1 - 2q_{i'j}(1)),$$
$$r_{ji}(1) = 1 - r_{ji}(0).$$
3. For the second half of one iteration, at the variable node c_i, the message passed to f_j is calculated as (one example is given in the right sub-figure

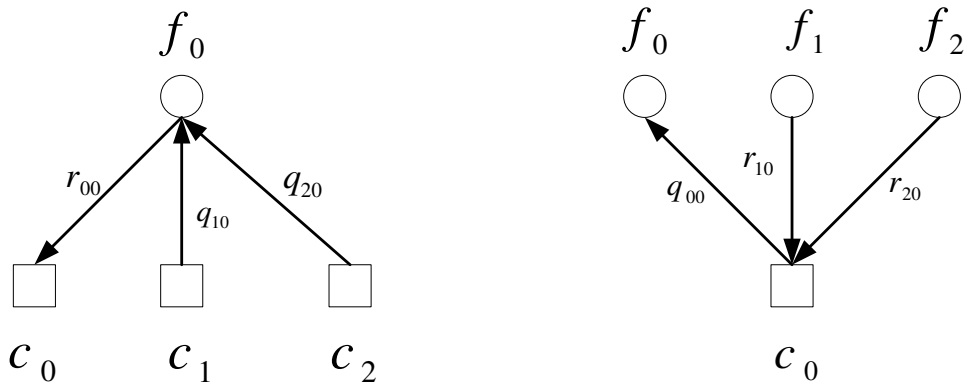

Figure 2.3: Message passing algorithm at variable/check node.

of Fig. 2.3)

$$q_{ij}(0) = K_{ij}(1-p_i) \prod_{j' \in C_{i\backslash j}} r_{j'i}(0),$$

$$q_{ij}(1) = K_{ij} p_i \prod_{j' \in C_{i\backslash j}} r_{j'i}(1),$$

where K_{ij} is selected so that $q_{ij}(0) + q_{ij}(1) = 1$.

4. The information q_{ij} and r_{ji} are exchanged between the variable and the check nodes until the maximum of iterations is reached and the final soft decisions are generated as

$$Q_i(0) = K_i(1-p_i) \prod_{j \in C_i} r_{ji}(0),$$

$$Q_i(1) = K_i p_i \prod_{j \in C_i} r_{ji}(1),$$

where K_i is selected so that $Q_i(0) + Q_i(1) = 1$.

5. Hard decisions are made using

$$c_i = \begin{cases} 1 & \text{if } Q_i(1) > 0.5, \\ 0 & \text{if } Q_i(0) > 0.5. \end{cases}$$

Message passing algorithm is often carried out using the logarithms of the ratios of probabilities to ensure numerical stability. We define the following log-likelihood ratios

$$L(c_i) = \log\left(\frac{1-p_i}{p_i}\right),$$
$$L(q_{ij}) = \log\left(\frac{q_{ij}(0)}{q_{ij}(1)}\right),$$
$$L(r_{ji}) = \log\left(\frac{r_{ji}(0)}{r_{ji}(1)}\right),$$
$$L(Q_i) = \log\left(\frac{Q_i(0)}{Q_i(1)}\right).$$

In this case, the computations in steps 2-4 above can be rewritten as

$$L(r_{ji}) = \prod_{i' \in R_{j\backslash i}} \alpha_{i'j} \cdot \left[\sum_{i' \in R_{j\backslash i}} \phi(\beta_{i'j})\right],$$
$$L(q_{ij}) = L(c_i) + \sum_{j' \in C_{i\backslash j}} L(r_{j'i}),$$
$$L(Q_i) = L(c_i) + \sum_{j \in C_i} L(r_{ji}),$$

where $\alpha_{ij} = \text{sign}[L(q_{ij})]$ and $\beta_{ij} = \text{abs}[L(q_{ij})]$, and the hard decisions are made using

$$c_i = \begin{cases} 1 & \text{if } L(Q_i) < 0, \\ 0 & \text{otherwise.} \end{cases}$$

The information update in the check node, i.e., obtaining $L(r_{ji})$, can be time-consuming in the SPA. Therefore, several low-complexity solutions are proposed at the cost of some performance degradation. Popular choices include the Min-Sum decoder and the Min-Sum-Plus-Correction-Factor decoder [58]. Details of these algorithms are omitted.

2.5 Multi-Antenna and Multi-User Communication Systems with Synchronization Errors

In wireless communications, multiple users often share the same medium. For instance, multiple transmitters may communicate to the same receiver over a multiple access channel. As a specific example, in the uplink of a cellular network, several mobile users may transmit data to the same base station simultaneously utilizing some kind of multiple access techniques, e.g., time-division multiple access (TDMA) or code-division multiple access (CDMA). In order to guarantee a reasonable performance over a multi-user communication system, in most cases, it is required that perfect synchronization is achieved among all the users. However, this is a strong assumption for certain types of applications.

One example scenario is given as follows. Motivated by the needs of environmental detection, military surveillance, health monitoring, etc., the idea of distributed sensing has been proposed, e.g., see [59]. The goal is to deploy a large number of cheap wireless nodes in the place of interest to sense some ongoing process and allow them to communicate with each other and also with wired access points through a multiple access channel. Since these sensor nodes are low-cost and with limited power, sometimes a low-cost yet inadequate timing recovery block is employed on the sensors resulting in a wireless multi-user communication system with a varying sampling rate [27]. During the interval of changing the sampling rate, insertion/deletion errors may occur, hence, when several sensors are communicating to one or more receivers, each transmitted signal may experience insertion/deletion errors.

In addition to multi-user communication systems, synchronization errors may be present among different transmitters used at each antenna and the receiving elements. As an example, consider a multi-track bit-patterned media recording system [1, 60]. As briefly discussed in Chapter 1, insertion/deletion errors occur when recoding the information bits onto the existing magnetic islands. During the writing and reading processes, for a multi-track system, there are more than one head working on several tracks simultaneously, yielding a MIMO channel. Therefore, treating the original data as the channel input and the read-out information as the channel output, this particular storage system can be viewed as the cascade of an insertion/deletion channel and a MIMO channel.

So far, no explicit channel models or corresponding coding/decoding schemes have been proposed for MIMO channels and multiple access channels with synchronization errors. In this dissertation, some approaches to modeling the two types of channels are developed and initial detection/decoding solutions are provided.

2.6 Chapter Summary

In this chapter, we first reviewed several insertion/deletion channel models introduced in the literature. Then, from both an information theoretic and a practical coding point of view, we gave a detailed survey of existing results on insertion/deletion channels. Furthermore, we presented a summary of channel coding techniques and LDPC codes in particular. We also described some basic ideas on MIMO systems, multi-user communication systems and motivate the need for models incorporating synchronization errors in these contexts.

Chapter 3

MARKER CODES CONCATENATED WITH LDPC CODES OVER INSERTION AND DELETION CHANNELS

In this chapter, our aim is to design a practical coding scheme for i.i.d. insertion/deletion channels with substitution errors. We start with a description of the system model in Section 3.1. In Section 3.2, we review the standard bit-level MAP detection algorithm and numerically evaluate the ultimate rate achievable by interleaved concatenated coding schemes. In Section 3.3, we introduce a novel symbol-level MAP detection algorithm and compare the relevant achievable rates with those characterizing the standard bit-level approach. Error-rate results for a practical LDPC-coded scheme are also reported for both bit-level and symbol-level detection algorithms. An EXIT chart-based LDPC code design process for the insertion and deletion channels is provided in Section 3.4 along with some example designs. Finally, a summary is provided in Section 3.5.

3.1 System Description

We consider transmission over binary channels impaired by insertion, deletion, and substitution errors, according to the Gallager's model described in Chapter 2. That is each input bit independently gets deleted (with probability P_d), or gets replaced by two uniformly distributed random bits (with probability P_i), or is correctly transmitted (with probability $P_t(1-P_s)$), where $P_t = 1 - P_d - P_i$, and P_s denotes the flipping probability.

Let $\mathbf{x}_1^T = \{x_k\}_{k=1}^T$ and $\mathbf{y}_1^R = \{y_n\}_{n=1}^R$ be the sequences of bits at the channel input and channel output, respectively, where the number T of trans-

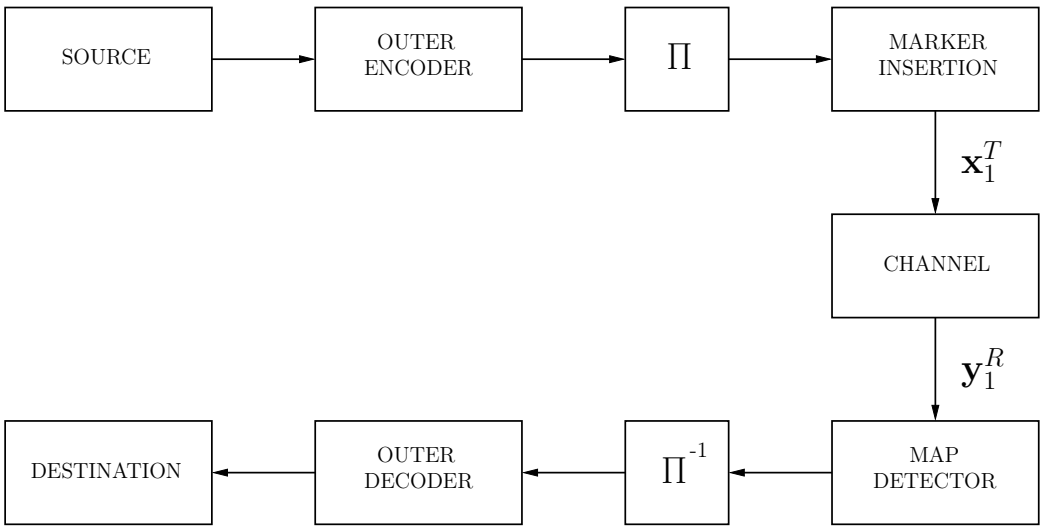

Figure 3.1: Block diagram of the considered concatenated coding scheme. Interleaving and deinterleaving blocks are denoted by Π and Π^{-1}, respectively.

mitted bits is a constant system parameter while the number R of received bits is a random variable depending on the realization of the insertion/deletion process.

We adopt the coding scheme depicted in Fig. 3.1, which consists of the interleaved serial concatenation of an outer error-correcting code with an inner marker code. Specifically, the information bits are first encoded by means of a powerful channel code (e.g., a turbo or an LDPC code), then the transmitted sequence is formed by inserting pilot bits, which are often referred to as markers, to the interleaved sequence of coded bits. The marker bits and their positions in the transmitted sequence are known to the receiver, which exploits this information using a MAP detector to recover the synchronization errors due to insertions/deletions, as explained later. For simplicity, we only focus on the case of regular marker codes with rate

$$r_M = \frac{N_C}{N_C + N_M} \ ,$$

i.e., the case when the same marker consisting of N_M consecutive bits is in-

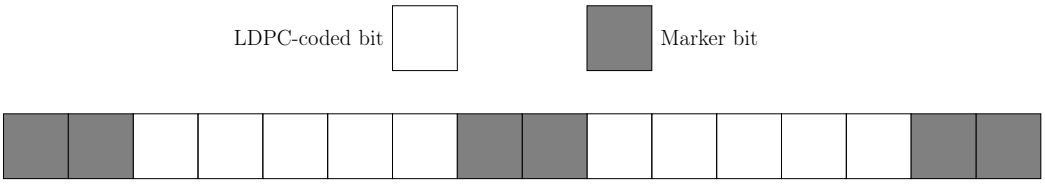

Figure 3.2: Example of a marker code with $N_M = 2$ and $N_C = 5$.

serted every N_C bits at the output of the outer encoder. Hence, if the outer code rate is denoted by r_C, the overall code rate is $r = r_C r_M$. We notice that the same coding scheme was considered in [6], while a similar scheme adopting watermark codes instead of marker codes was considered in [9]. A simple illustration of a marker code is given in Fig. 3.2.

At the receiver side, given the *a priori* log-likelihood ratios (LLRs or L-values) $\log \frac{P(x_k=0)}{P(x_k=1)}$, the MAP detection is first executed to generate the conditional probability $\xi_k(x_k) = P(\mathbf{y}_1^R|x_k)$ for $k \in \{1, 2, \ldots, T\}$ and $x_k \in \{0, 1\}$ by exploiting the perfect *a priori* information from the marker code. Then the extrinsic information [61] on the transmitted bits can be easily obtained as $\log \frac{P(\mathbf{y}_1^R|x_k=0)}{P(\mathbf{y}_1^R|x_k=1)} = \log \frac{\xi_k(0)}{\xi_k(1)}$. After being deinterleaved, the *a posteriori* information, i.e., the sum of *a priori* and extrinsic L-values, feeds the outer decoder, which finally generates an estimate of the information bits. We point out that decoding performance can be improved by adopting iterative schemes based on the exchange of extrinsic information between the MAP detector and the outer decoder. However, since the MAP detector is typically the bottleneck of the receiver in terms of latency, we assume that the MAP detection is executed only once in Sections 3.2 and 3.3. Iterative detection/decoding is considered in Section 3.4 where specific outer code designs are pursued.

3.2 Bit-Level Synchronization

Let us first review the bit-level MAP detection algorithm for the considered channel model. The algorithm, which already appeared in [8, 9] with some differences in the channel model, is similar to the general forward backward algorithm (FBA) [62], but it cannot be derived by means of the standard approach discussed in [62] because the channel model is not a finite-state Markov chain [8,9]. According to the turbo principle [62], the code constraints induced by the outer code are neglected in the derivation of the algorithm, and the bits $\mathbf{x}_1^T$ are considered to be statistically independent, namely the *a priori* probability $P(\mathbf{x}_1^T)$ is factorized as $\prod_{k=1}^{T} P(x_k)$, where $P(x_k)$ is 1/2 if x_k is a code bit, while it is 0 (or 1) if x_k is a pilot bit.

3.2.1 Bit Level MAP Detection

Let us define the binary event $D_{k,n}$, with two indices $k \in \{1, 2, \ldots, T\}$, and $n \in \{0, 1, \ldots, R\}$, which denotes whether, of the first k transmitted bits, exactly n bits are received, possibly after being corrupted by the channel or not. We are interested in the exact "frame synchronization" scenario, in which $D_{0,0}$ and $D_{T,R}$ are true with probability one, the values of T and R being known to the receiver. This assumption is not critical since frame synchronization can be obtained with great accuracy in practice [9]. For a better illustration of the resynchronization process, a two-dimensional grid is created to represent the synchronization errors. As shown in Fig. 3.3, the rows and columns on the grid correspond to the transmitted and received bits $x_k,\ k \in \{1, \ldots, T\}$ and $y_n,\ n \in \{1, \ldots, R\}$, respectively. The solid line refers to one particular channel realization and the dotted lines indicate the channel without any insertion or deletion errors. There are only three possible moves to reach a certain state. A

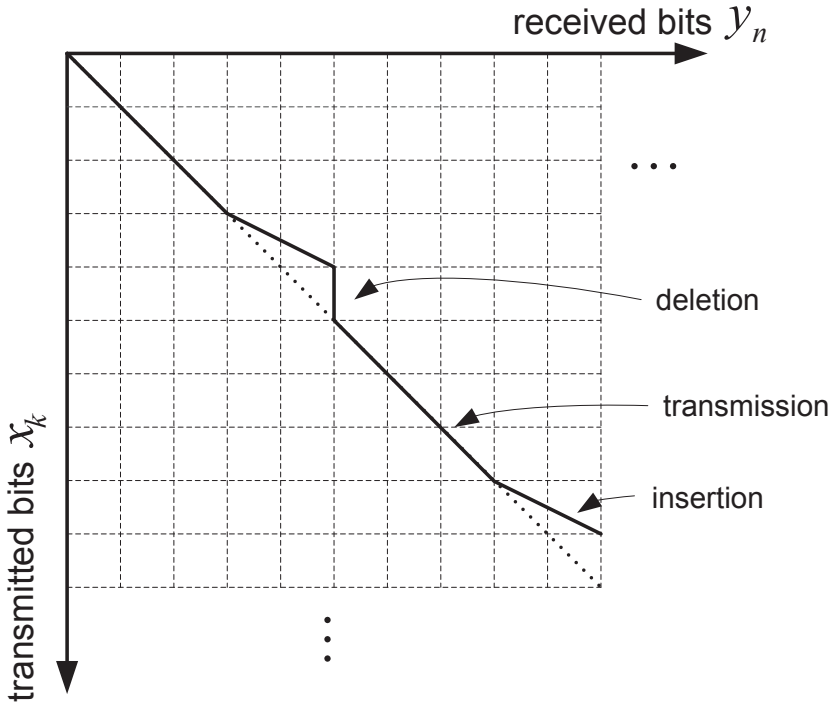

Figure 3.3: Synchronization represented by a path on a two dimensional grid.

diagonal move from the top left corner to the bottom right corner on the grid indicates a successful transmission, i.e., no insertion or deletion, but the bit may not be correctly received. An insertion event is represented by a diagonal move in two adjacent blocks and a vertical move denotes a deletion event. Let us also define the function

$$F(x_k, y_n) = \begin{cases} 1 - P_s & \text{if } y_n = x_k \\ P_s & \text{if } y_n \neq x_k \end{cases} , \tag{3.1}$$

and the coefficients

$$\alpha_k(n) = P(\mathbf{y}_1^n, D_{k,n}) , \tag{3.2}$$

$$\beta_k(n) = P(\mathbf{y}_{n+1}^R | D_{k,n}) . \tag{3.3}$$

These coefficients can be computed by means of the following forward recursion (where the differences with respect to [8, 9] are due to the adopted channel model being different):

$$\begin{aligned}
\alpha_k(n) &= P(\mathbf{y}_1^n, D_{k,n}, D_{k-1,n-2}) + P(\mathbf{y}_1^n, D_{k,n}, D_{k-1,n}) + P(\mathbf{y}_1^n, D_{k,n}, D_{k-1,n-1}) \\
&= P(\mathbf{y}_1^{n-2}, D_{k-1,n-2}) P(\mathbf{y}_{n-1}^n, D_{k,n} | D_{k-1,n-2}) \\
&\quad + P(\mathbf{y}_1^n, D_{k-1,n}) P(D_{k,n} | D_{k-1,n}) \\
&\quad + P(y_n | D_{k,n}, D_{k-1,n-1}) P(D_{k,n} | D_{k-1,n-1}) P(\mathbf{y}_1^{n-1}, D_{k-1,n-1}) \\
&= \frac{P_i}{4}\, \alpha_{k-1}(n-2) + P_d\, \alpha_{k-1}(n) + P_t\, \alpha_{k-1}(n-1) \sum_{x_k} P(x_k) F(x_k, y_n) \,,
\end{aligned} \tag{3.4}$$

and the following backward recursion [8, 9]:

$$\begin{aligned}
\beta_k(n) &= P(\mathbf{y}_{n+1}^R, D_{k+1,n+2} | D_{k,n}) + P(\mathbf{y}_{n+1}^R, D_{k+1,n} | D_{k,n}) \\
&\quad + P(\mathbf{y}_{n+1}^R, D_{k+1,n+1} | D_{k,n}) \\
&= P(\mathbf{y}_{n+1}^{n+2}, D_{k+1,n+2} | D_{k,n}) P(\mathbf{y}_{n+3}^R | D_{k+1,n+2}) \\
&\quad + P(D_{k+1,n} | D_{k,n}) P(\mathbf{y}_{n+1}^R | D_{k+1,n}) \\
&\quad + P(D_{k+1,n+1} | D_{k,n}) P(\mathbf{y}_{n+2}^R | D_{k+1,n+1}) P(y_{n+1} | D_{k+1,n+1}, D_{k,n}) \\
&= \frac{P_i}{4}\, \beta_{k+1}(n+2) + P_d\, \beta_{k+1}(n) \\
&\quad + P_t\, \beta_{k+1}(n+1) \sum_{x_{k+1}} P(x_{k+1}) F(x_{k+1}, y_{n+1}) \,,
\end{aligned} \tag{3.5}$$

which are both initialized by exploiting the "frame synchronization" assumption. Finally, the target conditional probability can be computed as

$$
\begin{aligned}
p(\mathbf{y}_1^R|x_k) &= \sum_{n=0}^{\min(2k,R)} P(\mathbf{y}_1^R, D_{k,n}|x_k) \\
&= \sum_{n=0}^{\min(2k,R)} P(\mathbf{y}_1^R, D_{k-1,n-2}, D_{k,n}|x_k) + P(\mathbf{y}_1^R, D_{k-1,n}, D_{k,n}|x_k) \\
&\qquad + P(\mathbf{y}_1^R, D_{k-1,n-1}, D_{k,n}|x_n) \\
&= \frac{P_i}{4} \sum_{n=0}^{\min(2k,R)} \alpha_{k-1}(n-2)\beta_k(n) + P_d \sum_{n=0}^{\min(2k,R)} \alpha_{k-1}(n)\beta_k(n) \\
&\quad + P_t \sum_{n=0}^{\min(2k,R)} \alpha_{k-1}(n-1)\beta_k(n)F(x_k, y_n) \; . \qquad (3.6)
\end{aligned}
$$

3.2.2 Achievable Rates by a Specific Marker Code

An interesting information-theoretic problem that arises is the following: what is the ultimate rate at which we can reliably (in the Shannon sense [7]) transmit information through the considered concatenated coding scheme? An approximate solution to this problem can be found in [6], where the authors investigate the BCJR-once bound [63] and characterize the capacity of a BSC with a time-varying substitution probability and conjecture that it provides an accurate characterization of the information rate. Here, we pursue a more precise solution to the problem. First, we notice that the ultimate rate r_C for the outer code that can be achieved through the considered concatenated coding system is given by the mutual information between the independent and uniformly distributed bits at the input of the interleaver at the transmitter side and the soft information at the output of the deinterleaver at the receiver side (see Fig. 3.1). Because of the complicated MAP detector, this mutual information cannot be computed in closed form, but it can be easily evaluated through Monte Carlo simulations with a large number of channel realizations by obtaining the histogram of the distribution of the extrinsic information

(L-values). The reason we choose histograms instead of the Arnold-Loeliger algorithm [64] is that the latter only gives the no-interleaving mutual information while our focus is mainly on interleaved systems using a soft damapper, as discussed in Section 3.3. Interestingly, this numerical method is equivalent to the evaluation of the EXIT chart for the MAP detector, particularly of its left-most point [10]. In fact, the left-most point of a detection EXIT chart gives the ultimate rate achievable by the outer code when it is concatenated with the inner detector through an interleaver and iterative detection/decoding is not allowed [10]. Hence, for a given marker code with rate r_M, we can evaluate the ultimate value of r_C by means of this numerical method, and then compute the ultimate overall rate as $r = r_C r_M$. We will exploit this result in the next subsection to find optimal marker codes for channels with insertions and deletions.

3.2.3 Marker Code Optimization

In this section, we study the problem of selecting a good marker code. We first notice that a lower marker code rate or smaller N_C leads to better synchronization capabilities since the positions of the insertions and deletions can be located more precisely; however, this is obtained with an increased overhead. This argument suggests that an optimal marker code rate r_M exists for different marker codes used over an insertion/deletion channel.

Some results obtained by means of the proposed information rate evaluation method of the previous subsection are shown in Figs. 3.4-3.6. Particularly, in Fig. 3.4, it is shown how the overall rate varies, for different deletion channels ($P_i = P_s = 0$), as a function of N_C, when the two-bit marker "01" is inserted every N_C information-carrying bits. For each value of the deletion

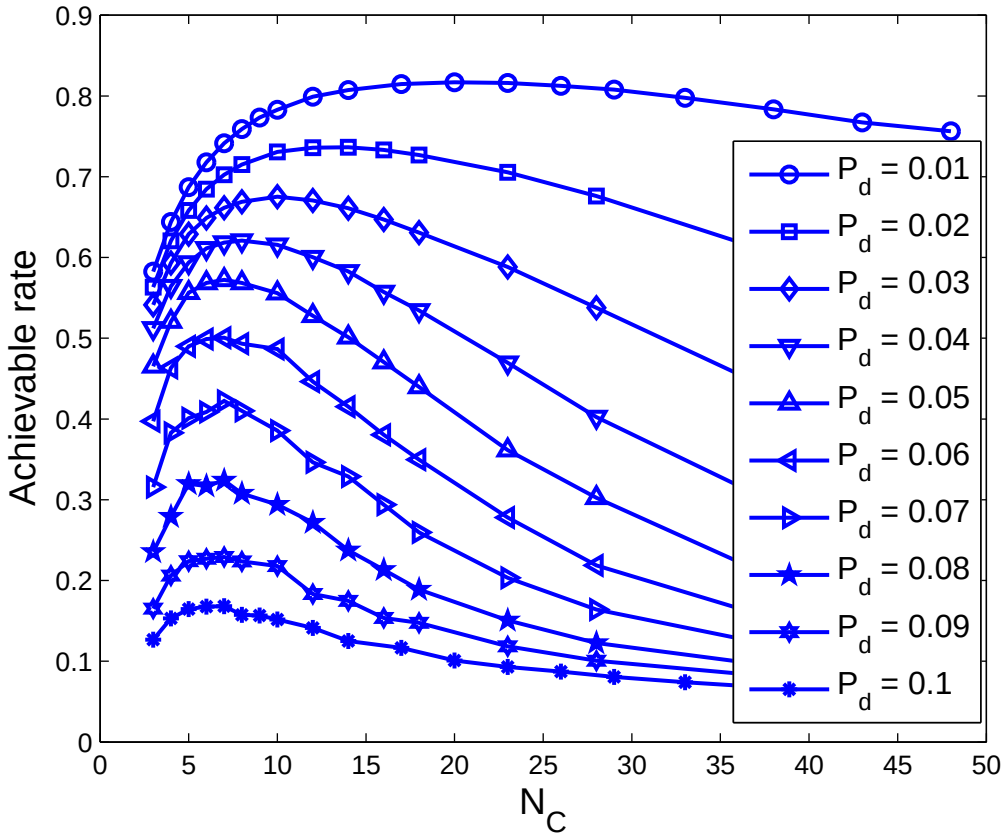

Figure 3.4: Achievable rates for different deletion channels for the marker "01" inserted every N_c bits.

probability P_d, a clear maximum is obtained, which determines the marker code rate that is information-theoretically optimal. Not surprisingly, as deletions become more frequent, the achievable rate decreases, so does the rate of the optimal marker code, since an effective synchronization process requires more pilot bits. As another example, Fig. 3.5 compares the impacts of insertion, deletion and substitution errors on the achievable rates with the constraint that $P_i + P_d + P_s = 0.03$. It is clear that for this particular example, the deletion errors cause more severe damage than the insertion errors to the performance while the substitution errors degrade the capacity much less than the synchronization errors. Note that these achievable rates are only valid if a

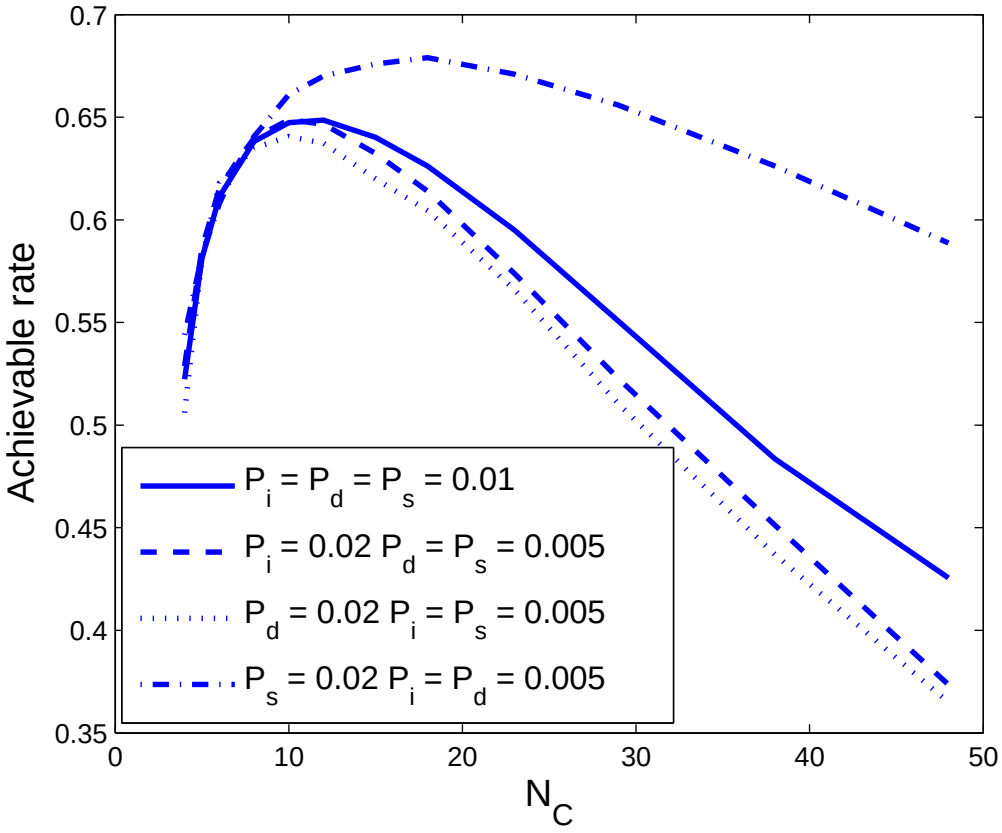

Figure 3.5: Achievable rates for different insertion and deletion channels for the marker "01" inserted every N_c bits.

single synchronization stage is employed (single-pass decoding) and they are violated when an iterative decoding/synchronization scheme is adopted. We also note that the gap to the existing Shannon capacity lower bounds is also large. For instance, a lower bound for the capacity of an i.i.d. deletion channel with $P_d = 0.05$ is 0.728 [65] while the maximum rate found in Fig. 3.4 is less than 0.6.

The proposed approach can be used not only to find the optimal rate for a given marker code, but also to compare different marker codes. As an example, the marker code "00" is clearly not a good choice compared to "01" since there is no transition between the two bits and the receiver cannot

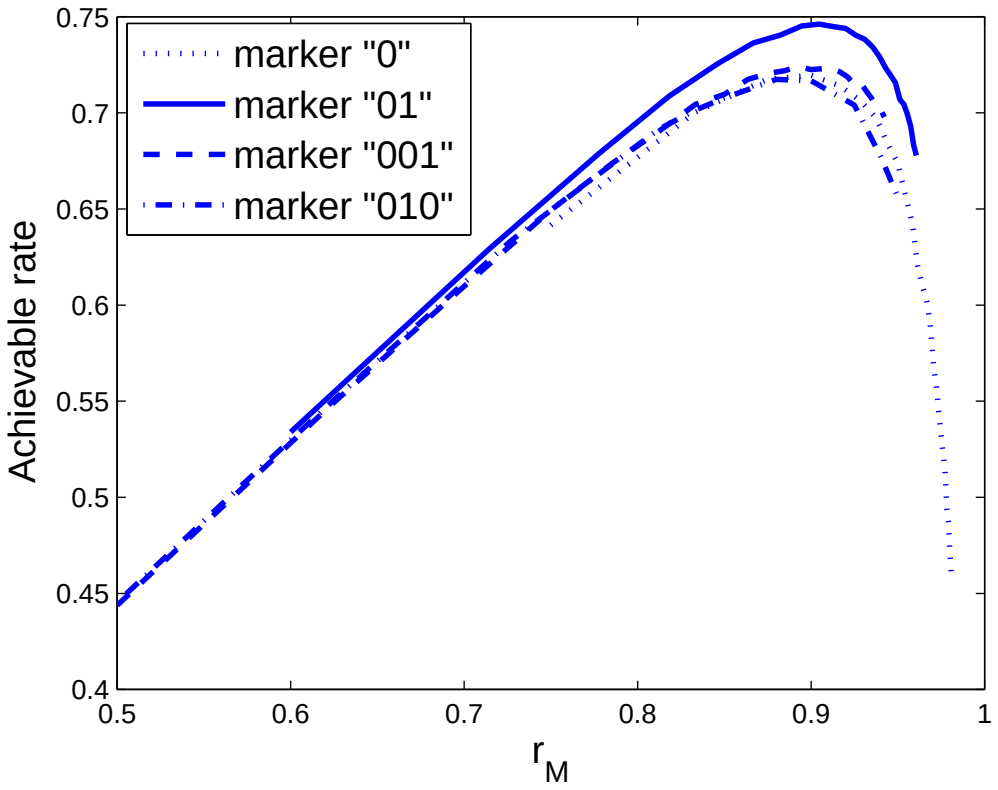

Figure 3.6: Achievable rates for different markers as a function of the marker code rate when $P_d = 0.01$, $P_s = 0.01$.

determine as precisely whether an insertion or deletion error happens prior to the specific marker. On the other hand, for "01", there is a transition in the marker sequence and a single deletion or insertion can be easily identified. In Fig. 3.6, we compare four regular marker codes obtained by inserting the markers "0", "01", "001", and "010" every N_C information bits, for the case of a deletion/substitution channel with $P_d = P_s = 10^{-2}$. The results, which are given in terms of the overall rate r as a function of the marker code rate r_M, show that for this particular channel the best choice of marker code among the three candidates is to insert the pilot bits "01" every 18 information bits, which provides an overall rate of about 0.75. It is also not surprising to see that the marker "001" outperforms "010" for higher marker code rates. This

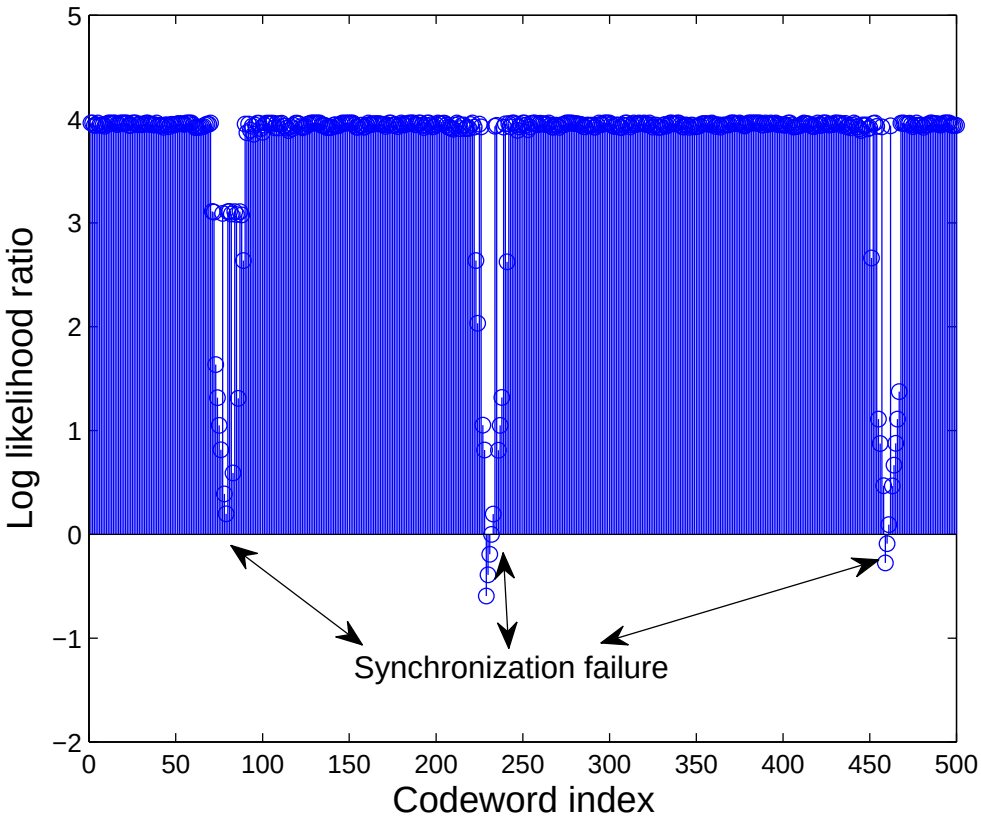

Figure 3.7: Example of the bit level MAP detector output.

is attributed to the following: it is more likely that more than one bit get deleted between two adjacent markers and hence the marker "010" may not be able to detect these synchronization errors while the marker "001" still can.

We conclude this section by noting that for all the studied scenarios, the guidelines for marker code design that we obtain through our analysis are in good agreement with the approximate analysis proposed in [6].

3.3 Symbol-Level Synchronization

One key observation is that, since insertion/deletion channels have memory, the soft information at the output of the MAP detector corresponding to two

bits with different time indices is correlated. An example is given in Fig. 3.7 which shows the LLRs of the decoding output from the bit level MAP detector. The detector identifies which blocks of bits experience synchronization errors by identifying blocks of bits with log-likelihood ratios close to zero, since synchronization cannot be re-achieved until the next marker is received. Consequently, if a bit is not correctly detected, the detection of the following bit fails with a high probability. If the correlations between consecutive bits are considered and utilized, better decoding performance may be achieved, and information is lost when such correlations are neglected, which is exactly what is done in our concatenated system due to the presence of the interleaver/deinterleaver[1] [10]. On the other hand, interleaving is fundamental because it allows us to split the decoding process into two serial steps, namely the inner detection and the outer decoding; the other option being joint detection/decoding, which would be computationally infeasible [5, 6, 9]. In the following, we propose a solution that allows us to recover part of the information loss while preserving the interleaving process, hence also its advantage of splitting the decoding process into inner detection and outer decoding.

3.3.1 Symbol Level MAP Detection

We introduce MAP detection at the symbol level, defining a symbol as a group of m consecutive bits. Consequently, the T transmitted bits are partitioned into T_S symbols, $S_k = \mathbf{x}_{m(k-1)+1}^{mk}$, $k \in \{1, 2, \ldots, T_S\}$, taking values on $\{0, 1\}^m$. The last symbol, however, may consists of less than m bits, but we assume that $T/m = T_S$ is an integer for simplicity. Such computations can be carried out by means of a symbol-level FBA which is obtained by extending the bit-

[1] In the context of outer LDPC codes, the interleaving is implicit.

level derivation given in [8, 9] to the symbol-level case. In the following, we provide the details of the algorithm.

Let us re-define the binary event $D_{k,n}$, with $k \in \{1, 2, \ldots, T_S\}$ and $n \in \{1, 2, \ldots, R\}$, which denotes whether, of the first k transmitted symbols (i.e., km bits), exactly n bits are received or not, possibly after being corrupted by the channel. With this redefinition of the event $D_{k,n}$, the definitions in (3.2) and (3.3) still hold. As in the bit-level case, the coefficients can be computed by means of the forward/backward recursions. For simplicity, we give here the formulations for the case $m = 2$, i.e., bits $\{x_{2k-1}, x_{2k}\}$ are grouped as one symbol, noting that the extension to the case of $m > 2$ is straightforward. In this case, there are 9 possible ways to reach a certain state on the trellis, and the resulting recursions are given as follows:

$$\begin{aligned}
\alpha_k(n) = {} & P_d^2\, \alpha_{k-1}(n) + P_d P_t\, \alpha_{k-1}(n-1) \sum_{i=0}^{1} \sum_{x_{2k-i}} P(x_{2k-i}) F(x_{2k-i}, y_n) \\
& + P_t^2\, \alpha_{k-1}(n-2) \sum_{x_{2k-1}} P(x_{2k-1}) F(x_{2k-1}, y_{n-1}) \cdot \sum_{x_{2k}} P(x_{2k}) F(x_{2k}, y_n) \\
& + \frac{P_i}{4} P_d\, \alpha_{k-1}(n-2) \cdot 2 + \frac{P_i^2}{16}\, \alpha_{k-1}(n-4) \\
& + \frac{P_i}{4} P_t\, \alpha_{k-1}(n-3) \sum_{i=0}^{1} \sum_{x_{2k-i}} P(x_{2k-i}) F(x_{2k-i}, y_{n-2i}) \qquad (3.7)
\end{aligned}$$

and

$$\begin{aligned}
\beta_k(n) = {} & P_d^2\, \beta_{k+1}(n) + P_d P_t\, \beta_{k+1}(n+1) \sum_{i=1}^{2} \sum_{x_{2k+i}} P(x_{2k+i}) F(x_{2k+i}, y_{n+1}) \\
& + P_t^2\, \beta_{k+1}(n+2) \sum_{x_{2k+1}} P(x_{2k+1}) F(x_{2k+1}, y_{n+1}) \\
& \qquad \cdot \sum_{x_{2k+2}} P(x_{2k+2}) F(x_{2k+2}, y_{n+2}) \\
& + \frac{P_i}{4} P_d\, \beta_{k+1}(n+2) \cdot 2 + \frac{P_i^2}{16}\, \beta_{k+1}(n+4) \\
& + \frac{P_i}{4} P_t\, \beta_{k+1}(n+3) \sum_{i=1}^{2} \sum_{x_{2k+i}} P(x_{2k+i}) F(x_{2k+i}, y_{n+2i-1}) \,, \qquad (3.8)
\end{aligned}$$

respectively, and are both initialized by exploiting again the exact "frame synchronization" assumption. Finally, the target extrinsic information can be computed to be

$$\begin{aligned} p(\mathbf{y}_1^R|x_{2k-1},x_{2k}) &= P_d^2 \sum_{n=0}^{\min(4k,R)} \alpha_{k-1}(n)\beta_k(n) \\ &\quad + P_d P_t \sum_{n=0}^{\min(4k,R)} \sum_{i=0}^{1} \alpha_{k-1}(n-1)\beta_k(n)F(x_{2k-i},y_n) \\ &\quad + P_t^2 \sum_{n=0}^{\min(4k,R)} \alpha_{k-1}(n-2)\beta_k(n)F(x_{2k-1},y_{n-1})F(x_{2k},y_n) \\ &\quad + \frac{P_i}{4} P_d \sum_{n=0}^{\min(4k,R)} \alpha_{k-1}(n-2)\beta_k(n)\cdot 2 \\ &\quad + \frac{P_i}{4} P_t \sum_{n=0}^{\min(4k,R)} \sum_{i=0}^{1} \alpha_{k-1}(n-3)\beta_k(n)F(x_{2k-i},y_{n-2i}) \\ &\quad + \frac{P_i^2}{16} \sum_{n=0}^{\min(4k,R)} \alpha_{k-1}(n-4)\beta_k(n)\ . \end{aligned} \tag{3.9}$$

3.3.2 Achievable Rate Improvement with Symbol Level Synchronization

As an example use of the proposed algorithm, Fig. 3.8 compares the mutual information between the symbols at the input of the interleaver at the transmitter side and the soft information at the output of the deinterleaver at the receiver side, for the case of one-bit symbols and two-bit symbols. Specifically, it is shown how the overall achievable rate varies, for an i.i.d. deletion channel ($P_i = P_s = 0,\ P_d = 0.01$), as a function of N_C, when the two-bit marker "01" is inserted every N_C information-carrying bits. For comparison, the mutual information computed in the absence of interleaving, i.e., by evaluating the expectations $E[\log P(\mathbf{y}_1^R)]$ and $E[\log P(\mathbf{x}_1^T,\mathbf{y}_1^R)]$ using Monte Carlo techniques with a large number of channel simulations, and obtaining $I(\mathbf{x}_1^T;\mathbf{y}_1^R)$ as $T - E[\log P(\mathbf{y}_1^R)] + E[\log P(\mathbf{x}_1^T,\mathbf{y}_1^R)]$ [18], is also shown. This

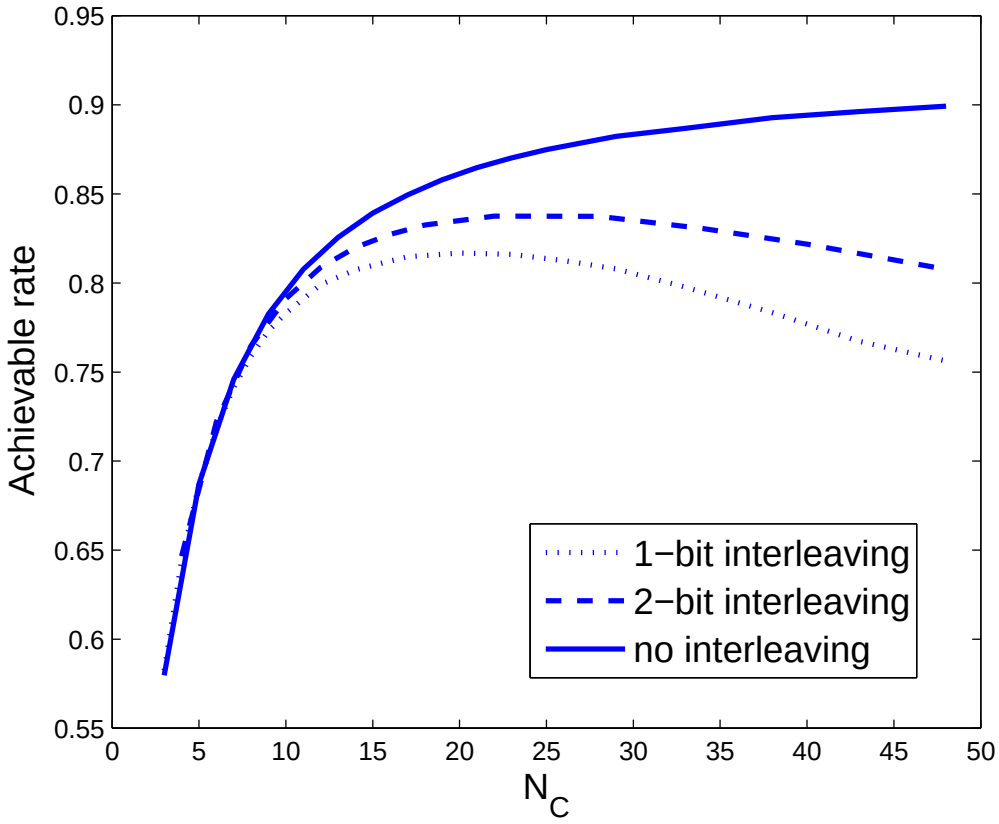

Figure 3.8: Achievable rate improvement through symbol-level decoding for the marker "01" inserted every N_c bits.

curve quantifies the transmission rate loss due to interleaving. Since the complexity of the algorithm grows exponentially in the group size m which makes it infeasible for large values of m, only the achievable rate for the case of 2-bit interleaving is shown. It is clear that adopting symbol-level detection recovers a significant part of the interleaving loss, particularly as the marker code rate increases. For instance, by comparing the two relevant maximum achievable rates, we can conclude that symbol-level detection is about 5% better in capacity for the given example. Although omitted from this chapter, other simulation results also show similar gains for different channels, e.g., the insertion only channel or insertion/deletion channels.

3.3.3 Exploiting Correlation via Demapper/Detector

In this section, we consider a practical coding scheme with the aim of confirming the performance gain predicted by our information-theoretic analysis for the symbol-level detection over the bit-level detection. Specifically, we adopt a binary LDPC code of length 16383 and rate $r_C = 0.87$ concatenated with a marker code with rate $r_M = 30/32$, obtained by inserting the marker "01" every 30 LDPC-coded bits. Hence, the value $r = 0.8156$ is obtained for the overall code rate. We compare the performance obtained by feeding the LDPC decoder with the soft information produced by the bit-level detector and the symbol-level detector (with $m = 2$ and $m = 3$). In the bit-level detection case, the output of the detector directly feeds the LDPC decoder, which performs 100 self iterations and then produces the estimate of the information bits. In the symbol-level detection case, the output of the detector cannot directly feed the LDPC decoder, which is binary and cannot manage symbol-level soft information. Hence, to convert the symbol-level information to bit-level information, we adopt the soft demapper module proposed in [66].

Fig. 3.9 illustrates the process of iterative information exchange between the demappers and the variable nodes from the outer LDPC code. Suppose that the 2-bit level MAP decoder generates the output information $\xi(S_k) = P(\mathbf{y}_1^R|S_k)$ where S_k is the symbol representation of two consecutive bits x_{2k-1} and x_{2k}. We first start the LDPC decoder with the *a priori* information only from $\xi(S_k)$. When the decoding output of bit x_{2k-1}, $\xi(x_{2k-1})$, is available after a few iterations, we treat it as the approximation of $P(x_{2k-1})$ and use it to generate new *a priori* information for bit x_{2k}, $\varphi(x_{2k}) = P(\mathbf{y}_1^R|x_{2k})$,

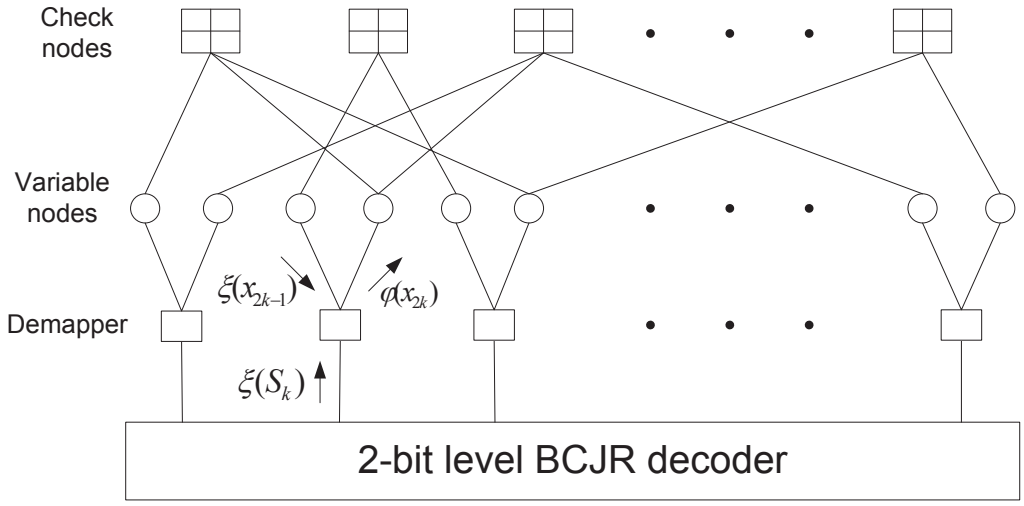

Figure 3.9: Achievable rate improvement through symbol-level decoding for the marker "01" inserted every N_c bits.

using

$$\begin{aligned}
\varphi(x_{2k}) &= \sum_{S_k} P(\mathbf{y}_1^R, S_k|x_{2k}) \\
&= \sum_{S_k} P(S_k|x_{2k}) P(\mathbf{y}_1^R|S_k, x_{2k}) \\
&= \sum_{S_k} P(S_k|x_{2k}) P(\mathbf{y}_1^R|S_k) \\
&= \sum_{S_k} \sum_{x_{2k-1}} P(S_k, x_{2k-1}|x_{2k}) P(\mathbf{y}_1^R|S_k) \\
&= \sum_{S_k} \sum_{x_{2k-1}} \xi(x_{2k-1}) \xi(S_k) P(S_k|x_{2k-1}, x_{2k}). \qquad (3.10)
\end{aligned}$$

According to (3.10), we can re-initialize the bit level LDPC decode with the updated *a priori* information for the variable nodes and obtain better estimates of the transmitted bits.

In the simulation, for every 10 self iterations of the LDPC decoder, we perform one iteration of the soft demapper, so that the total number of 100 self iterations of the LDPC decoder is preserved for a fair comparison with the

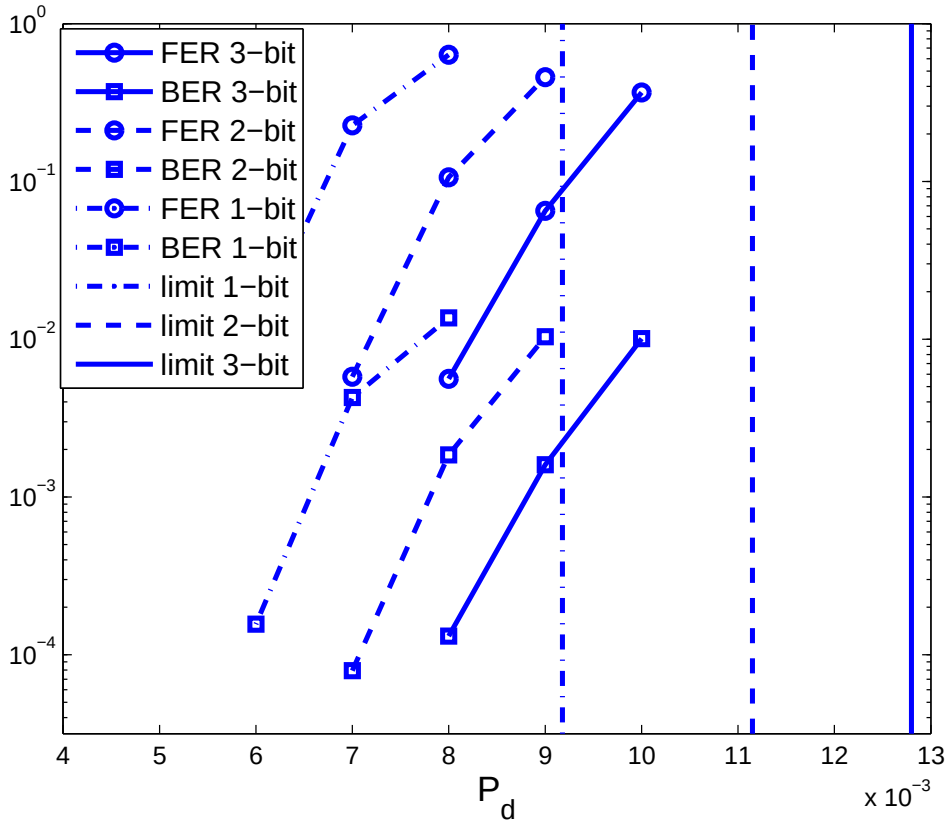

Figure 3.10: Decoding improvement through symbol-level decoding for 1-bit interleaving, 2-bit interleaving and 3-bit interleaving.

bit-level case. The resulting frame-error rate (FER) and bit-error rate (BER) curves are compared in Fig. 3.10, for the case of a deletion only channel. For comparison, the ultimate deletion probability P_d at which a scheme with the considered marker code and an outer code with rate $r_C = 0.87$ can provide reliable communications is also shown — these values are obtained by means of the information-theoretic analysis described in the previous sections. An interesting fact is that a BER lower than 10^{-2} is obtained by means of MAP detection with $m = 3$ at values of the deletion probability at which bit-level detection cannot converge even in the presence of an information-theoretically optimal code as shown in Fig. 3.10. The improvement provided by the symbol-

level detection is evident: for a given BER, using a MAP detector with $m = 2$ allows the receiver to work with a deletion probability increased by about 10^{-3} with respect to the bit-level one, and the MAP detector with $m = 3$ provides an even greater robustness to deletion errors.

3.4 EXIT Chart-Based Outer LDPC Code Design for Insertion/Deletion Channels

In the previous sections, with the interest of reducing decoding latency, we focused on the case of single-pass decoding for the outer code concatenated with the inner marker code over insertion/deletion channels. We now consider an iterative scheme where extrinsic information is exchanged between the MAP detector (synchronization) block and the outer decoder. This is motivated by the observation that when iterative decoding is allowed, specifically designed LDPC codes for insertion and deletion channels may provide performance gains over the ones optimized for AWGN-only channels. Detailed EXIT chart based analysis offers an insight into this problem.

In this section, we consider an LDPC code consisting of N variable nodes and $N - K$ check nodes connected by an edge interleaver [57] with code rate $r_C = K/N$. For simplicity, as in [10], only check-regular LDPC codes are considered, i.e., every parity-check equation involves a constant number of variable nodes, denoted by d_c. We emphasize that joint design of variable and check nodes may offer a better performance but the check-regular LDPCs already give good results as reported in the previous literature. Suppose I is the total number of different variable node degrees of the LDPC code denoted by $d_{v,i},\ i = 1, \ldots, I$. Let a_i to be the fraction of variable nodes with degree $d_{v,i}$. The goal of code design is to find the set of parameters $\{\lambda_i\}$ that provides

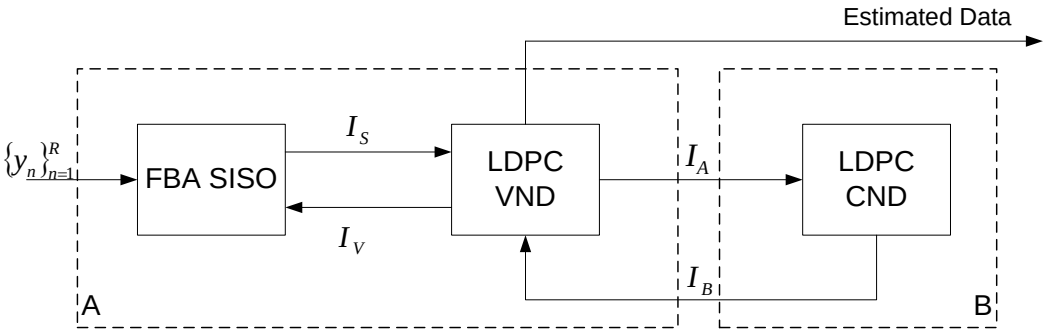

Figure 3.11: Detailed decoder/detector block diagram at the receiver side.

the best decoding performance where [10]

$$\sum_{i=1}^{I} \lambda_i = 1, \quad 0 \leq \lambda_i \leq 1,$$
$$\lambda_i = \frac{d_{v,i}}{(1 - r_C)d_c} \cdot a_i. \tag{3.11}$$

Because of the first constraint, we need $I \geq 3$ to have any flexibility in our code design.

3.4.1 EXIT Chart Based Analysis of the Decoding Performance

Since the outer LDPC decoder can be partitioned into LDPC variable node detector (VND) and LDPC check node detector (CND) [10], for multiple-pass decoding, the information exchanged between the inner MAP detector and outer LDPC decoder is further illustrated in Fig. 3.11, where Block A consists of two sub-blocks which are referred to as FBA SISO and LDPC VND. Mutual information between the LDPC-coded bits and the corresponding L-values, $\{I_A, I_B, I_S, I_V\} \in [0, 1]$, are exchanged between these blocks during the iterative decoding process. It is worth mentioning that only the extrinsic information, i.e., the difference between the *a priori* and the *a posteriori* L-values, is exchanged [10].

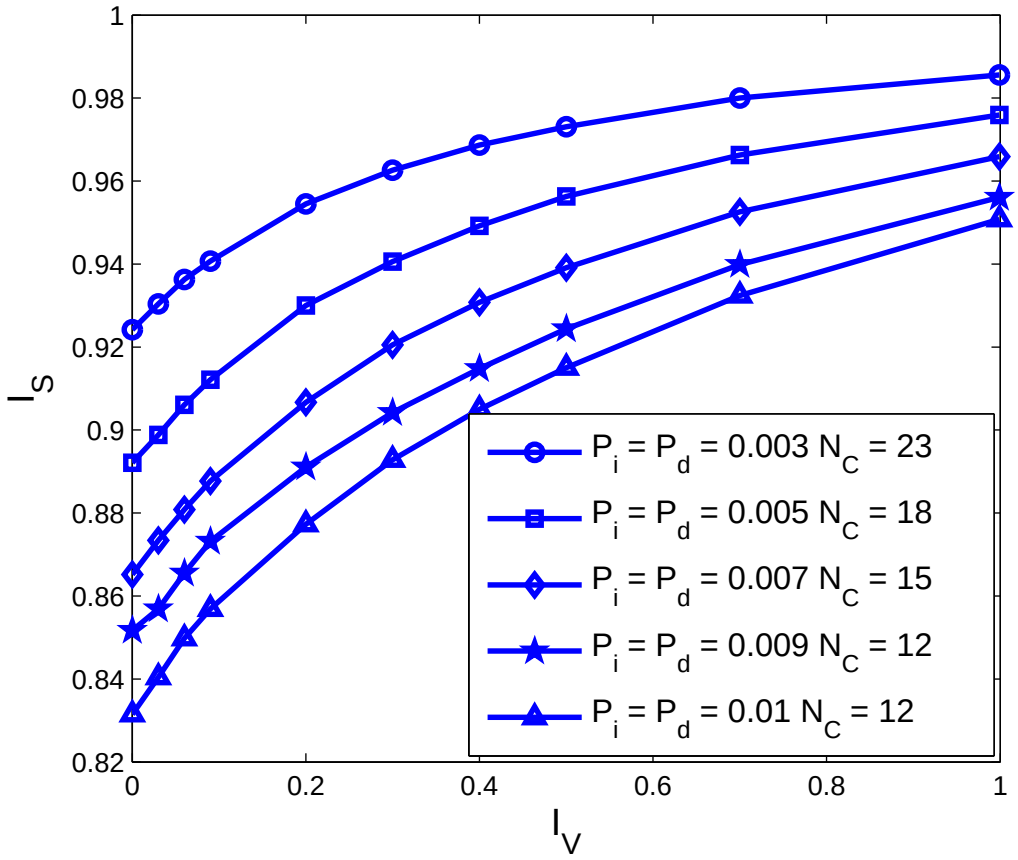

Figure 3.12: Detection EXIT chart for several insertion and deletion channels for the marker "01" inserted every N_c bits.

As stated in Section 3.2, in the sub-block FBA SISO, MAP detection is applied on the received sequence $\{y_k\}$ with soft input *a priori* information given by I_V and extrinsic L-values of the transmitted bits are generated. I_S measures the reliability of these L-values. It is difficult to describe the relationship between I_V and I_S in closed form, instead, Monte Carlo simulations are performed to generate the so-called detection EXIT chart. A detection EXIT chart example for insertion and deletion channels is shown in Fig. 3.12 using bit-level synchronization and the marker code "01". Marker code rates are chosen based on the scheme proposed in Section 3.2.3.

The variable nodes take I_S as the *a priori* information and perform the standard sum-product algorithm (SPA) with information received from the LDPC CND. The EXIT curve of the combined FBA SISO and LDPC VND is described by the relationship between I_A and I_B, given by [10]

$$I_A(I_B, d_v) = J\left(\sqrt{(d_v - 1)[J^{-1}(I_B)]^2 + [J^{-1}(I_S)]^2}\right), \tag{3.12}$$

where the function $J(\sigma)$ is defined as

$$J(\sigma) = 1 - \int_{-\infty}^{\infty} \frac{e^{-(\xi - \sigma^2/2)^2/2\sigma^2}}{\sqrt{2\pi\sigma^2}} \cdot \log_2[1 + e^{-\xi}]d\xi. \tag{3.13}$$

In this case, I_S can be numerically evaluated from the detection EXIT chart using a polynomial approximation with input $I_V(I_B, d_v) = J\left(\sqrt{d_v} \cdot J^{-1}(I_B)\right)$. For instance, when $P_i = P_d = 0.01$, we can write,

$$I_S = 0.41491 \cdot I_V^5 - 1.1518 \cdot I_V^4 + 1.2405 \cdot I_V^3 - 0.71968 \cdot I_V^2 + 0.33549 \cdot I_V + 0.83146.$$

For a certain variable node degree distribution, the effective VND transfer curve is thus

$$I_A(I_B) = \sum_{i=1}^{I} \lambda_i \cdot I_A(I_B, d_{v,i}) = \sum_{i=1}^{I} \lambda_i \cdot J\left(\sqrt{(d_{v,i} - 1)[J^{-1}(I_B)]^2 + [J^{-1}(I_S)]^2}\right). \tag{3.14}$$

At the CND, "box plus" operation [61] is done to generate I_B from I_A which can be approximately written as (it is useful to express it as the inverse function) [10]

$$I_B^{-1}(I_A, d_c) = I_A(I_B, d_c) \approx 1 - J\left(\frac{J^{-1}(1 - I_B)}{\sqrt{d_c - 1}}\right). \tag{3.15}$$

The EXIT curves $I_A(I)$ and $I_B^{-1}(I)$ form the EXIT chart for the entire receiver which predicts the decoding performance. At the initialization step of the decoding process, FBA SISO computes the output extrinsic L-values with no *a priori* information i.e., $I_V = 0$. VND utilizes the mutual information I_S

to start the SPA during which the mutual information $I_A(0)$ and $I_B\big(I_A(0)\big)$ are exchanged between VND and CND. After one iteration of SPA, VND generates the output extrinsic information which serves as the *a priori* information for FBA SISO and starts to iterate from the first step. The difference between the current and previous iteration is that VND produces more reliable information $I_A\Big(I_B\big(I_A(0)\big)\Big)$ if $I_A(I) > I_B^{-1}(I)\ \forall I \in [0,1)$. Iterative decoding stops when a valid LDPC codeword is obtained or the maximum number of iterations is reached. At the end of the process, the overall L-values are produced for the estimation of the transmitted bit sequence. Note that when the condition $I_A(I) > I_B^{-1}(I)\ \forall I \in [0,1)$ is satisfied, i.e., the "tunnel" created by the two curves $I_A(I)$ and $I_B^{-1}(I)$ is open, the mutual information of VND output will converge to 1 after several iterations which leads to a decoding performance with an error rate approaching zero. Thus, the goal of LDPC code design is to find a set of $\{\lambda_i\}$ that keeps the tunnel open for the highest deletion/insertion rate.

3.4.2 LDPC Code Design Example for Insertion/Deletion Channels

Design examples are given in Table 3.1 using the bit-level synchronization algorithm for several insertion/deletion channels, deletion only channels and insertion only channels, respectively. We choose $I = 3$ and fix the average variable node degree $\bar{d}_v$ to be 3. Listed LDPC code degree distributions guarantee convergence with the highest code rate r_C for different deletion/insertion rates. Therefore, the overall code rate r, product of r_M and r_C, denotes the highest achievable rate when iterative decoding is performed. For deletion probabilities of 0.01 and 0.1, the overall rates are obtained as 0.860 and 0.486, respectively, where the capacity lower bound is 0.919 for $P_d = 0.01$ and 0.531 for $P_d = 0.1$. The corresponding gaps are 0.059 and 0.045 for the two cases,

Table 3.1: LDPC Code Parameters for Insertion and Deletion Channels

	r_M	r_C	d_c	d_v	a
$P_i = P_d = 0.001$	0.96	0.9841	189	{2 3 225}	{0.8183 0.178 0.0037}
$P_i = P_d = 0.003$	0.92	0.96	75	{2 3 104}	{0.1782 0.82 0.0018}
$P_i = P_d = 0.005$	0.9	0.9412	51	{2 3 75}	{0.2762 0.72 0.0038}
$P_i = P_d = 0.007$	0.8824	0.9231	39	{2 3 57}	{0.2769 0.718 0.0051}
$P_i = P_d = 0.009$	0.8571	0.9091	33	{2 3 44}	{0.244 0.75 0.006}
$P_i = P_d = 0.01$	0.8571	0.9	30	{2 3 45}	{0.3233 0.669 0.0077}
$P_i = P_d = 0.02$	0.8333	0.8125	16	{2 3 52}	{0.4175 0.574 0.0085}
$P_i = P_d = 0.03$	0.8333	0.7273	11	{2 3 56}	{0.5751 0.414 0.0109}
$P_i = P_d = 0.04$	0.8333	0.625	8	{2 3 97}	{0.4641 0.531 0.0049}
$P_i = P_d = 0.05$	0.8333	0.5	6	{2 3 97}	{0.2286 0.769 0.0024}
$P_d = 0.002$	0.96	0.9836	183	{2 3 227}	{0.3723 0.626 0.0017}
$P_d = 0.006$	0.9355	0.9605	76	{2 3 161}	{0.1371 0.862 0.0009}
$P_d = 0.01$	0.9091	0.9464	56	{2 3 90}	{0.1928 0.805 0.0022}
$P_d = 0.02$	0.875	0.9091	33	{2 3 65}	{0.2401 0.756 0.0039}
$P_d = 0.04$	0.8	0.85	20	{2 3 33}	{0.2148 0.778 0.0072}
$P_d = 0.06$	0.7778	0.7857	14	{2 3 31}	{0.2868 0.703 0.0102}
$P_d = 0.08$	0.7778	0.7	10	{2 3 50}	{0.1978 0.798 0.0042}
$P_d = 0.1$	0.7778	0.625	8	{2 3 69}	{0.2679 0.728 0.0041}
$P_i = 0.002$	0.96	0.9839	186	{2 3 256}	{0.4801 0.518 0.0019}
$P_i = 0.006$	0.9355	0.9605	76	{2 3 170}	{0.1402 0.859 0.0008}
$P_i = 0.01$	0.9091	0.9464	56	{2 3 87}	{0.2293 0.768 0.0027}
$P_i = 0.02$	0.875	0.9091	33	{2 3 53}	{0.2775 0.717 0.0055}
$P_i = 0.04$	0.8	0.85	20	{2 3 25}	{0.2554 0.733 0.0116}
$P_i = 0.06$	0.7778	0.7857	14	{2 3 30}	{0.2507 0.74 0.0093}
$P_i = 0.08$	0.7778	0.7273	11	{2 3 29}	{0.4314 0.552 0.0166}
$P_i = 0.1$	0.7778	0.6667	9	{2 3 28}	{0.5125 0.467 0.0205}

which are clearly smaller than the one demonstrated in Section 3.2.3. We also expect that the gap to the capacity bound can be further narrowed by allowing $I > 3$ and not fixing $\bar{d}_v$ to be 3.

The advantages of the designed codes are also confirmed by the error rate simulation results shown in Fig. 3.13 and Table 3.2. In the figure, we pick three codes for insertion/deletion channels with rates $r_C = 0.96$, $r_C = 0.9$ and $r_C = 0.5$ from Table 3.1 and compare them with the codes optimized for AWGN channels. The length of the LDPC codeword is set to be $N = 5000$

Table 3.2: Performance Improvement at a BER level of 10^{-3} with Specific LDPC Code Design over Insertion/Deletion Channels

$P_i = P_d$	Ratio of BERs	$P_i = P_d$	Ratio of BERs
0.001	1.04	0.1	8.477
0.003	1.256	0.2	14.83
0.005	3.061	0.3	22.07
0.007	5.065	0.4	29.31
0.009	7.018	0.5	35.68

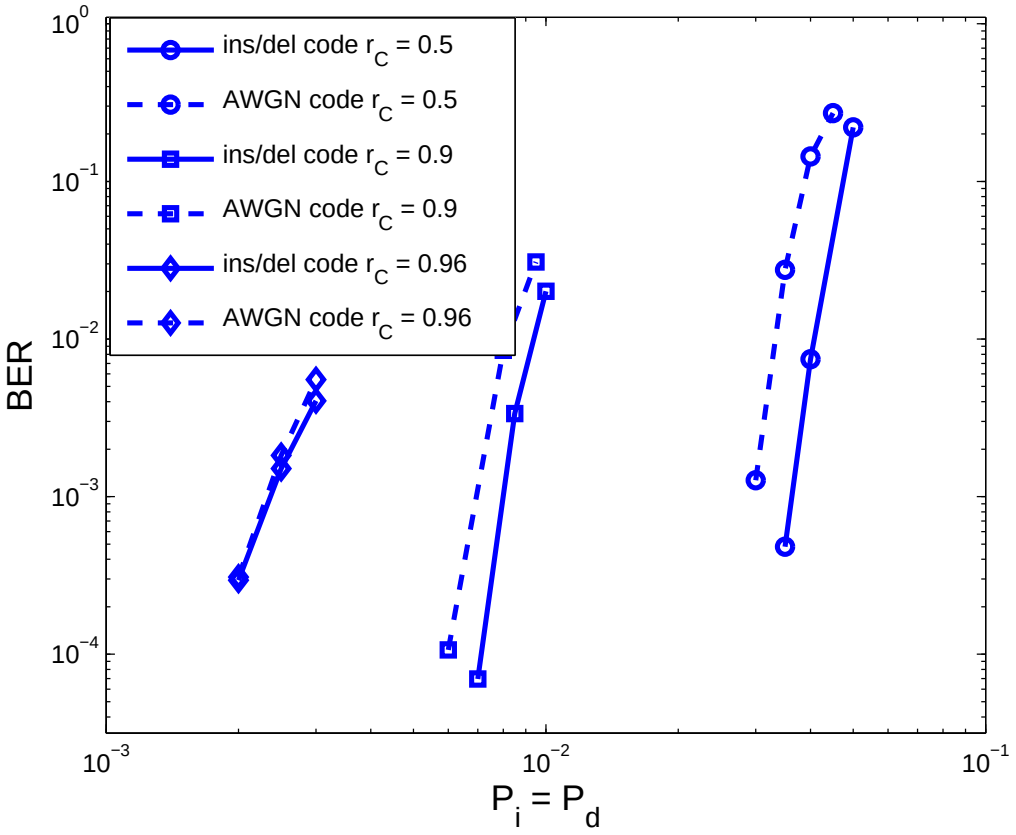

Figure 3.13: BER performance of different LDPC codes over an insertion/deletion channel with $P_i = P_d$.

and the selected marker code rate is determined to maximize the transmission rate. In Table 3.2, we calculate the ratio of the BERs of the two codes (optimized one versus the AWGN-only code) when the codes optimized for insertion and deletion channels attain a BER of 10^{-3}. The higher the ratio, the greater the improvement. Clearly, all of the codes outperform the ones designed for AWGN channels. However, the gap becomes less obvious as the insertion/deletion rate decreases. This is not a surprising result because for low insertion/deletion probabilities, the detection EXIT chart tends to be flat as illustrated in Fig. 3.12, which is similar to the one for a memoryless AWGN channel. In this case, specific design of LDPC code for insertion/deletion channel may not be required since the gain is negligible. Similar conclusions are drawn for ISI channels with short channel impulse responses in [13]. Also, when the symbol-level detection is performed, the left-most point in the detection EXIT chart is much better than the bit-level case as explained in Section 3.3.2. The right-most point in the detection EXIT chart is identical for both cases since MAP detector achieves ideal synchronization in this case. Therefore, the detection EXIT chart for symbol-level detection is flatter than the one for the bit-level case. This observation suggests that for channels with low insertion/deletion rates, it is more likely that symbol-level detection itself already yields a good performance and iterative decoding and LDPC code design may not be needed, which is also an obvious fact, since when $m = T$, optimal detection (i.e., for synchronization purposes) is achieved and there is no gain with iterative decoding/demapping. Clearly, this is not feasible in practice since the detection complexity in m is exponential and T is typically large.

3.5 Chapter Summary

In this chapter, we have studied performance of outer LDPC codes concatenated with inner marker codes for data transmission over insertion/deletion channels. Two decoding strategies are considered: single-pass decoding and multi-pass decoding with information exchange between the inner detector and the outer decoder. For the first case, through numerical mutual information analyses, we have developed a technique that allows us to optimize the marker code based on the ultimate rate achievable by the concatenated scheme. Moreover, we have presented a new symbol-level detection algorithm, which has been proved to outperform the standard bit-level one in terms of achievable rates. An iterative detector/demapper is also designed which is able to exploit the results of the symbol level synchronizer. Finally, when iterative decoding is allowed, we have shown that by choosing good variable and check node degree distributions, LDPC codes designed for insertion and deletion channels offer better error correcting capabilities than those optimal for the AWGN-only channels. Simulation results related to practical LDPC codes showing clear performance gains have been provided for both cases under consideration. Although we only focus on marker codes (as the inner synchronization code), similar analyses and design procedure can also be applied to other concatenated coding schemes, e.g., an LDPC code concatenated with an inner watermark code [9].

Chapter 4

CAPACITY BOUNDS AND CONCATENATED CODES OVER SEGMENTED DELETION CHANNELS

In this chapter, we focus on segmented deletion channels. As described in Chapter 2, the deletion errors are no longer i.i.d., since for each segment of bits, only a limited number of deletions are allowed to occur. This channel model has recently been proposed as motivated by the fact that for practical systems, when a deletion error occurs, it is more likely that the next one will not appear very soon. Our specific focus is on the elementary segmented deletion channel where at most one bit is allowed to be deleted from each segment.

We first argue that such channels are information stable, hence their channel capacity exists. Then, we introduce several upper and lower bounds with two different methods in an attempt to understand their channel capacity behavior. The first scheme utilizes certain information provided to the transmitter and/or receiver while the second one explores the asymptotic behavior of the bounds when the average bit deletion rate is small. Obtained results indicate that when the deletion probability is near zero or near unity (for each segment), the upper and lower bounds are close to each other hence a characterization of the channel capacity is obtained. Also, for a certain range of deletion probability, the capacity lower bound and estimated capacity using the second approach behave similarly, however, there is a wide-range of deletion probabilities where they are far apart. In the second part of the chapter, we utilize the same concatenated coding scheme as in Chapter 3 to correct possible deletion errors introduced by the channel. We introduce different

MAP based channel synchronization algorithms operating at the bit and symbol levels. Specific LDPC code designs for segmented deletion channels and simulation results are also given.

In Section 4.1, we prove that the Shannon capacity exists, and describe several capacity upper and lower bounds for the segmented deletion channel. In Section 4.2, we introduce the proposed concatenated coding scheme along with the suitable MAP detection algorithms to provide synchronization which incorporate the segmentation assumption. In Section 4.3, simulation results for some practical codes are reported. Finally, chapter summary is provided in Section 4.4.

4.1 Capacity Bounds for Segmented Deletion Channels

4.1.1 Existence of the Shannon Capacity

We first show that the results of Dobrushin in [15] can be applied directly to the segmented deletion channel model and as a result the Shannon capacity exists. The key observation is that Dobrushin's result is more general than the usual set-up that it is applied to, that is, information stability [67] holds for a memoryless channel with synchronization errors, indicating that the asymptotic behavior of the mutual information density between the input and output sequences over the sequence length converges to its mean. Therefore, the Shannon capacity exists, even when the channel input and output alphabets are not identical (e.g. binary) and the information and the transmission capacities are equal. The segmented deletion channel model can equivalently be described by a 2^b-ary input symbol X', and binary sequence of output bits $\mathbf{Y}'$ (of varying lengths, e.g., for the elementary segmented deletion channel, of length b or $b-1$ bits), it is clear that the model in [15] encompasses as a special case the

Table 4.1: Example of Transition Probability $P(\mathbf{Y}'|X')$ for $b = 2$

X'	$\mathbf{Y}' = 00$	$\mathbf{Y}' = 01$	$\mathbf{Y}' = 10$	$\mathbf{Y}' = 11$	$\mathbf{Y}' = 0$	$\mathbf{Y}' = 1$
0 (00)	$1 - P_d$	0	0	0	P_d	0
1 (01)	0	$1 - P_d$	0	0	$P_d/2$	$P_d/2$
2 (10)	0	0	$1 - P_d$	0	$P_d/2$	$P_d/2$
3 (11)	0	0	0	$1 - P_d$	0	P_d

segmented deletion channel model (when the deletions occur independently in different segments). To illustrate this point further, let us give a simple example. Consider the segmented deletion channel with $b = 2$ and deletion probability of P_d. The equivalent channel transition matrix $P(\mathbf{Y}'|X')$ is as given in Table 4.1.

With the above explanation, from [15], we can safely say that the segmented deletion channel is information stable, and hence its Shannon capacity exists. In fact, the capacity per transmitted bit is given by

$$C = \lim_{T\to\infty} \frac{1}{T} \max_{P(\mathbf{X})} I(\mathbf{X}; \mathbf{Y}),$$

where $I(\cdot;\cdot)$ is the mutual information between the input sequence $\mathbf{X}$, of length T, and output sequence $\mathbf{Y}$.

Although the channel capacity exists, evaluation of the capacity expression is not straightforward. That is, there is no single-letter or finite-letter formulation which may be amenable for practical computation which is also the case for other channel models with synchronization errors. With this motivation, we next introduce two simple upper/lower bounds on the capacity of segmented deletion channels. First of all, an obvious capacity upper bound can be obtained by providing side information to the receiver about the positions of all the deletions. Therefore, the channel becomes a binary erasure channel with memory and an erasure probability P_d/b. Since the memory

does not affect the capacity of an erasure channel [68], $1 - P_d/b$ becomes a trivial upper bound on the channel capacity. To obtain a lower bound, we assume that a long interleaver has been introduced before transmission, and the corresponding deinterleaver is used at the receiver before decoding. The equivalent channel is then a binary i.i.d. deletion channel. Since this is a specific signaling scheme, any achievable rate over a binary i.i.d. deletion channel with probability P_d/b would be achievable on the segmented deletion channel providing us with a lower bound on the channel capacity.

4.1.2 Capacity Upper and Lower Bounds with Side Information

In [29], to obtain an upper bound on the capacity for an i.i.d. deletion channel, some suitable genie-aided information on the deletion process is revealed to the receiver so that the channel becomes memoryless. For the segmented deletion channel, we propose a similar method of obtaining upper and lower bounds on the capacity by providing some side information to both the transmitter and the receiver.

4.1.2.1 Upper Bound - Version 1

Define the random process $\mathbf{V} = \{V_n\}_{n=1}^{N}$, where V_n is a binary valued random variable which determines whether the n-th segment $\mathbf{X}_n$ experiences a deletion error or not. With the side information being provided to both the transmitter and receiver, we have

$$C \leq \frac{1}{b} \max_{P(\mathbf{X}_n)} I(\mathbf{X}_n; \mathbf{Y}_n) \ ,$$

where $\mathbf{Y}_n$ is the received sequence corresponding to $\mathbf{X}_n$ with length either b or $b-1$.

Obviously, $1 - P_d$ fraction of the blocks see noiseless channels, hence with the transmitter/receiver side information, we can transmit b bits with no error. The remaining P_d fraction of blocks will equivalently see a deletion channel with b input bits and exactly one deletion at the output. The capacity of such a channel can be computed (for reasonable values of b)[1] using the Blahut-Arimoto Algorithm (BAA) [69, 70]. Denoting the capacity of the deletion channel with b input bits and $b-1$ output bits as $C_d(b,1)$, we can write an upper bound on the capacity of segmented deletion channel as

$$C \leq 1 - P_d + P_d \frac{1}{b} C_d(b,1). \tag{4.1}$$

4.1.2.2 Upper Bound - Version 2

Following similar line of arguments, we expect the capacity upper bound to be tighter when "less" side information is provided to the transmitter and the receiver. For example, we define the random process $\mathbf{W} = \{W_{n'}\}_{n'=1}^{N/2}$, where $W_{n'}$ is a random variable taking values $\{0,1,2\}$, which determines the number of deletions in every pair of segments, i.e., in $\mathbf{X}_{2n'-1}$ and $\mathbf{X}_{2n'}$. When $W_{n'}$ equals 0 or 2, it contains the same information as in the previous case. Ambiguity only rises when $W_{n'} = 1$, since in this case, we have no idea which one of the two segments has the deleted bit, and we simply have a channel with $2b$ bits at the input and one deletion. Therefore, we can write the capacity upper bound as

$$\begin{aligned} C &\leq (1-P_d)^2 + 2P_d(1-P_d)\frac{1}{2b}C_d(2b,1) + P_d^2\frac{1}{2b}C_d(2b,2) \\ &= (1-P_d)^2 + P_d(1-P_d)\frac{1}{b}C_d(2b,1) + P_d^2\frac{1}{b}C_d(b,1), \end{aligned} \tag{4.2}$$

where $C_d(2b,2)$ denotes the capacity of a channel with $2b$ bits of input and one deletion in the first b bits and another one among last b bits. The second line

[1]The largest value of b we could handle in our computations was 24.

follows since for the channel with K segments of input bits and one deletion in each segment, we can deduce the boundaries of every segment in the received bit sequence without any additional information and hence, $C_d(Kb, K) = KC_d(b, 1)$. Comparing (4.1) and (4.2), it is obvious that with the random process $\mathbf{W}$, we are able to expand the capacity upper bound as a quadratic function of P_d and thus obtain a tighter result, as will be shown later. Even tighter bounds can be achieved when less and less side information is used at the expense of a much heavier computational load on the BAA algorithm. Details of this further generalization is omitted from this paper.

For large values of b that are not amenable for the BAA, one can resort to the upper bound $C_d(B, 1) \leq C_d(b, 1) + (n-1)b$ reported in [29], where $B = nb$. The bound is tight for large B as it is also shown that

$$C_d(B, 1) \geq C'_d(b, 1) + (n-1)b - H\left(\frac{1}{n}\right) , \tag{4.3}$$

where $H(\cdot)$ is the binary entropy function and $C'_d(b, 1)$ is the achievable rate for a deletion channel with b independent uniformly distributed (i.u.d.) input bits and exactly one deletion. The gap between the upper and lower bounds of $C_d(B, 1)$ gets smaller as n increases, since the entropy term $H(\frac{1}{n})$ approaches zero. When $B \neq nb$, another upper bound can also be used [29]:

$$C_d(B, 1) \leq \frac{B-1}{B}\left(2 + C_d(B-1, 1)\right) . \tag{4.4}$$

4.1.2.3 Lower Bounds

Capacity lower bounds can be obtained by revealing some side information to the receiver, and then by subtracting a certain term to make sure what is obtained is in fact a lower bound. Specifically, we can write

$$I(\mathbf{X}; \mathbf{Y}) = I(\mathbf{X}; \mathbf{Y}, \mathbf{V}) - I(\mathbf{X}; \mathbf{V}|\mathbf{Y}) \geq I(\mathbf{X}; \mathbf{Y}, \mathbf{V}) - H(\mathbf{V}) .$$

To compute $I(\mathbf{X};\mathbf{Y},\mathbf{V})$, we cannot optimize the input distribution for every segment, since the side information is only provided to the receiver. Instead, we consider i.u.d. inputs. Hence, the following capacity lower bound is obtained,

$$C \geq 1 - P_d + P_d \frac{1}{b} C'_d(b,1) - \frac{1}{b} H(P_d), \tag{4.5}$$

where $C'_d(b,1)$ refers to the achievable rates with i.u.d. inputs for a b bit input one deletion channel.

Comparing the capacity upper bound in (4.1) and lower bound in (4.5), we see that the difference is $P_d \frac{1}{b} \left(C_d(b,1) - C'_d(b,1)\right) + \frac{1}{b} H(P_d)$. When P_d approaches zero or one, the term $\frac{1}{b} H(P_d)$ tends to zero. In fact, when P_d equals zero or one, the segmented deletion channel becomes a memoryless channel without any synchronization problems, and the capacity is exactly as given in (4.1). Furthermore, for large b values, we would expect $\frac{1}{b} \left(C_d(b,1) - C'_d(b,1)\right)$ to be small. The reason is that since the i.u.d. input sequences are optimal for the calculation of $C_d(b,0)$, when the overall deletion rate per transmitted bit $\frac{1}{b}$ goes to zero, i.u.d. inputs will be close to optimal, and therefore, the gap between the upper and lower bound on the capacity becomes very small. This observation is quantified in [16], which proves that for an i.i.d. deletion channel with a small deletion probability, i.u.d. inputs achieve the first order term of the channel capacity when we express it as a series expansion in terms of the deletion probability.

Our final argument is that this approach can be easily extended to the case when more than one synchronization errors are allowed in each segment, i.e., the ideas are not limited to the elementary segmented deletion channel only.

4.1.3 Asymptotic Behavior of the Segmented Deletion Channel Capacity

We now focus on the case where P_d/b is small and characterize the capacity for a segmented deletion channel using a similar approach employed in [16] for the case of i.i.d. deletion channels. In particular, for a finite segment length b with P_d approaching zero, and for a fixed P_d with a large the segment length b, we show that the capacity can be characterized asymptotically, and therefore, an approximation to the exact channel capacity can be obtained for small P_d/b values.

It is proved in [16] that when computing the channel capacity or capacity bounds for an i.i.d. deletion channel, one can restrict the input sequence to be a stationary ergodic process $\mathbf{X} = \{X_i\}$ with $X_i \in \{0, 1\}$. We make an observation that the same argument also holds for the segmented deletion channel as all the steps in the proof remain valid for our case following a similar approach.

First of all, some notation is established. Following [16], let L be the length of the bit runs in the input sequence. Clearly, for the i.i.d. Bernoulli(1/2) process, $P_L(l) = \frac{1}{2^l}$, $E[L] = \sum_{l=0}^{\infty} l2^{-l} = 2$, and we refer $P_L(l)$ to as the block perspective run length distribution. We then introduce a new random variable L_0, whose distribution is given as

$$P(L_0 = l) = \frac{lP_L(l)}{E[L]}.$$

The random variable L_0 denotes, for an arbitrary transmitted bit, the length of the run it belongs to, which is called the bit perspective run length distribution of the input sequence [16]. Let $\mathbf{X}^n$ be the input sequence of length n and $\mathbf{Y} = \mathbf{Y}(\mathbf{X}^n)$ represent the corresponding output sequence. Define S_L to be

the set of stationary ergodic processes such that no run has length larger than L and $\mathbf{X}^*$ to be the i.i.d. Bernoulli(1/2) process, i.e., X_i^* equals 0 or 1 with probability 1/2 each. We define $I(\mathbf{X}^n) = \lim_{n\to\infty} \frac{1}{n} I(\mathbf{X}^n; \mathbf{Y}(\mathbf{X}^n))$, $H(\mathbf{X}) = \lim_{n\to\infty} \frac{1}{n} H(\mathbf{X}^n)$. Theorems 1-3 below present our main results. We note that the proofs of these theorems and the related lemmas are extensions of the corresponding ones in [16], which considers i.i.d. deletion channels.

Theorem 1. *Consider a segmented deletion channel with a fixed segment length b and deletion probability P_d approaching zero. We have $\forall \epsilon > 0$,*

$$\lim_{n\to\infty} \frac{1}{n} I(\mathbf{X}^{*,n}; \mathbf{Y}(\mathbf{X}^{*,n})) = 1 - \frac{P_d}{b}\left(1 + \log_2 b - A\right) - \frac{H(P_d)}{b} - O(P_d^{2-\epsilon}) \ , \quad (4.6)$$

where $A = \sum_{l=1}^{\infty} 2^{-l-1} l \log_2 l \approx 1.28853$, and $O(\cdot)$ is the standard big O notation. Clearly, this is an achievability result and serves as a lower bound on the capacity of the segmented deletion channel as $P_d \to 0$.

Theorem 2. *For a segmented deletion channel with a fixed segment length b and deletion probability P_d approaching zero, there exists $P_{d,0} > 0$ such that $\forall P_d < P_{d,0}$ and $\epsilon > 0$, for any input process we have*

$$\lim_{n\to\infty} \frac{1}{n} I(\mathbf{X}^n; \mathbf{Y}(\mathbf{X}^n)) \leq 1 - \frac{P_d}{b}\left(1 + \log_2 b - A\right) - \frac{H(P_d)}{b} + O(P_d^{3/2-\epsilon}) \ . \quad (4.7)$$

Clearly, the right-hand side serves as an upper bound on the capacity for the segmented deletion channel with a finite b and $P_d \to 0$.

Before proving the given theorems, we present two lemmas whose proofs are given in Appendix A.1.

Lemma 1. *For a segmented deletion channel with i.i.d. Bernoulli(1/2) process as the input, we have $\forall \epsilon > 0$,*

$$\lim_{n\to\infty} \frac{1}{n} H(\mathbf{Y}(\mathbf{X}^{*,n}) | \mathbf{X}^{*,n}) = \frac{P_d}{b} \log_2 b + \frac{H(P_d)}{b} - A\frac{P_d}{b} + O(P_d^{2-\epsilon}) \ . \quad (4.8)$$

Lemma 2. *For any $\epsilon > 0$, there exists $K < \infty$ and $P_{d,0} > 0$ such that $\forall P_d < P_{d,0}$ the following statement holds for the segmented deletion channel. For any positive integer L^*, if $\mathbf{X} \in S_{L^*}$, and $H(\mathbf{X}) \geq 1 - \left(\frac{P_d}{b}\right)^{1-\gamma}$ with $\gamma > 0$, then*

$$\lim_{n\to\infty} \frac{1}{n} H(\mathbf{Y}(\mathbf{X}^n)|\mathbf{X}^n) \geq \frac{P_d}{b}\log_2 b + \frac{H(P_d)}{b} - A\frac{P_d}{b} - KP_d^{2-\epsilon}(1+P_d^{1/2}L^*) \ . \quad (4.9)$$

Proof of Theorem 1. Without loss of generality, assume that n is a multiple of b. We have $I(\mathbf{X}^n;\mathbf{Y}) = H(\mathbf{Y}) - H(\mathbf{Y}|\mathbf{X}^n)$. With the i.i.d. Bernoulli(1/2) input process $\mathbf{X}^{*,n}$, for the output process $\mathbf{Y}(\mathbf{X}^{*,n})$, we obtain

$$\begin{aligned} H(\mathbf{Y}(\mathbf{X}^{*,n})) &= -\sum_{\mathbf{y}} P(\mathbf{y})\log_2(P(\mathbf{y})) \\ &= -\sum_{m=0}^{n/b} \binom{n/b}{m}(1-P_d)^{n/b-m}P_d^m \\ &\qquad \cdot \log_2\left(\frac{1}{2^{n-m}}\binom{n/b}{m}(1-P_d)^{n/b-m}P_d^m\right) \\ &= n\Big(1-\frac{P_d}{b}\Big) + H_T, \end{aligned} \quad (4.10)$$

where $\mathbf{y}$ and m represent the realization of process $\mathbf{Y}$ and the corresponding total number of deletions in $\mathbf{y}$, respectively. The term $H_T = -\sum_{m=0}^{n/b}\binom{n/b}{m}(1-P_d)^{n/b-m}P_d^m \log_2\left(\binom{n/b}{m}(1-P_d)^{n/b-m}P_d^m\right) = \frac{1}{2}\log_2(2\pi e\frac{n}{b}P_d(1-P_d)) + o(1) = O(\log_2 n)$ (Corollary 1 of [71]). The proof follows by combining the results of $H(\mathbf{Y}(\mathbf{X}^{*,n}))$ and $H(\mathbf{Y}(\mathbf{X}^{*,n})|\mathbf{X}^{*,n})$ given in Lemma 1. □

Proof of Theorem 2. It is clear that for any input $\mathbf{X}^n$, the number of deletions is Binomial$(n/b, P_d)$ distributed, leading to

$$\lim_{n\to\infty} \frac{1}{n} H(\mathbf{Y}(\mathbf{X}^n)) \leq 1 - \frac{P_d}{b} \ , \quad (4.11)$$

where the equality is achieved when input sequence is i.i.d. Bernoulli(1/2) distributed. In light of Theorem 1, for i.u.d inputs $I(\mathbf{X}^{*,n}) > 1 - \left(\frac{P_d}{b}\right)^{1-\gamma}$, $\gamma > 0$, therefore, we only need to consider stationary ergodic processes with $H(\mathbf{X}) \geq I(\mathbf{X}^{*,n}) > 1 - \left(\frac{P_d}{b}\right)^{1-\gamma}$ when computing the upper bounds on the capacity. Combining (4.11) and Lemma 2, we obtain an upper bound on $I(\mathbf{X}_{L^*})$ for $\mathbf{X}_{L^*} \in S_{L^*}$, which is constructed from $\mathbf{X}$ by flipping the (L^*+1)-th bit in each run with a length longer than L^*, until no run length exceeds L^*.

Next, we show that we do not lose much with this restriction for large enough L^* values. Let $\mathbf{F}$ be the vector (of the same length as $\mathbf{Y}(\mathbf{X}^n)$) taking values of 1 wherever the corresponding bit in $\mathbf{Y}(\mathbf{X}^n_{L^*})$ is flipped and 0 otherwise. From [16] (Eqn. (27) and Eqn. (28)), we have $|H(\mathbf{Y}(\mathbf{X}^n)) - H(\mathbf{Y}(\mathbf{X}^n_{L^*}))| \leq H(\mathbf{F})$, $|H(\mathbf{Y}(\mathbf{X}^n)|\mathbf{X}^n) - H(\mathbf{Y}(\mathbf{X}^n_{L^*})|\mathbf{X}^n_{L^*})| \leq H(\mathbf{F})$ and also $\lim_{n\to\infty} \frac{1}{n} H(\mathbf{F}) \leq \left(\frac{P_d}{b}\right)^{1/2-\epsilon'} \log_2 L^*/2L^* \ \forall\ \epsilon' > 0$, if $L^* > \log_2(b/P_d)$. Therefore, there exists $\mathbf{X}_{L^*} \in S_{L^*}$ such that

$$|I(\mathbf{X}) - I(\mathbf{X}_{L^*})| \leq \left(\frac{P_d}{b}\right)^{1/2-\epsilon'} \log_2 L^*/L^* \,, \tag{4.12}$$

Combining (4.11), (4.12), Lemma 2 and taking $L^* = \lfloor \frac{1}{P_d} \rfloor$, we get the claim. □

Theorems 1 and 2 give the asymptotic capacity for an elementary segmented deletion channel with a fixed segment length b for small P_d values. For a fixed $P_d > 0$ and a large segment length b, we have a different characterization.

Theorem 3. *For a fixed P_d, for any $\epsilon > 0$, there exists $b_0 > 0$, such that $\forall b > b_0$, the following statement holds for the segmented deletion channel,*

$$\lim_{n\to\infty} \frac{1}{n} I(\mathbf{X}^{*,n}; \mathbf{Y}(\mathbf{X}^{*,n})) \geq 1 - \frac{P_d}{b}\,(1 + \log_2 b - A) - \frac{H(P_d)}{b} - O(b^{-2+\epsilon}) \,, \tag{4.13}$$

where $\mathbf{X}^*$ *is the Bernoulli(1/2) process, and*

$$\lim_{n\to\infty}\frac{1}{n}I(\mathbf{X}^n;\mathbf{Y}(\mathbf{X}^n)) \le 1-\frac{P_d}{b}\left(1+\log_2 b - A\right) - \frac{H(P_d)}{b} + O(b^{-3/2+\epsilon}) \ , \quad (4.14)$$

where $\mathbf{X}$ *is any input process.*

Before the proof of the theorem, a lemma is given (whose proof is in Appendix A.3) is given .

Lemma 3. *For any stationary ergodic process* $\mathbf{X} \in S_b$ *with* $H(\mathbf{X}) \ge 1 - \left(\frac{P_d}{b}\right)^{1-\gamma}$ $\gamma > 0$*, and any* $\epsilon > 0$*, there exists* $\kappa < \infty$ *and* $b_0 > 0$*, such that* $\forall b > b_0$

$$\lim_{n\to\infty}\frac{1}{n}H(\mathbf{Y}(\mathbf{X}^n)|\mathbf{X}^n) \ge \frac{P_d}{b}\log_2 b + \frac{H(P_d)}{b} - A\frac{P_d}{b} - \kappa b^{-2+\epsilon}(1+b^{1/2}) \ . \quad (4.15)$$

Specifically, consider an an i.i.d. Bernoulli(1/2) process $\mathbf{X}^*$*. By flipping the* $(b+1)$*-th bit in each run with a length longer than* b*, until no run length exceeds* b*, we obtain a modified process* $\mathbf{X}_b^* \in S_b$*. We can show that*

$$\lim_{n\to\infty}\frac{1}{n}H(\mathbf{Y}(\mathbf{X}_b^{*,n})|\mathbf{X}_b^{*,n}) = \frac{P_d}{b}\log_2 b + \frac{H(P_d)}{b} - A\frac{P_d}{b} + O(b^{-2+\epsilon}) \ . \quad (4.16)$$

Proof of Theorem 3. From (4.10) and (4.16), we have

$$I(\mathbf{X}_b^*) = 1 - \frac{P_d}{b}\left(1+\log_2 b - A\right) - \frac{H(P_d)}{b} - O(b^{-2+\epsilon}) \ . \quad (4.17)$$

As in [16], define $\alpha = P(L_0 > b)/b$, which is the upper bound of the density of bits in $\mathbf{X}^*$ to be flipped to ensure no run length exceeds b. For an i.i.d. Bernoulli(1/2) process, we have $\alpha = \frac{1}{b}\sum_{l=b+1}^{\infty} l/2^{l+1} = (1+\frac{2}{b})2^{-b-1}$. Therefore, $\lim_{n\to\infty}\frac{1}{n}H(\mathbf{F}) \le H(\alpha) = O(b^{-\zeta})$ with $\zeta > 2$, where $\mathbf{F}$ has the same definition as the one in the proof of Theorem 2. Following the same

steps leading to (4.12), we can write, $|I(\mathbf{X}^*) - I(\mathbf{X}_b^*)| = O(b^{-\zeta})$. Combining this result with (4.17), the lower bound on the capacity given in (4.13) is proved.

To obtain the upper bound, again, in light of the achievability result, we only consider stationary and ergodic processes with $H(\mathbf{X}) \geq 1 - \left(\frac{P_d}{b}\right)^{1-\gamma}$, $\gamma > 0$. Under this condition, (4.12) still holds. Taking $L^* = b$, we conclude that $|I(\mathbf{X}) - I(\mathbf{X}_b)| = O(b^{-1.5+\epsilon})$. Combining this result with (4.11) and (4.15) (which provides the upper bound on $I(\mathbf{X}_b)$ for $\mathbf{X}_b \in S_b$), we get the claim. □

From the above theorems, we conclude that the channel capacity for segmented deletion channel as $\frac{P_d}{b} \to 0$ is dominated by the expression

$$C_{\text{est}} = 1 - \frac{P_d}{b}\left(1 + \log_2 b - A\right) - \frac{H(P_d)}{b}, \tag{4.18}$$

where $A \approx 1.28853$.

4.2 Concatenated Coding over Segmented Deletion Channels

We now focus on a practical channel coding scheme suitable for segmented deletion channels. The proposed encoding and decoding procedure is the same as the ones developed in Chapter 3. Optimized marker code structure and rate can be found by choosing the ones with the maximum achievable rates.

Let $\mathbf{x}_1^T = \{x_k\}_{k=1}^T$ and $\mathbf{y}_1^R = \{y_n\}_{n=1}^R$ be the sequences of bits at the channel input and channel output, respectively, where the number T of transmitted bits is a constant system parameter. We assume $T = Nb$, where N is the total number of segments. Since the channel is an elementary segmented deletion channel, the number R of received bits is a random variable taking values in the set $\{T - N, T - N + 1, \ldots, T\}$, depending on the realization of

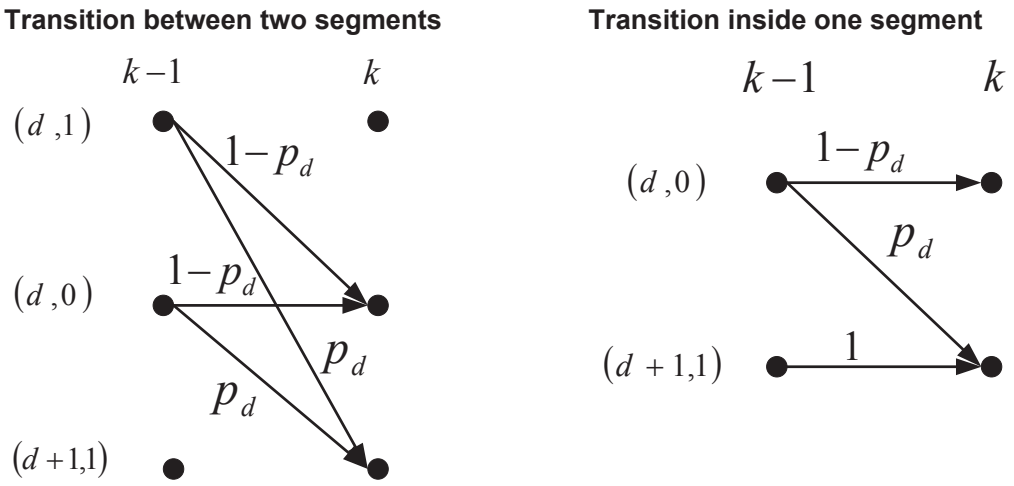

Figure 4.1: Trellis for bit-level MAP detection.

the deletion process. The transmitter and the receiver have no information on the positions of the deletions.

In Chapter 3, MAP detection algorithm was specifically designed for i.i.d. deletion channels. This detector can be directly applied to a segmented deletion channel with a deletion probability for each bit set to $p_d = P_d/b$. However, this would be a sub-optimal choice since it ignores the additional information due to the segmentation assumption. For example, if the detector determines that the first bit of a segment is deleted, we can naturally deduce that there will be no error in the next $b-1$ bits. In the following sections, we describe two other detectors that take the additional segmentation assumption into consideration and provide improved results.

4.2.1 Improved Bit Level Synchronization

Let us introduce a trellis diagram, as shown in Fig. 4.1, with the state of trellis at time k (when x_k is transmitted) defined to be $s_k = (d_k, i)$. The term d_k denotes the number of deletions at time k and i is an indicator, where

$i = 0$ when no deletion occurs in the segment, and $i = 1$ otherwise. The transition probability from one state to another state is determined by the bit-wise deletion probability, which is set to be $p_d = P_d/b$. When x_k is not the first bit of a segment, transition for state $(d, 1)$ to $(d+1, 1)$ or $(d+1, 0)$ is prohibited since there is already one bit been deleted in the segment.

Similar to [72], we define the function $F(x_k, y_n) = \begin{cases} 1 & \text{if } y_n = x_k \\ 0 & \text{if } y_n \neq x_k \end{cases}$, and also the sets of forward/backward variables in the usual sense, $\alpha_k(s_k) = P(\mathbf{y}_1^{k-d_k}, s_k)$, $\beta_k(s_k) = P(\mathbf{y}_{k-d_k+1}^{R}|s_k)$. These coefficients can be computed by means of the following forward/backward recursion [8]:

Case 1: x_k is the first bit of the segment:

$$\begin{aligned}\alpha_k(s_k) &= P\left(s_k = (d_k, i), \mathbf{y}_1^{k-d_k}\right) \\ &= (1-i)(1-p_d)\big(\alpha_{k-1}(s_k) + \alpha_{k-1}(d_k, 1)\big)\sum_{x_k} P(x_k)F(x_k, y_{k-d_k}) \\ &\quad + ip_d\big(\alpha_{k-1}(d_k - 1, 1) + \alpha_{k-1}(d_k - 1, 0)\big), \end{aligned} \tag{4.19}$$

$$\begin{aligned}\beta_{k-1}(s_{k-1}) &= P\left(\mathbf{y}_{k-1-d_{k-1}+1}^{R}|s_{k-1} = (d_{k-1}, i)\right) \\ &= (1-i)\Big(p_d\beta_k(d_{k-1}+1, 1) \\ &\qquad + (1-p_d)\beta_k(s_{k-1})\sum_{x_k} P(x_k)F(x_k, y_{k-d_k})\Big) \\ &\quad + i\Big((1-p_d)\beta_k(d_{k-1}, 0)\sum_{x_k} P(x_k)F(x_k, y_{k-d_k}) \\ &\qquad + p_d\beta_k(d_{k-1}+1, 1)\Big), \end{aligned} \tag{4.20}$$

Case 2: x_k is not the first bit of the segment:

$$\begin{aligned}\alpha_k(s_k) =& \big(1 - p_d(1-i)\big)\alpha_{k-1}(s_k)\sum_{x_k} P(x_k)F(x_k, y_{k-d_k}) \\ &+ ip_d\alpha_{k-1}(d_k - 1, 0), \end{aligned} \tag{4.21}$$

$$\beta_{k-1}(s_{k-1}) = \big(1 - p_d(1-i)\big)\beta_k(s_{k-1}) \sum_{x_k} P(x_k) F(x_k, y_{k-d_k})$$
$$+ (1-i) p_d \beta_k(d_{k-1}+1, 1). \tag{4.22}$$

We are interested in the exact "frame synchronization" scenario, leading to

$$\alpha_0(s_k) = \begin{cases} 1 & \text{if } s_k = (0,0) \\ 0 & \text{otherwise} \end{cases}, \quad \beta_T(s_k) = \begin{cases} 1 - P_d & \text{if } s_k = (T-R, 0) \\ P_d & \text{if } s_k = (T-R, 1) \\ 0 & \text{otherwise} \end{cases}.$$

Finally, the target probability can be computed as

Case 1:

$$P(\mathbf{y}_1^R | x_k) = (1-p_d) \sum_{d_k=0}^{k/b} \sum_{i=0}^{1} \alpha_{k-1}(d_k, i) \beta_k(d_k, 0) F(x_k, y_{k-d_k})$$
$$+ p_d \sum_{d_k=0}^{k/b} \sum_{i=0}^{1} \alpha_{k-1}(d_k - 1, i) \beta_k(d_k, 1), \tag{4.23}$$

Case 2:

$$P(\mathbf{y}_1^R | x_k) = \sum_{d_k=0}^{k/b} \sum_{i=0}^{1} \big(1 - p_d(1-i)\big) \alpha_{k-1}(d_k, i) \beta_k(d_k, i) F(x_k, y_{k-d_k})$$
$$+ p_d \sum_{d_k=0}^{k/b} \alpha_{k-1}(d_k - 1, 0) \beta_k(d_k, 1). \tag{4.24}$$

4.2.2 Symbol Level Synchronization

The MAP detection algorithms we described in the previous subsections is not optimal for two reasons, the bit-level interleaving and the inaccurate approximation of p_d. However, a symbol-level MAP detector can be easily applied under this scenario by treating one segment as a symbol.

Let us define the binary event $D_{k,n}$, with $k \in \{1, 2, \ldots, N\}$ and $n \in \{1, 2, \ldots, R\}$, which denotes whether, of the first k transmitted segments of

bits, exactly n bits are received or not. Thanks to the assumption of 1-deletion per segment, symbol-level MAP detection becomes feasible for large values of b, and the forward/backward recursions are given as follows:

$$\begin{aligned}\alpha_k(n) &= P(\mathbf{y}_1^n, D_{k,n}) \\ &= P_d\,\alpha_{k-1}(n-b+1)\sum_{j=0}^{b-1}\prod_{\substack{i=0\\ i\neq j}}^{b-1}\sum_{x_{bk-i}} P(x_{bk-i})F(x_{bk-i}, y_{n-i'}) \\ &\quad + (1-P_d)\,\alpha_{k-1}(n-b)\prod_{i=0}^{b-1}\sum_{x_{bk-i}} P(x_{bk-i})F(x_{bk-i}, y_{n-i}) \quad (4.25)\end{aligned}$$

and

$$\begin{aligned}\beta_k(n) &= P(\mathbf{y}_{n+1}^R | D_{k,n}) \\ &= P_d\,\beta_{k+1}(n+b-1)\sum_{j=1}^{b}\prod_{\substack{i=1\\ i\neq j}}^{b}\sum_{x_{bk+i}} P(x_{bk+i})F(x_{bk+i}, y_{n+i'}) \\ &\quad + (1-P_d)\,\beta_{k+1}(n+b)\prod_{i=1}^{b}\sum_{x_{bk+i}} P(x_{bk+i})F(x_{bk+i}, y_{n+i}) \quad (4.26)\end{aligned}$$

respectively, where $i' = i$ when $i < j$ and $i' = i-1$ when $i > j$. The final soft output information is generated as

$$\begin{aligned}p(\mathbf{y}_1^R | x_{bk-1}, \ldots, x_{bk}) \;=\; & P_d \sum_{n=0}^{\min(bk,R)}\sum_{j=0}^{b-1}\alpha_{k-1}(n-b+1)\beta_k(n)\prod_{\substack{i=0\\ i\neq j}}^{b-1}F(x_{bk-i}, y_{n-i'}) \\ & +(1-P_d)\sum_{n=0}^{\min(bk,R)}\alpha_{k-1}(n-b)\beta_k(n)\prod_{i=0}^{b-1}F(x_{bk-i}, y_{n-i}). \quad (4.27)\end{aligned}$$

Note that both the bit-level and symbol-level synchronization algorithms can be extended to the generalized segmented deletion channel. For instance, consider at most two deletion errors are allowed in each segment. For the bit-level synchronization algorithm, the indicator i now should take values of 0, 1 and 2 and the trellis in Fig. 4.1 needs to be modified accordingly. For the symbol-level synchronization algorithm, the only necessary change is

to consider one more state in the FBA algorithm, e.g., add $\alpha_{k-1}(n-b+2)$ in the forward recursion.

4.2.3 Computational Complexity Comparisons

For the sake of computational safety, all the calculations of MAP detection algorithm are implemented in the log domain to avoid numerical instability. Therefore, instead of the multiplication operation, the most time-consuming part becomes log domain addition, denoted as `log_add`. To compare the complexity of the two algorithms, in this section, we use the number of `log_add` operations required as a metric.

Consider the symbol-level MAP detection with T bits inputs and R bits output, the size of the trellis diagram is $(R+1) \times (N+1)$, where $N = T/b$. For every time instance we only care about $T-R+1$ states instead of all the $R+1$ states, since the maximum bits allowed to be deleted is $T-R$. From (4.25), computation of each forward quantity needs $2b+2$ `log_add` operations. Therefore, there are altogether $N(T-R+1)(2b+2)$ `log_add` operations for the forward recursion as well as for the backward recursion. For the same reason, to generate output soft information in (4.27), approximately $2^b N(T-R+1)(b+1)$ `log_add` operations needed for the symbol-level MAP detection.

For the bit-level MAP detection, the size of the trellis diagram is $2(T-R+1) \times (T+1)$. Computation of each forward quantity needs 2 or 4 `log_add` operations for (6.20) and 3 or 2 operations for (4.21), depending on the value of i. Hence, on average, total number of $T(T-R+1)(5b+1)/b = N(T-R+1)(5b+1)$ `log_add` operations required for the forward recursion and the same observation holds for the backward recursion. Following the same logic, approximately $2T(T-R+1)(3b+1)/b = 2N(T-R+1)(3b+1)$

Table 4.2: Capacity Upper Bounds Comparison for $b \leq 15$.

	$C \leq$	
b	UB (4.1)	UB (4.2)
2	$1 - 0.5P_d$	$1 - 0.915P_d + 0.445P_d^2$
3	$1 - 0.510P_d$	$1 - 0.794P_d + 0.284P_d^2$
4	$1 - 0.458P_d$	$1 - 0.694P_d + 0.236P_d^2$
5	$1 - 0.428P_d$	$1 - 0.617P_d + 0.189P_d^2$
6	$1 - 0.397P_d$	$1 - 0.555P_d + 0.158P_d^2$
7	$1 - 0.370P_d$	$1 - 0.507P_d + 0.137P_d^2$
8	$1 - 0.347P_d$	$1 - 0.466P_d + 0.120P_d^2$
9	$1 - 0.326P_d$	$1 - 0.433P_d + 0.107P_d^2$
10	$1 - 0.308P_d$	$1 - 0.405P_d + 0.097P_d^2$
11	$1 - 0.292P_d$	$1 - 0.380P_d + 0.089P_d^2$
12	$1 - 0.277P_d$	$1 - 0.362P_d + 0.085P_d^2$
13	$1 - 0.264P_d$	$1 - 0.314P_d + 0.050P_d^2$
14	$1 - 0.253P_d$	$1 - 0.275P_d + 0.023P_d^2$
15	$1 - 0.242P_d$	$1 - 0.245P_d + 0.001943P_d^2$

`log_add` operations needed for the bit-level MAP detector to generate output soft information, .

It is clear that the number of deletions, $T - R \sim TP_d/b$. Therefore, the recursions require similar computation load for both detectors, i.e., the number of `log_add` operations equals $O(T^2/b)$. Difference lies in the generation of the soft information. As expected, complexity of symbol-level MAP detection grows exponentially with b while the one for the bit-level MAP detector only depends on the length of codeword, i.e., T.

4.3 Numerical Examples of the Elementary Segmented Deletion Channels

In this section, we first list some numerical results of the approximation and upper/lower bounds on the capacity of the elementary segmented deletion channels. Comparison of the bit-level and symbol-level synchronization algorithms is also provided along with some results on the outer LDPC code design [72] for this channel model.

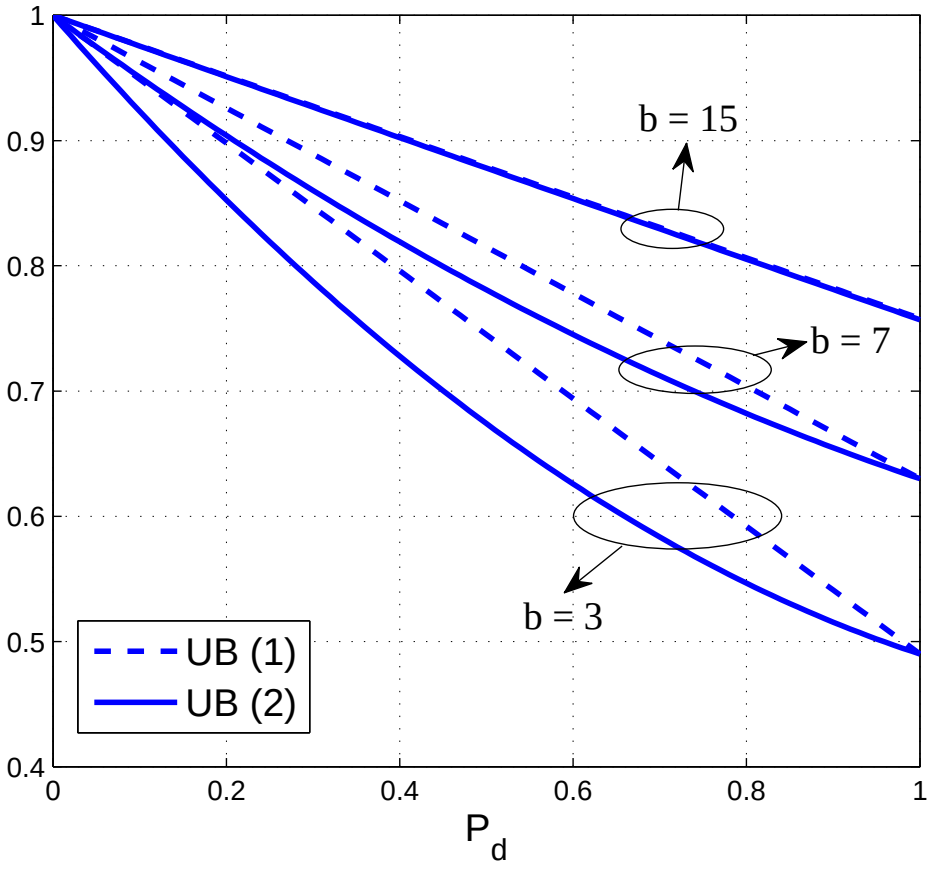

Figure 4.2: Capacity Upper Bound Comparison for $b = 3,\ 7,\ 15$.

4.3.1 Examples for Capacity results

In this subsection, some explicit results on the capacity bounds are provided as a function of P_d and b. First of all, using BAA, the largest value of b we can handle for the calculation of $C_d(b, 1)$ is 24, resulting in $C_d(24, 1) = 19.65$, therefore from (4.1), $C \leq 1 - 4.35\ \frac{P_d}{b},\ \forall\ b \geq 24$. The two versions of upper bounds in Section 4.1.2 is compared in Table 4.2 and Fig. 4.2. In Table 4.2, we compute the upper bounds in (4.1) and (4.2) for the case of $2 \leq b \leq 15$. For the second upper bound (4.2), since we could not obtain exact values of $C_d(2b, 1)$ when $b > 12$, we resort to (4.4). Fig. 4.2 compares the capacity upper

Table 4.3: Capacity Bounds Comparison

	$b = 3$			$b = 12$		
P_d	LB (4.5)	C_{est}	UB (4.1)	LB (4.5)	C_{est}	UB (4.1)
0.001	0.99557	0.99576	0.99949	0.99876	0.99877	0.99972
0.01	0.96688	0.96874	0.99493	0.99039	0.99052	0.99721
0.05	0.87361	0.88292	0.97466	0.96179	0.96239	0.98608
0.1	0.78182	0.80045	0.99972	0.93223	0.93344	0.97217
0.2	0.63566	0.67292	0.89866	0.88247	0.88489	0.94434
0.3	0.52069	0.57659	0.84799	0.84051	0.84414	0.91652
0.5	0.35743	0.45059	0.74665	0.77326	0.77931	0.86086
0.75	0.26572	0.40546	0.61997	0.71728	0.72636	0.79130
1	0.38153	0.56785	0.49330	0.71319	0.72529	0.72173

bound for $b = 3$, 7 and 15. As expected, the improvement is more obvious as b decreases. Another observation is that when $b > 15$, it makes no sense to use (4.2), as the bound on $C_d(2b, 1)$ becomes very loose.

We present C_{est} for different segment lengths in Fig. 4.3. The result illustrates that for the same value of P_d/b, segmented deletion channels with a larger b offer a higher capacity. Comparison of upper/lower bounds from Section 4.1.2 and C_{est} is provided in Fig. 4.4 for $b = 12$. It is clear that the lower bound remains tight up to around $P_d = 0.4$ while the upper bound is quite loose. When $P_d/b = 0.08333$, i.e., $P_d = 1$, every segment has deletion errors, and the decoupling of different segments is possible without any side information. As discussed before, the upper bound gives the exact value of capacity and C_{est}, although exceeds the capacity as given in Table 4.3, still remains close to it. We also observe that both the lower bound and C_{est} are not monotonically decreasing and there is a "tail" like behavior close to $P_d = 1$. It is not a surprising result, as the deletion rate approaches unity, segment-level synchronization becomes less critical and almost every segment has deletion errors. In this case, a higher capacity may be achieved as the synchronization errors become less and less important.

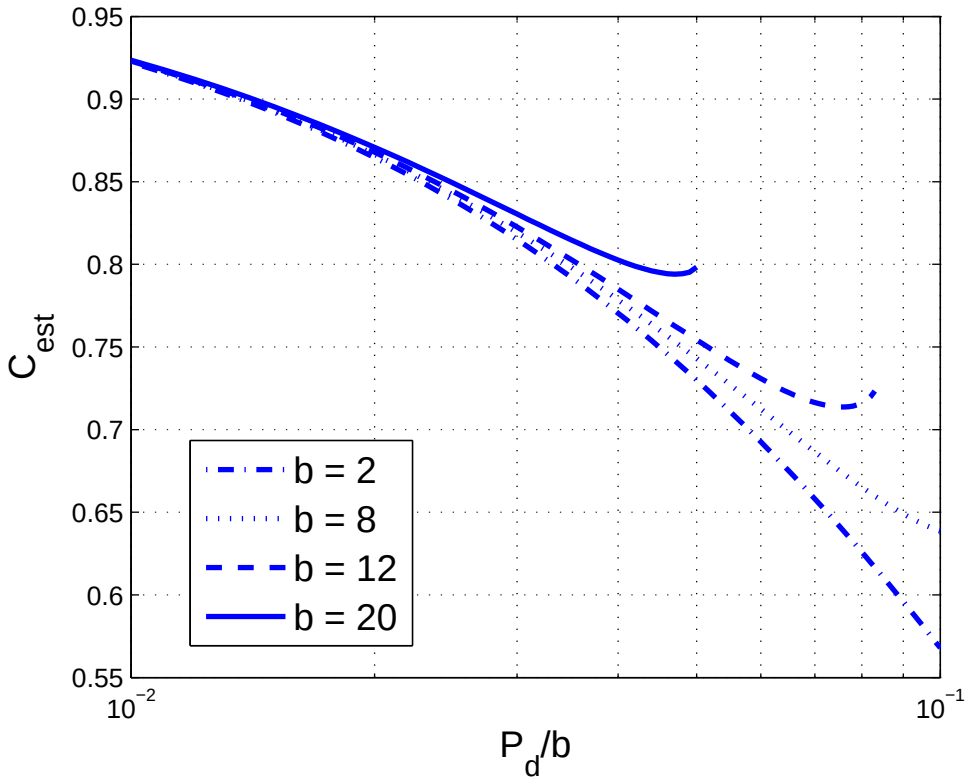

Figure 4.3: Estimate of the segmented deletion capacity (C_{est}) for small P_d/b.

4.3.2 Detection/decoding Results

We first consider practical coding schemes with the aim of confirming the performance gain over the existing techniques. The only reported practical coding scheme is introduced in [17], where for $b = 8$ the code rate is 0.448. This code is able to achieve zero error when at most one deletion error occurs per segment. Although codes with higher rates are also provided which allow for random errors, we will not consider them in our paper, since it is assumed that some information generated from the transmitted sequence, e.g., parity check bits, is known at the receiver, which requires another perfect (side) channel to communicate.

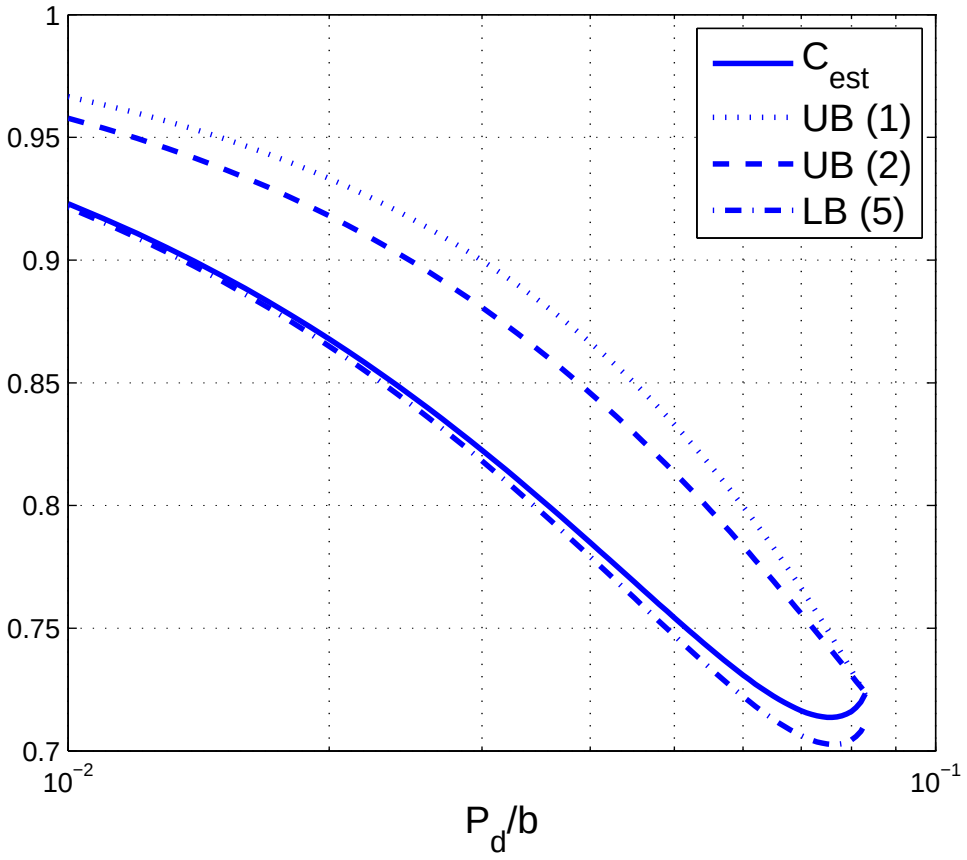

Figure 4.4: Comparison of upper and lower bounds on the segmented deletion channel capacity for $b = 12$.

In Fig. 4.5, we compare the BER performance of several detectors with single-pass decoding, i.e., MAP detection for synchronization purpose is only executed once. We adopt a binary LDPC code with rate 0.78, length 4521 and insert the marker "01" every 15 LDPC-coded bits. Obviously, symbol-level MAP detection with iterative soft demapping [66] outperforms other detectors. However for large b, it becomes infeasible. One solution is to consider only the M largest soft values among the 2^b outputs as for greedy multiuser detection algorithm [73]. Another observation is that the bit-level MAP detector for i.i.d. deletion channel [72] works well at low deletion rate. With

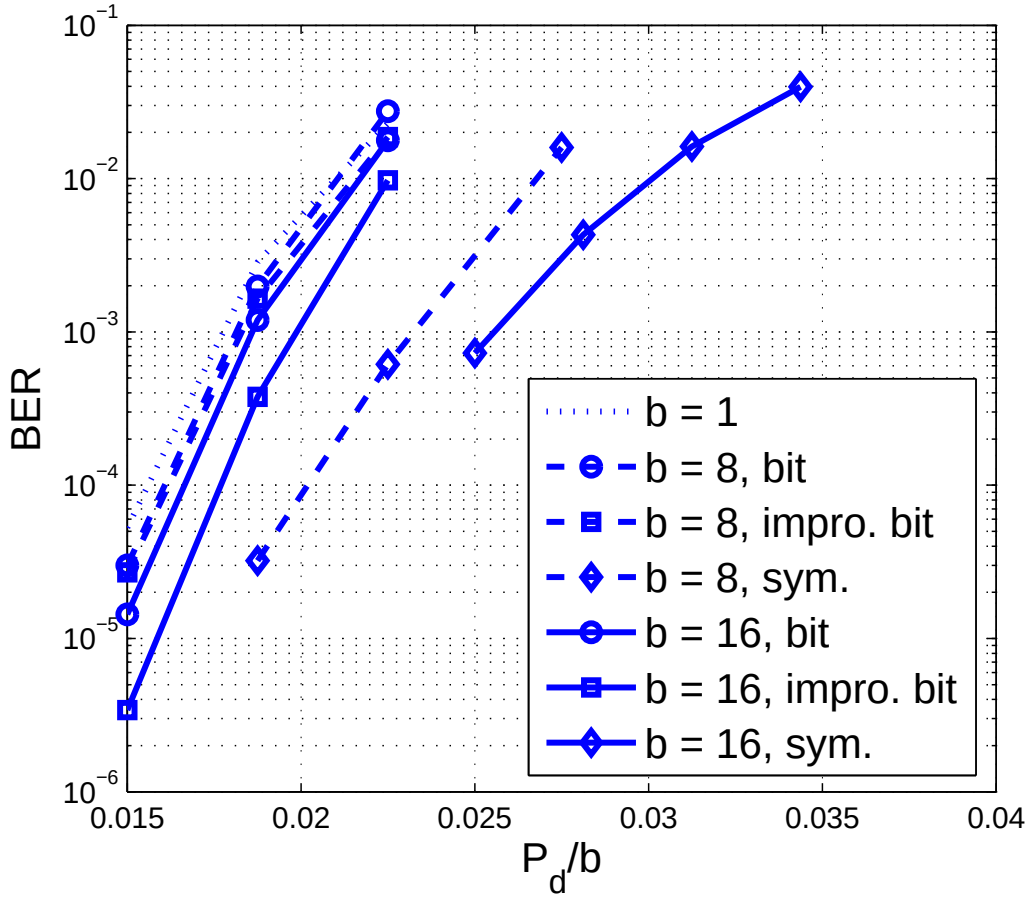

Figure 4.5: BER performances of different MAP detectors.

the same overall code rate $R = 0.693$ and single-pass decoding, the bit-level MAP detector for i.i.d. deletion channel almost provide the same performance compared to the one discussed in Section 4.2. This is not a surprising result since the segmentation assumption may not provide additional information to the detector due to the limited number of deletions. Our final comment is that when the segment length b is increased for the same average bit deletion probability P_d/b, the error probability is lower (which is a parallel to the findings (e.g., in terms of capacity bounds) to the results in the paper).

Table 4.4: Example LDPC Code Parameters for Segmented Deletion Channels

	b	r_M	r_C	d_c	d_v	a
$P_d = 0.1$	8	0.9	0.9423	52	{2 3 71}	{0.5667 0.425 0.0083}
$P_d = 0.3$	8	0.8333	0.8636	22	{2 3 42}	{0.5284 0.458 0.0136}
$P_d = 0.5$	8	0.75	0.8	15	{2 3 16}	{0.3936 0.576 0.0304}
$P_d = 0.7$	8	0.7143	0.75	12	{2 3 14}	{0.3354 0.634 0.0306}
$P_d = 0.9$	8	0.7143	0.7	10	{2 3 12}	{0.4301 0.522 0.0479}
$P_d = 0.2$	16	0.9	0.9444	54	{2 3 64}	{0.6385 0.351 0.0105}
$P_d = 0.4$	16	0.8571	0.9062	32	{2 3 31}	{0.5493 0.431 0.0197}
$P_d = 0.6$	16	0.8	0.875	24	{2 3 16}	{0.4206 0.547 0.0324}
$P_d = 0.8$	16	0.7778	0.8421	19	{2 3 14}	{0.395 0.569 0.036}
$P_d = 1$	16	0.75	0.8	15	{2 3 14}	{0.3318 0.638 0.0302}

4.3.3 LDPC Code Design Examples

As discussed in Chapter 3, the design of LDPC codes for insertion/deletion channels can rely on utilizing the EXIT charts [10] to predict the error rate when iterative decoding algorithm is applied. For the MAP detectors described in Section 4.2, let I_V and I_S be the mutual information between the LDPC-coded bits and the corresponding input/output soft values (log-likelihood ratios), respectively. It is shown in [10] that when the detection EXIT chart, which describes the relationship between output I_S and input I_V, is non-flat, i.e., each received symbol depends on multiple transmitted symbols, LDPC code design for this case is beneficial. For the segmented deletion channel, since it is not memoryless, instead of using randomly picked LDPC codes as in Fig. 4.5 or the ones optimized for the AWGN channels (with a flat detection EXIT chart), specially designed LDPC codes can provide a better performance.

Consider a check-regular LDPC code with constant check node degree d_c. Let I to be the total number of different variable node degrees of the LDPC

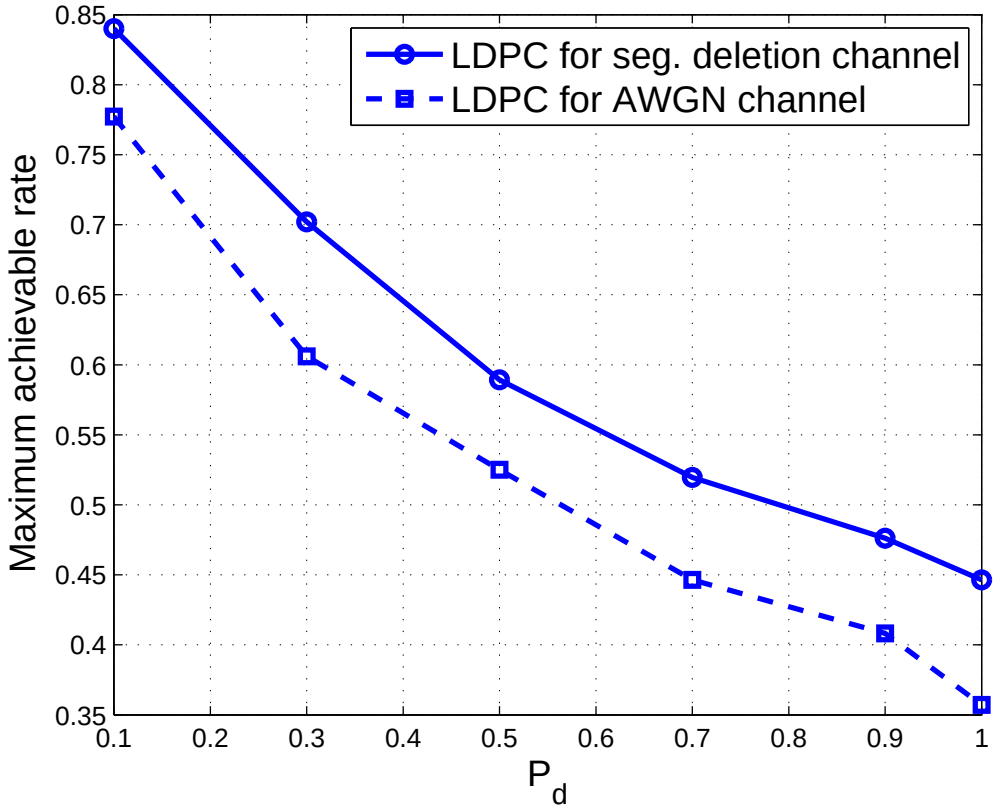

Figure 4.6: Decoding improvement through symbol-level decoding.

code denoted by $d_{v,i},\ i = 1, \ldots, I$ and a_i to be the fraction of variable nodes with degree $d_{v,i}$. The goal of code design for a fixed code rate r_C is to find the set of parameters $a_i,\ d_{v,i}$ and d_c which provide the best detection/decoding performance.

Some optimized codes are listed in Table 4.4 with an average variable node degree of [10] $\bar{d}_v = 3$ and r_M is the optimized 2-bit marker code rate obtained using a similar approach as in [72]. In Fig. 4.6, the highest achievable code rates for the concatenated coding scheme are plotted as a function of P_d for $b = 8$. The solid line denotes the achievable rates when LDPC codes from Table 4.4 are used while the dashed line represents the case for codes optimized for the AWGN channels. As the deletion rate increases, the rate

Table 4.5: Rates for Simulated Codes

	r_M	r_C	overall code rate
Code 1	0.833	0.863	0.719
Code 2	0.75	0.8	0.6
Code 3	0.714	0.75	0.5357

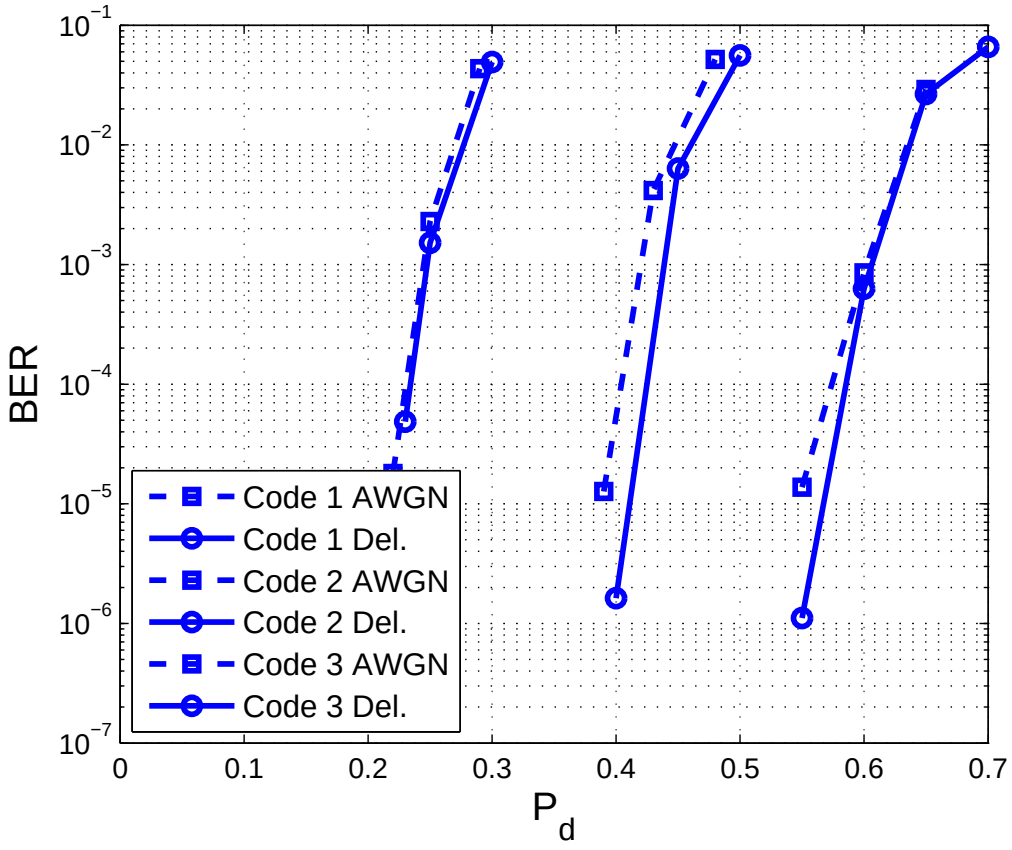

Figure 4.7: Error rate performances for some codes.

drops from 0.84 to 0.446 bits/channel use. Compared to the codes in [17], we can always achieve a higher code rate for $P_d < 1$ due to the more sophisticated detector/decoder configuration and possibility of arbitrarily low error probabilities (instead of no-errors).

To further illustrate the advantage of the designed codes, we pick several codes from Table 4.4, each of length 10000, and depict their error rate

performance in Fig. 4.7 using the bit-level synchronization algorithm over the segmented deletion channel. Again, performance of LDPC codes optimized for the AWGN channels are given in dashed lines (of the same rate but different variable/check node distributions). Parameters of Code 1, 2 and 3 for the segmented deletion channel are given in the second, third and forth row of Table 4.4, and their overall code rates are 0.719, 0.6 and 0.5337, respectively. It is obvious that the specifically designed outer LDPC codes for the segmented deletion channel offer a better performance. We also observe that the concatenated coding scheme can achieve a higher code rate when $\frac{P_d}{b}$ gets smaller. We note, however, that the results obtained are not very close to the capacity bounds. For instance, if we consider an error rate of 10^{-3} as reliable communications, from Fig. 4.7, the corresponding P_d for these three codes are 0.24, 0.44 and 0.6, while the corresponding capacity lower bounds are 0.8127, 0.7152 and 0.6589, respectively. A difference of 0.1 bits/channel use exist between the capacity lower bounds, and the actual achieved code rates with the practical channel coding approach, indicating that there is certainly room for significant improvement with more sophisticated practical coding solutions.

4.4 Chapter Summary

In this chapter, we have considered channels with synchronization errors modeled by a bit deletion process with an additional segmentation assumption. We started with the argument that such channels are information stable, and their channel capacity exists. Then, we introduced several capacity upper and lower bounds in an attempt to understand the channel capacity behavior. The results indicate that when the deletion probability is near zero or near unity (for each segment), the upper and lower bounds behave similarly and we have obtained results very close to the capacity. However, there is a wide-range

of deletion probabilities where they are far apart, hence there is clearly more room for improvement (in terms of obtaining tighter capacity bounds). In addition to the information theoretic analysis of the channel, we have also considered a practical channel coding approach. Specifically, we used outer LDPC codes concatenated with inner marker codes, and developed suitable channel detection algorithms for this case. Different MAP based channel synchronization algorithms operating at the bit level and at the symbol level were introduced. Furthermore, we have compared complexity of the two algorithms and designed specific LDPC code for the segmented deletion channels which provide better decoding performance than the one optimized for the AWGN channels. Simulation results clearly showed the advantages of the proposed approach. In particular, for the entire range of deletion probabilities less than unity, the proposed approach offers a significantly larger transmission rate than the only other alternative solution of the zero-error codes designed in [17].

Chapter 5

DETECTION/DECODING OVER CHANNELS WITH SYNCHRONIZATION ERRORS AND INTER-SYMBOL INTERFERENCE

In this chapter, we consider channels with symbol insertions or deletions, along with ISI. We first start with our motivation of studying this channel model. Then, we design a MAP detection algorithm at the bit level based on a modification of the trellis diagram used in [18], which jointly achieves equalization for the ISI channel and synchronization for the insertion/deletion channel. Furthermore, as an alternative to the joint MAP detection, we introduce several low-complexity solutions. We utilize the M- and T- algorithms implemented as simplifications of the full complexity forward backward algorithm and also consider a separate channel detection scheme, i.e., the concatenation of an equalizer for the ISI channel and a MAP detector for synchronization purposes only. In addition to the two schemes, we also propose a detection algorithm based on the idea of sequential decoding. We show that these approaches greatly reduce the decoding complexity, especially for channels with long memory or for high insertion/deletion rates, at the expense of reduced decoding performance.

The rest of the chapter is organized as follows. In Sections 5.1 and 5.2, we present the research motivation and detailed introduction to the specific channel model. In Sections 5.3 and 5.4, we introduce the full-complexity bit-level MAP detection algorithm and investigate several sub-optimal, low-complexity detectors aimed at reducing the decoding complexity. Then, in Section 5.5, error-rate results for a practical LDPC-coded scheme are reported for these detectors. Finally, chapter summary is given in Section 5.6.

5.1 Motivation

We consider channels with symbol insertions or deletions and also suffering from ISI. Such channels appear in various practical transmission systems such as bit-patterned media (BPM) recording [1], which is a promising technology for future storage products due to its increased recording densities and much higher capacity compared to the conventional storage media. It typically consists of extremely small magnetically stable islands on which data bits are stored [1]. During the writing process, the write head moves from one island to another to magnetize them in certain directions based on the bits to be written. As the island positions show irregularities (jitter) and it is impossible to perfectly adjust the write head to every island, during the writing process some bits may not be correctly aligned (and written) and other bits may be inserted in error [74]. Since these errors caused by mis-synchronization are random and unpredictable, the BPM write channel can be modeled as one with insertions/deletions whose positions are unknown to the transmitter and/or the receiver. As for the reading process, multiple islands can be read at the same time, and therefore, the impairments of the readback channels include ISI, inter-track interference (ITI) and also AWGN [75, 76]. Combining the errors occurring in the write and read processes, a BPM recording system can be viewed as the cascade of an i.i.d. insertion/deletion channel and an ISI channel.

Synchronization errors along with ISI result in significant difficulties in the design and analysis of communication systems. To date, very few results have been reported on this type of a channel model. Since, even the computation of the capacity for insertion/deletion channels (with no ISI) is very

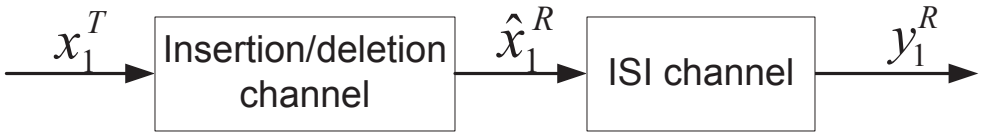

Figure 5.1: Channel model with deletions and ISI.

challenging [28–30], the capacity for insertion/deletion channels with ISI remains out of reach. Achievable rates are studied in [18] via a simulation-based approach, with a specifically designed trellis representation and independent and uniformly distributed (i.u.d.) inputs. A preliminary work in terms of practical codes over BPM channels is reported in [77], where the authors consider a BSC instead of an insertion/deletion channel concatenated with an ISI channel, and introduce several suitable channel detection algorithms along with channel codes. However, in the current literature, no channel coding results for insertion/deletion channels with ISI are available.

5.2 Channel Model

We consider transmission over channels impaired by insertion/deletion errors and ISI. We adopt the same coding scheme as in Chapters 3 and 4, with a modified MAP detection algorithm. The channel model is depicted in Fig. 5.1.

Assume that the binary phase-shift keying (BPSK) is used and let $\mathbf{x}_1^T = \{x_k\}_{k=1}^T$ and $\mathbf{y}_1^R = \{y_n\}_{n=1}^R$ be the sequences of symbols at the channel input and channel output, respectively. The number of transmitted symbols T is a fixed and given system parameter once the channel code is specified, while the number of received symbols R is a random variable depending on the realization of the insertion/deletion process. We adopt the channel model given

in [22] (except that substitution errors are not considered), i.e., each input symbol either gets deleted (with probability P_d), or replaced by two uniformly distributed random symbols (with probability P_i), or correctly transmitted (with probability $P_t = 1 - P_d - P_i$). We assume that all the synchronization errors are i.i.d., and neither the transmitter nor the receiver has information on the positions at which these errors occur. For the ISI channel, the output at time instant n, y_n, depends both on the inputs at n and the previous L symbols, which is given by

$$y_n = \sum_{l=0}^{L} h_l \hat{x}_{n-l} + z_n, \tag{5.1}$$

where $\hat{\mathbf{x}}_1^R$ is the resulting output of the insertion/deletion channel and L is the length of the ISI channel, i.e., $\{h_l\}_{l=0}^{L}$ are the coefficients of the $L+1$ channel taps which are normalized so that $\sum_{l=0}^{L} |h_l|^2 = 1$ and z_n is the AWGN with zero mean and variance σ^2, i.e., $z_n \sim \mathcal{N}(0, \sigma^2)$.

5.3 Bit Level MAP Detection Algorithm

As the channel model is a concatenation of an i.i.d. insertion/deletion channel and an ISI channel, an optimal channel MAP detector should be able to jointly achieve both equalization for the ISI channel and re-synchronization for the insertion/deletion channel. In the following, we propose such a joint bit-level MAP detection algorithm.

5.3.1 Trellis Diagram

Let $\mathbf{B}$ be the set of all possible vectors of length L with each element taking values ± 1, and $\mathbf{b}_n$ be state of the ISI channel at time n, i.e., $\mathbf{b}_n = [\hat{x}_n, \ldots, \hat{x}_{n-L+1}]^T$. Also define the binary event $D_{k,n,\mathbf{b}_n}$, with $k \in \{1, \ldots, T\}$, $n \in \{1, \ldots, R\}$, and $\mathbf{b}_n \in \mathbf{B}$, which denotes whether, after the first k transmit-

ted symbols, exactly n symbols are received and the state of the ISI channel is $\mathbf{b}_n$. Note that the received symbol y_n is corrupted by the ISI channel, and the first symbol of $\mathbf{b}_n$, $b_{n,0}$, equals x_k under a successful transmission. When a deletion error occurs, $\mathbf{b}_n = \mathbf{b}_{n-1}$, and for an insertion event, $b_{n,0}$ and $b_{n,1}$ are two random symbols taking values ± 1 with equal probability.

For further illustration, let us consider a simple ISI channel with $L = 2$ (i.e., a three tap channel) and unity channel taps. We can illustrate the trellis diagram as in Figure 5.2. With the input x_k, the transitions due to deletion, insertion and successful transmission result in 0, 2 and 1 output symbols, respectively. Note that a transmitted symbol cannot experience both insertion and deletion errors at the same time according to the channel model, and therefore, the state transitions with one output symbol can only happen due to a successful transmission (corrupted by ISI).

5.3.2 FBA for Insertion/Deletion Channel with ISI

To simplify the notation, we define $\mathbf{b}_n^{(i)} \triangleq [\hat{x}_{n-i}, \ldots, \hat{x}_{n-L+1}]^T$ and $\mathbf{b}_n^{(-i)} \triangleq [\hat{x}_n, \ldots, \hat{x}_{n-L+i+1}]^T$, $1 \leq i \leq L$, which are the rest of $\mathbf{b}_n$ except $b_{n,0}, \cdots, b_{n,i-1}$, and the rest of $\mathbf{b}_n$ except $b_{n,L-i}, \cdots, b_{n,L-1}$, respectively. Thus, we can express $\mathbf{b}_n$ as $\mathbf{b}_n = [b_{n,0}; \cdots; b_{n,i-1}; \mathbf{b}_n^{(i)}] = [\mathbf{b}_n^{(-i)}; b_{n,L-i}; \cdots; b_{n,L-1}]$.

Similar to the general FBA [9, 62], we first define the forward and backward quantities as

$$\alpha_k(n, \mathbf{b}_n) = P(\mathbf{y}_1^n, D_{k,n,\mathbf{b}_n}), \tag{5.2}$$

$$\beta_k(n, \mathbf{b}_n) = P(\mathbf{y}_{n+1}^R | D_{k,n,\mathbf{b}_n}), \tag{5.3}$$

which can be calculated recursively. As in [72], we obtain (for $L \geq 2$)

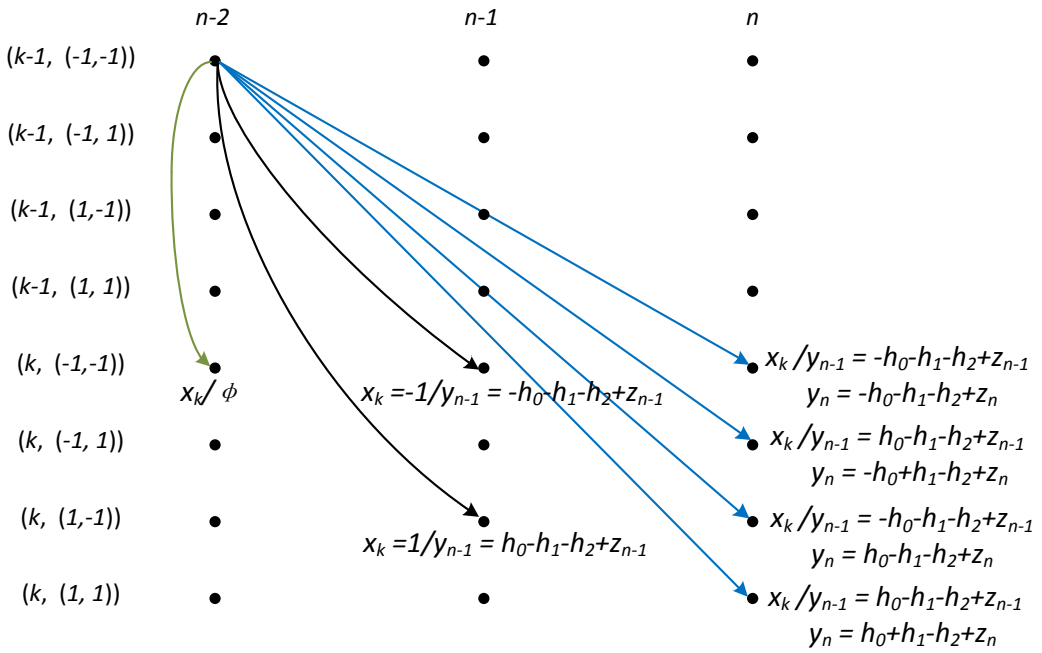

Figure 5.2: Example state transitions for insertion/deletion channel with ISI ($L = 2$).

$$\begin{aligned}\alpha_k(n, \mathbf{b}_n) =& P_t \sum_{j=\pm 1} \alpha_{k-1}(n-1, \mathbf{b}_{n-1} = [\mathbf{b}_n^{(1)}; j]) \cdot P(x_k = b_{n,0}) F_1(b_{n,0}, y_n, \mathbf{b}_{n-1}) \\ &+ \frac{P_i}{4} \sum_{j_1, j_2 = \pm 1} \alpha_{k-1}(n-2, \mathbf{b}_{n-2} = [\mathbf{b}_n^{(2)}; j_1; j_2]) \cdot F_2(y_n, y_{n-1}, \mathbf{b}_n, \mathbf{b}_{n-2}) \\ &+ P_d \, \alpha_{k-1}(n, \mathbf{b}_n) \ , \end{aligned} \tag{5.4}$$

$$\begin{aligned}\beta_k(n, \mathbf{b}_n) =& P_t \sum_{j=\pm 1} \beta_{k+1}(n+1, \mathbf{b}_{n+1} = [j; \mathbf{b}_n^{(-1)}]) \cdot P(x_{k+1} = j) F_1(j, y_{n+1}, \mathbf{b}_{n+1}) \\ &+ \frac{P_i}{4} \sum_{j_1, j_2 = \pm 1} \beta_{k+1}(n+2, \mathbf{b}_{n+2} = [j_1; j_2; \mathbf{b}_n^{(-2)}]) \\ &\qquad \cdot F_2(y_{n+2}, y_{n+1}, \mathbf{b}_{n+2}, \mathbf{b}_n) \\ &+ P_d \, \beta_{k+1}(n, \mathbf{b}_n) \ , \end{aligned} \tag{5.5}$$

where $P(x_k)$ is the a priori probability for transmitted sequence. If the detector has no a priori information for information bits, then $P(x_k)$ is 1/2 if x_k is an

LDPC-coded symbol, or 0 (or 1) if x_k belongs to the marker bits. From the ISI channel model, the two functions, $F_1(x, y, \mathbf{b})$ and $F_2(y, y', \mathbf{b}, \mathbf{b}')$, are expressed as

$$F_1(x, y, \mathbf{b}) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{ -\frac{|y - \mathbf{h}^T \cdot \mathbf{c}|^2}{2\sigma^2} \right\} , \tag{5.6}$$

$$F_2(y, y', \mathbf{b}, \mathbf{b}') = \frac{1}{2\pi\sigma^2} \exp\left\{ -\frac{|y - \mathbf{h}^T \mathbf{c}_1|^2 + |y' - \mathbf{h}^T \mathbf{c}_2|^2}{2\sigma^2} \right\} , \tag{5.7}$$

where $\mathbf{h} = [h_0, \cdots, h_L]^T$, $\mathbf{c} = [x; \mathbf{b}]$, $\mathbf{c}_1 = [\mathbf{b}; b'_{L-2}]$ and $\mathbf{c}_2 = [b_1; \mathbf{b}']$, respectively.

We are interested in the exact "frame synchronization" scenario, in which the values of T and R are known to the receiver. Therefore, the forward recursion can be initialized by setting (we assume that the last L symbols being transmitted before the current block are $\mathbf{b}^*$.)

$$\alpha_0(0, \mathbf{b}_0) = \begin{cases} 1, & \text{if } \mathbf{b}_0 = \mathbf{b}^*, \\ 0, & \text{else.} \end{cases} \tag{5.8}$$

Similarly, for the backward recursion, we have

$$\beta_T(R, \mathbf{b}_R) = \begin{cases} P(x_T = 1)/2^{L-1}, & \text{if } b_{R,0} = 1, \\ P(x_T = -1)/2^{L-1}, & \text{if } b_{R,0} = -1. \end{cases} \tag{5.9}$$

Having the forward and backward quantities, finally, the target conditional probability can be computed as

$$\begin{aligned} P(\mathbf{y}_1^R | x_k) =& P_t \sum_{n=0}^{\min(2k,R)} \sum_{\mathbf{b}_n \in \mathbf{B}} \alpha_{k-1}(n-1, \mathbf{b}_{n-1} = [\mathbf{b}_n^{(1)}; x_k]) \beta_k(n, \mathbf{b}_n) F_1(x_k, y_n, \mathbf{b}_{n-1}) \\ &+ \frac{P_i}{4} \sum_{n=0}^{\min(2k,R)} \sum_{\mathbf{b}_n \in \mathbf{B}} \sum_{j_1, j_2 = \pm 1} \beta_k(n, \mathbf{b}_n) \cdot F_2(y_n, y_{n-1}, \mathbf{b}_n, \mathbf{b}_{n-2}) \\ &\qquad \cdot \alpha_{k-1}(n-2, \mathbf{b}_{n-2} = [\mathbf{b}_n^{(2)}; j_1; j_2]) \\ &+ P_d \sum_{n=0}^{\min(2k,R)} \sum_{\mathbf{b}_n \in \mathbf{B}} \alpha_{k-1}(n, \mathbf{b}_n) \beta_k(n, \mathbf{b}_n) \ . \end{aligned} \tag{5.10}$$

We remark here that for the case of $L = 1$, slight changes of the algorithm are needed. The forward/backward recursions in this case are given as

$$\begin{aligned}\alpha_k(n, b_n) =& P_t \sum_{j=\pm 1} \alpha_{k-1}(n-1, b_{n-1} = j) \cdot P(x_k = b_n) \cdot F_1(x_k, y_n, b_{n-1}) \\ &+ \frac{P_i}{4} \sum_{j=\pm 1} \alpha_{k-1}(n-2, b_{n-2} = j) \cdot F_3(y_n, y_{n-1}, b_n, b_{n-2}) \\ &+ P_d\, \alpha_{k-1}(n, b_n)\ , \end{aligned} \tag{5.11}$$

$$\begin{aligned}\beta_k(n, b_n) =& P_t \sum_{j=\pm 1} \beta_{k+1}(n+1, b_{n+1} = j) \cdot P(x_{k+1} = j) \cdot F_1(x_{k+1}, y_{n+1}, b_{n+1}) \\ &+ \frac{P_i}{4} \sum_{j=\pm 1} \beta_{k+1}(n+2, b_{n+2} = j) \cdot F_3(y_{n+2}, y_{n+1}, b_{n+2}, b_n) \\ &+ P_d\, \beta_{k+1}(n, b_n)\ , \end{aligned} \tag{5.12}$$

where the function $F_3(y, y', b, b')$ is defined as

$$F_3(y, y', b, b') = \frac{1}{4\pi\sigma^2} \sum_{j=\pm 1} \exp\left\{ -\frac{|y - \mathbf{h}^T \mathbf{c}_j|^2 + |y' - \mathbf{h}^T \mathbf{c}'_j|^2}{2\sigma^2} \right\}\ , \tag{5.13}$$

with $\mathbf{c}_j = [b; j]$ and $\mathbf{c}'_j = [j; b']$. The final step is therefore given by

$$\begin{aligned}P(\mathbf{y}_1^R | x_k) =& P_t \sum_{n=0}^{\min(2k,R)} \sum_{j=\pm 1} \alpha_{k-1}(n-1, b_{n-1} = j) \cdot \beta_k(n, x_k) \cdot F_1(x_k, y_n, j) \\ &+ \frac{P_i}{4} \sum_{n=0}^{\min(2k,R)} \sum_{j=\pm 1} \beta_k(n, x_k) \cdot F_3(y_n, y_{n-1}, x_k, j) \cdot \alpha_{k-1}(n-2, j) \\ &+ P_d \sum_{n=0}^{\min(2k,R)} \sum_{j=\pm 1} \alpha_{k-1}(n, b_n = j) \beta_k(n, x_k)\ . \end{aligned} \tag{5.14}$$

5.4 Low Complexity Detection Algorithms

When an ISI channel with a long memory, i.e., large L, and/or an insertion/deletion channel with a high insertion/deletion rate, is considered, the

joint MAP detection algorithm in Section 5.3 can be highly time-consuming and it becomes the bottleneck. Hence, low-complexity detection solutions are essential. With this motivation, in the rest of this section, we consider several reduced-complexity detection approaches [72].

5.4.1 Separate Detection

We first consider separate detection instead of joint detection to alleviate the complexity problem. As for the separate detection, the FBA [77] or a soft-MMSE equalizer [78] is first executed to generate the soft information for $\hat{\mathbf{x}}_1^R$, i.e., $\xi_n = \log\left\{\frac{P(\hat{x}_n=1|\mathbf{y}_1^R)}{P(\hat{x}_n=-1|\mathbf{y}_1^R)}\right\}$ for $n \in \{1, 2, \ldots, R\}$. Then a modified bit-level MAP detector for channel synchronization can be applied. Assuming that the LLRs belong to observations through an insertion/deletion channel and a binary asymmetric substitution channel where the substitution probabilities P_s are time-varying across the trellis with $P_s = \frac{1}{e^{\xi_n}+1}$ for bit "1" and $P_s = \frac{1}{e^{-\xi_n}+1}$ for bit "0". This is clearly a suboptimal approach, however, as we will illustrate, it performs reasonably well.

Comparing the detection schemes, we see that for the joint detection, on average, a total number of $N \cdot 2^L$ states are considered in each time instance, while this number is reduced to $N + 2^L$ for the separate detection algorithm, where N is the average number of states with non-zero forward/backward quantities obtained in (3.4) and (3.5). It is clear that the difference becomes significant for large P_d and P_i values.

5.4.2 Reduced-Complexity FBA with M- and T-Algorithms

In addition to the separate detection algorithm, we also borrow ideas from reduced-complexity Viterbi algorithm implementations, i.e., the M-algorithm

[19] and T-algorithm [20], and design low-complexity MAP detectors for the insertion/deletion channels with ISI. Their main ideas are very similar, i.e., to keep only the best several paths as survivor paths to be extended to the next trellis interval, and thereby reducing the overall number of operations in execution of the algorithm.

As for the reduced-complexity FBA of joint MAP detection, the algorithms are described as follows. For the M-algorithm, in each trellis interval of forward/backward recursion, we first calculate the forward/backward quantities for all states and then sort them. Only the states with the M largest quantities are retained and the rest of the states are not considered when obtaining the forward/backward quantities for the next trellis interval. For the T-algorithm, instead of keeping the paths with the M largest metrics, we choose the survivor states according to a certain threshold. Suppose S_k is value of the largest forward/backward quantities at time k. Since all the operations are done in the log domain for computational safety, we only retain the states with forward/backward quantities larger than $\eta \cdot S_K$, with $1 \leq \eta \leq \infty$, where $\eta = \infty$ represents the case of full-complexity MAP detection algorithm. Clearly, the numbers M and η determine the tradeoff between the detection complexity and overall decoding performance.

5.4.3 Soft-Input Soft-Output Stack Decoding Algorithm

Sequential decoding algorithms [79], e.g., stack and Fano algorithms [58], are the first practical method to decode convolutional codes and they can also be used in solving our problem. Compared to the maximum-likelihood (ML) decoder, which operates on the code trellis, sequential decoding works on the code tree and does not explore all possible paths as opposed to the Viterbi

algorithm. It is a sub-optimal solution, however, its performance is good particular at high signal-to-noise ratios (SNRs). For instance, the soft-input soft-output (SISO) twin stack decoder developed in [21] offers similar performance compared to the MAP decoder for the decoding of recursive systematic convolutional codes. In [23], it is shown that the SISO stack equalizer offers near-optimum performance over the MIMO frequency selective fading channels.

In the following, we aim at building a low-complexity channel detector using the stack algorithm suitable for our current problem, which jointly achieves equalization for the ISI channel and synchronization for the insertion/deletion channel.

5.4.3.1 The Stack Algorithm

The objective of the stack algorithm is to find the "best" path throughout the tree by comparing the metrics associated with different paths (maybe of different depths). The decoding process starts by initializing a finite-size stack with the root of the code tree, e.g., normally the all-zero state where the encoding begins. During each time interval k, the decoder extends the top node from the stack. The metric associated with each extended partial path, which depends on the channel transition probability matrix and a-priori information, for a rate b/c convolutional code is given by [80] ,

$$\sum_{j=1}^{c} \log \left(\frac{P(y_k^j | v_k^j)}{P(y_k^j)} \right) + \sum_{i=1}^{b} \log \left(P(u_k^i) \right) \ , \tag{5.15}$$

where $\{u_k^i\}$, $i = 1, \ldots, b$; $\{v_k^j\}$, $\{y_k^j\}$, $j = 1, \ldots, c$ are the input/output bits of the encoder and the corresponding received symbols at time interval k, respectively. The decoding follows by extending the top node of the stack (a

sorting procedure is needed based on the metric associated with each path) until it reaches the leaf of the code tree.

In order to generate the soft-output LLRs for the transmitted symbols, the FBA is utilized [21]. Following the notation in [21], we define the branch transition probability as

$$\begin{aligned} \gamma(y_k, m, m') &= P(y_k, s_k = m | s_{k-1} = m') \\ &= P(y_k | v_k) \cdot P(s_k = m | s_{k-1} = m') \ . \end{aligned} \tag{5.16}$$

It is shown that either of the two metrics in (5.15) and (5.16) can be used to decode the convolutional code [21], and is stored in a Gamma matrix during the decoding process. The generation of the soft output is done by a post-processing module, which computes the forward/backward quantities and the final output using the γ metrics recorded in the Gamma matrix [21].

5.4.3.2 Detection Strategy for Insertion/Deletion Channels with ISI

Suppose for a given transmitted sequence and a given received sequence, we obtain the maximum value of $H_{k-1,n,\mathbf{b}_n}$ for a particular k and all n, $0 \leq n \leq 2k-2$, $\mathbf{b}_n \in \mathbf{B}$. Then, the maximum of $H_{k,n,\mathbf{b}_n}$ will be the largest of the the following quantities [22]:

$$\max(H_{k-1,n,\mathbf{b}_n}) + \gamma_{\text{del}}, \tag{5.17}$$

$$\max(H_{k-1,n-2,\mathbf{b}_{n-2}=[\mathbf{b}_n^{(2)};j_1;j_2]}) + \gamma_{\text{ins}}, \tag{5.18}$$

$$\max(H_{k-1,n-1,\mathbf{b}_{n-1}=[\mathbf{b}_n^{(1)};j]}) + \gamma_{\text{tra}}, \tag{5.19}$$

where $j, j_1, j_2 = \pm 1$ and the quantities

$$\gamma_{\text{del}} = \log(P_d), \tag{5.20}$$

$$\gamma_{\text{ins}} = \log(P_i/4) + F_2'(y_n, y_{n-1}, \mathbf{b}_n, \mathbf{b}_{n-2}), \tag{5.21}$$

$$\gamma_{\text{tra}} \quad = \quad \log(P_t) + \log(P(x_k = b_{n,0})) + F_1'(b_{n,0}, y_n, \mathbf{b}_{n-1}), \qquad (5.22)$$

are the modified metrics in (5.16) corresponding to the insertion, deletion and successful transmission. $F_1'(x, y, \mathbf{b})$ and $F_2'(y, y', \mathbf{b}, \mathbf{b}')$ are calculated by taking the logarithm of the right hand side functions in (5.6) and (5.7). Therefore, the soft-input soft-output stack algorithm for insertion/deletion channels with ISI can be generalized as follows:

step 1 Initialize the stack with the root of the tree, i.e., $k = 0$, $n = 0$, $\mathbf{b} = -\mathbf{1}$.

step 2 Extend the top state (node) from the stack.

step 3 Compute the gamma metrics for all possible transitions due to the insertion, deletion and successful transmission according to (5.20), (5.21) and (5.22). If the metric of a particular state transition is not recorded, update the Gamma matrix.

step 4 Update and re-order the stack in the order of metrics. If two paths merge at the same state, choose the one with a larger metric $H_{k,n,\mathbf{b}_n}$.

step 5 If the top node of the stack is a leaf of the tree, terminate the decoding process and start the post-processing block to computer the soft output, otherwise goes to Step 2.

5.5 Simulation Results

In this section, we consider a practical coding scheme with the detection algorithms described in Sections 5.3 and 5.4. We first consider the concatenation of an i.i.d. deletion channel and a dicode channel in Fig. 5.3. The dicode channel is the simplest ISI channel with $L = 1$, described by $y_n = \frac{1}{\sqrt{2}}\hat{x}_n - \frac{1}{\sqrt{2}}\hat{x}_{n-1} + z_n$. We adopt an outer LDPC code with length 3001 and rate 0.667 concatenated

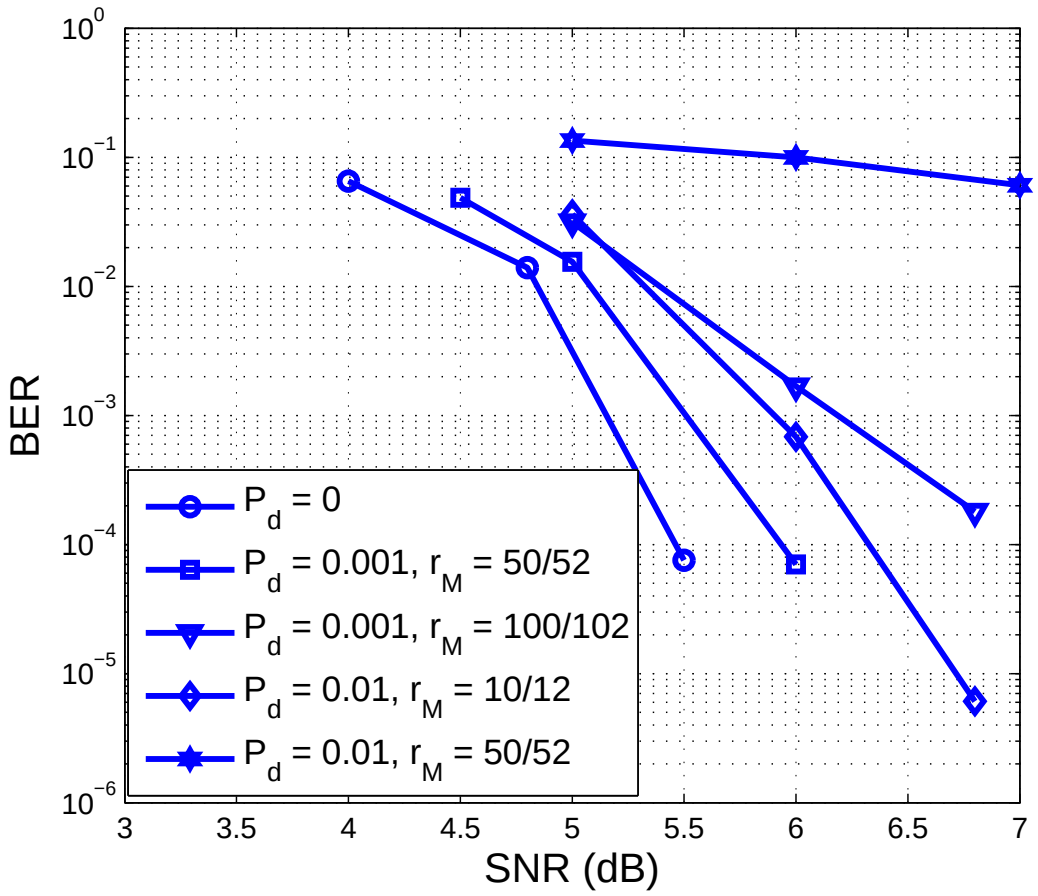

Figure 5.3: BER performance over concatenation of an i.i.d. deletion channel and a dicode channel.

with marker codes of varying rates. The marker codes are obtained by inserting the marker "01" every 10, 50 and 100 LDPC-coded bits, respectively. In the simulation, we implement the joint detection algorithm introduced in Section 5.3 and set the SNR to be $1/\sigma^2$. Fig. 5.3 compares the error rate performance for different values of P_d and r_M, and also the case of an ISI only channel ($P_d = 0$). It is clear that when P_d is small, we can achieve a good error rate performance with a small overhead introduced by the marker code, e.g., the gap between the cases when $P_d = 0$ and $P_d = 0.001$, $r_M = 50/52$

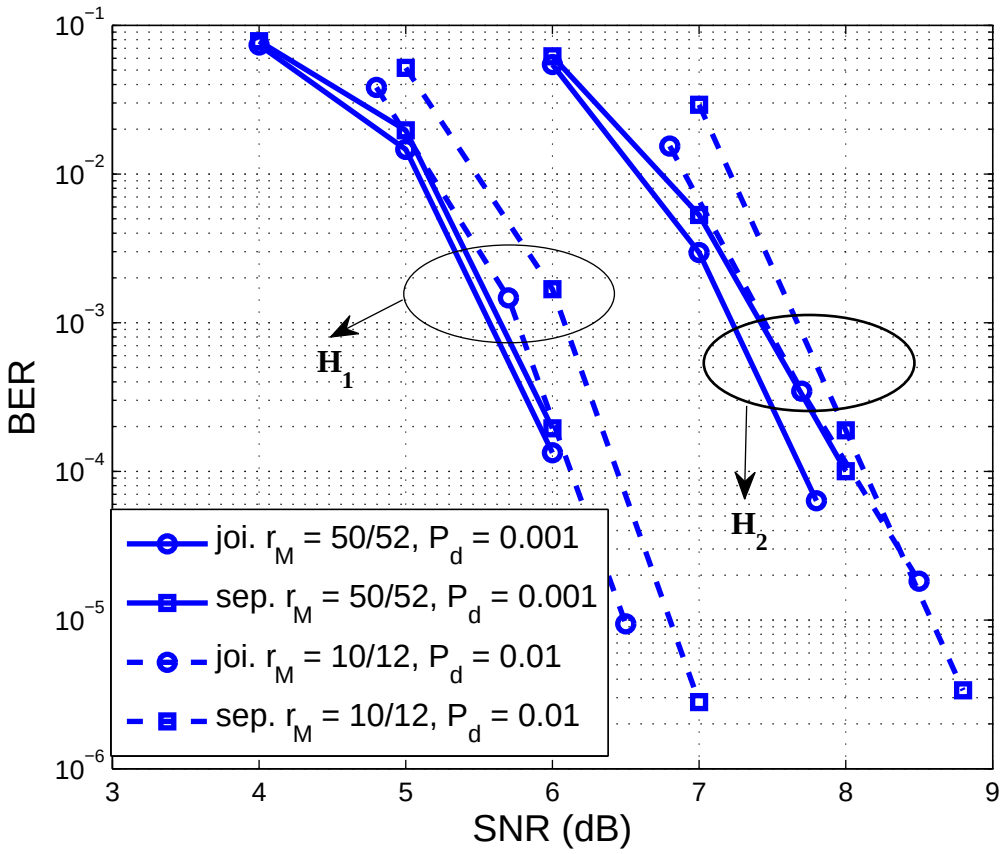

Figure 5.4: BER performance for the joint and separate MAP detectors.

is only about 0.3 dB. As the deletion probability increases, the performance with the same marker code degrades significantly.

In Fig. 5.4, we compare the resulting error rate performance of the joint and separate MAP detectors with the same LDPC code adopted in Fig. 5.3. Two ISI channels are considered with the channel tap coefficients $H_1 = [1\ 0\ 1]/\sqrt{2}$ and $H_2 = [1\ 1\ 1]/\sqrt{3}$. As expected, the complexity reduction comes at the price of some performance degradation and the difference becomes significant for large P_d values. The reason is that an i.i.d. deletion channel with a large deletion rate means more deletions per transmitted block,

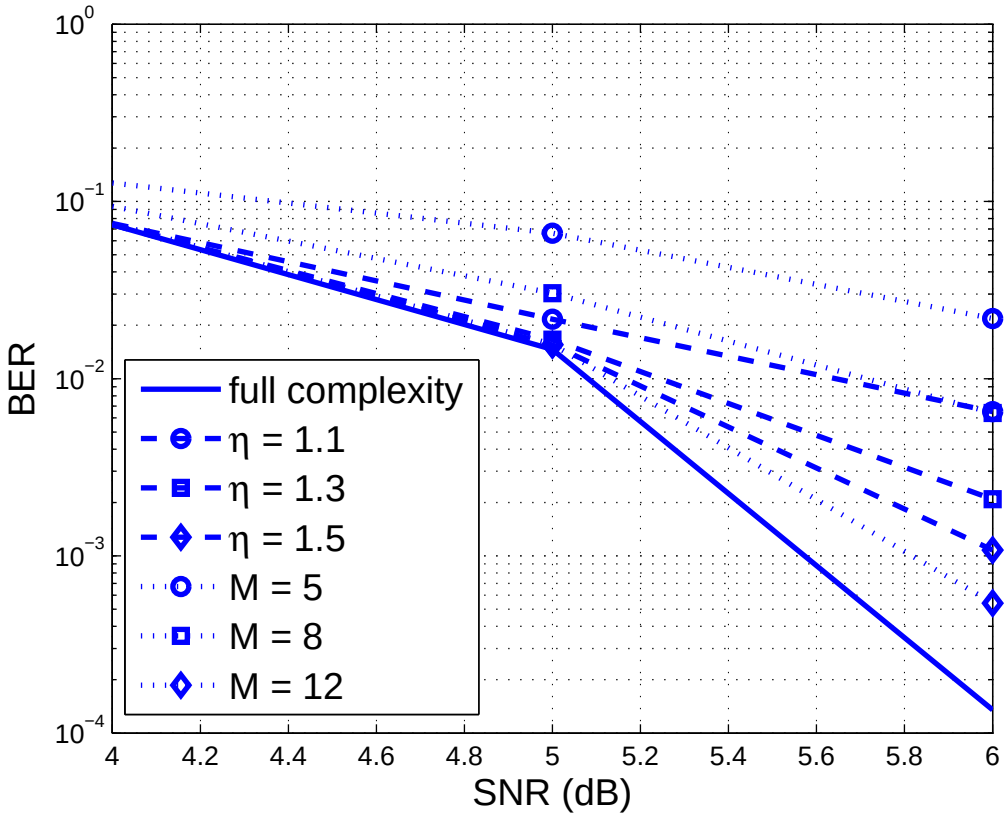

Figure 5.5: BER performance for the M- and T-algorithms.

and therefore, more states are introduced in the trellis diagram leading to a more obvious complexity reduction for the separate detection algorithm.

Fig. 5.5 illustrates the performance of the M- and T-algorithms with $M = 5$, 8, 12 and $\eta = 1.1$, 1.3, 1.5, respectively. We utilize the same LDPC code along with a rate $r_M = 50/52$ marker code and consider an i.i.d. deletion channel with $P_d = 0.001$ and an ISI channel with tap coefficients H_1. For comparison, the performance of the full-complexity joint MAP detection algorithm is also shown. For the M-algorithm, on average, we select 30.3%, 48.5% and 72.8% of all the states as survivors. For the T-algorithm, we eliminate around 92%, 83% and 67% of all the states when SNR is 6 dB. The performance

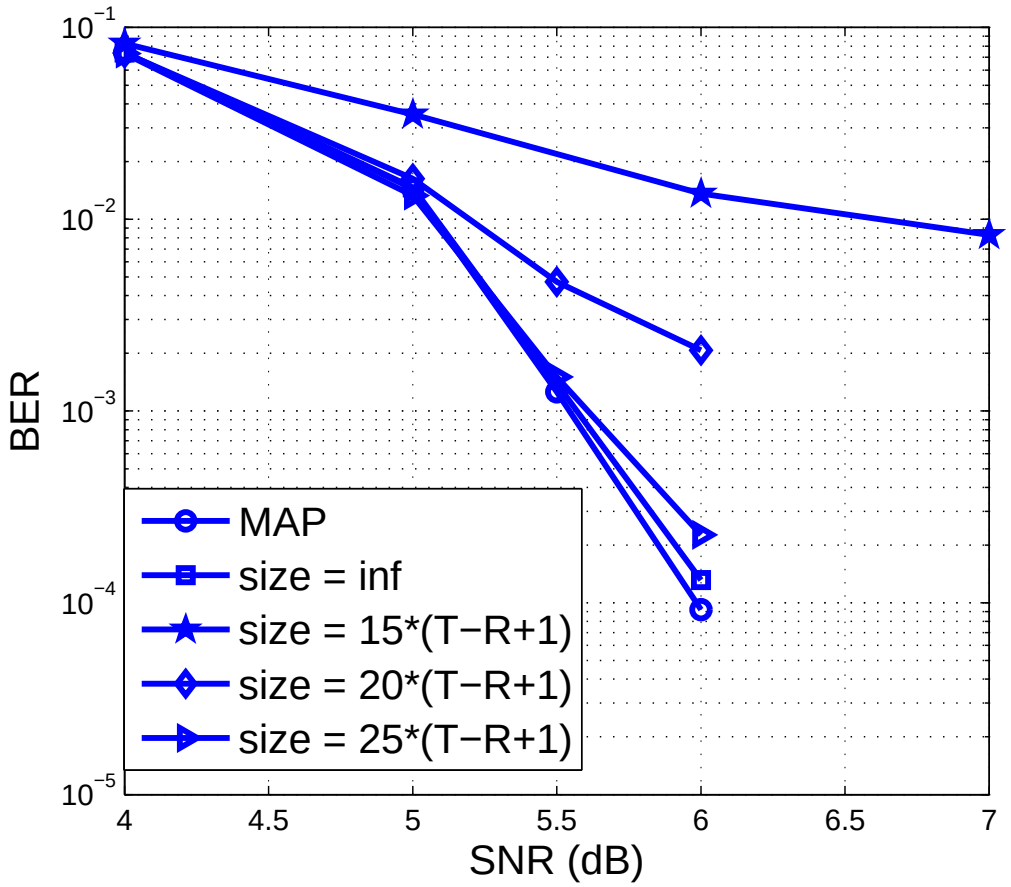

Figure 5.6: BER performance for the stack algorithms with different stack size.

improvement when more states are retained in the FBA is evident, as the percentage of the surviving states increases, the slopes of the BER curves become closer to the one obtained by the full complexity algorithm. Comparing the two algorithms, we notice that the T-algorithm has two advantages. First, it does not require the step of sorting as we only need to find the state with largest forward/backward quantity. Second, the number of states to be kept in each interval is varying which results in significant complexity savings at high SNRs, where the most possible state is much more "clearer" and there are few states exceeding the threshold.

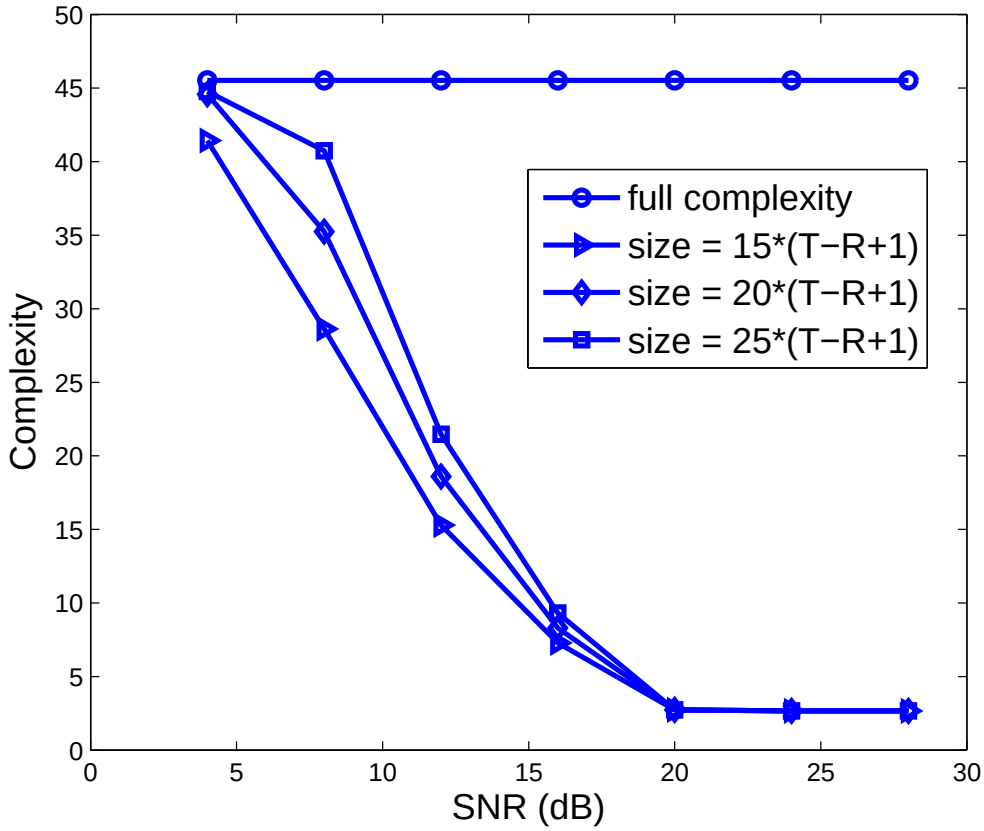

Figure 5.7: Complexity comparison for stack algorithm.

The performance comparison of the MAP detector, stack algorithm with infinite/finite stack size is provided in Fig. 5.6. The system parameters are the same as the ones in Fig 5.5. As expected, a larger stack size leads to a better performance. When an infinite-size stack is enforced, the stack algorithm offers almost the same performance of the MAP detector.

The improved performance of the stack algorithm with a larger stack size comes at a price of a more complicated system. The numerical results for complexity comparison between the stack algorithm-based detector and the MAP detector are shown in Fig. 5.7. We use the same definition of complexity

as in [23], which is the average number of extended branches, i.e., the number of metrics stored in the Gamma matrix, per transmitted symbol. Obviously, the complexity increases with a larger stack size and converges to the same level (around 2.66) at high SNRs. Contrary to the full-complexity MAP detector, at high SNRs, the complexity drops dramatically representing a considerable reduction in the number of computations.

5.6 Chapter Summary

In this chapter, we have considered detection/decoding algorithms for i.i.d. insertion/deletion channels with ISI. Through a suitable trellis diagram, we have developed a MAP detection algorithm operating at the bit level which provides both equalization for the ISI channel and synchronization for the insertion/deletion channel. Besides the joint MAP detection algorithm, we have also introduced several sub-optimal detectors aimed at reducing the detection complexity for channels with long memory. Three different approaches have been proposed, i.e., the separate detection algorithm, M- and T- algorithms, and the soft-input soft-output stack detection algorithm. Numerical examples are provided to illustrate the tradeoffs between the channel detection algorithm complexity and the error rate performance.

Chapter 6

MULTI-ANTENNA AND MULTI-USER COMMUNICATION SYSTEMS WITH DELETION ERRORS

In this chapter, instead of considering point-to-point systems with a single transmit and a single receive element, we take a step forward and study two cases: one utilizing multiple transmit/receiver elements and the other considering the case of multiple users communicating with a receiver through a multiple access channel.

We first consider an $N \times M$ MIMO channel with i.i.d. deletion errors. Specifically, during transmission, symbols may be deleted independently of each other (both spatially and temporally). This model is an extension of the i.i.d. deletion channel model for the single-input single-output communication scenario usually considered in the literature (e.g., see [28]). We further consider modulated symbols (using BPSK) and the effects of the electronic noise at the receiver(s). To communicate reliably over the MIMO deletion channel, we adopt a coding scheme, which is a serial concatenation of an LDPC code, a marker code [72] and a layered space-time code, and design two channel detectors. At the receiver, a bit-level MAP detector is deployed which jointly achieves synchronization for the deletion channel and detection for the MIMO channel. The resulting soft information is then fed to the message passing decoder (for the LDPC code). We also propose an alternative detector exploring the idea of interference cancellation (IC) which enables a layer-by-layer detection, and, therefore, existing synchronization algorithms (such as the one in [72]) can be directly utilized.

We then study a coding scheme for the case of multiple access channel in the presence of synchronization errors with a focus on the case of 2-user Gaussian multiple access channel with i.i.d. deletion errors. Again, generalization to the N-user Gaussian multiple access channel with insertion/deletion errors is possible. The system model can be viewed as a 2×1 (distributed) MIMO system, however, detection/decoding algorithms are different from the ones introduced for the MIMO deletion channel as shown in later sections.

The chapter is organized as follows. Our motivation and system model including the specific channel model and the coding scheme are described in Sections 6.1 and 6.2. In Section 6.3, two different channel detectors for the MIMO deletion channel are proposed. In Section 6.4, a channel detection algorithm along with an iterative decoding scheme is introduced for the 2-user Gaussian multiple access deletion channel. Simulation results are provided in Section 6.5, and concluding remarks are given in Section 6.6.

6.1 Motivation

The previous works on insertion/deletion channels in the literature focus exclusively on the case where there is a single transmitter and a single receiver. On the other hand there are many applications in which multiple-transmit and multiple-receive elements are employed [81]. Also multiple users may transmitting to the same receiver at the same time. The main objective of this chapter is to bring about a new channel modeling synchronization errors by also considering possible mismatches between different transmit and receive element pairs, and to consider a practical channel coding solution that can be employed for reliable communication.

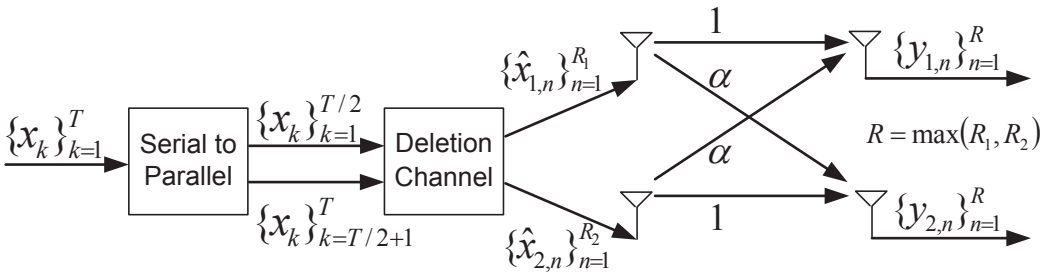

Figure 6.1: MIMO deletion channel model.

The MIMO deletion channel model we advocate is motivated by some practical digital communications applications, e.g., multi-track BPM recording systems [1, 60]. In this recording technology, the medium is prepatterned with magnetically stable (small) islands which suffer from imperfect synchronization. Furthermore, synchronization of the islands in the parallel tracks is not guaranteed. During the writing process, the signals are recorded in multiple tracks and synchronization errors occur due to possible imperfect alignment between the write head and the magnetic islands [74]. There are M heads reading N tracks simultaneously, yielding an $N \times M$ MIMO channel. Combining the errors occurring in the write and read processes, a multi-track BPM recording system can be viewed as the cascade of a deletion channel and an $N \times M$ MIMO channel. In a wireless sensor network, due to the power consumption constraints and other hardware limitations [59], perfect synchronization may not be feasible among different sensor nodes. When multiple sensors communicate simultaneously (over a multiple access channel) with one or more receivers, each transmitted sequence may experience deletion events independently from the other transmissions, and the resulting channel would be an asynchronous multiple access channel.

6.2 System Model

In this section, we describe the details of MIMO deletion channels and multiple access channels with deletion errors. Particularly, for the MIMO deletion channel, we focus on the case of $N = 2$ and $M = 2$ with the stipulation that extensions to arbitrary N, M are straightforward. The channel model, shown in Fig. 6.1, is the concatenation of a deletion channel with a 2×2 MIMO channel. The deletion channel is a standard i.i.d. deletion channel with symbol deletion probability P_d. For the MIMO channel, if the transmitted sequence is $\mathbf{x}$, the received signal (in the absence of noise) is given by $\mathbf{y} = \mathbf{A} \cdot \mathbf{x}$, where

$$A = \begin{bmatrix} 1 & \alpha \\ \alpha & 1 \end{bmatrix} \tag{6.1}$$

whose ij-th entry is the channel gain from the j-th transmitter to the i-th receiver. The matrix A is assumed to be deterministic and symmetric for simplicity. For instance, the coefficient α could represent the amount of ITI in a magnetic recording channel.

As for the proposed channel coding solution, we adopt a concatenated coding scheme similar to the one in [72], which consists of the interleaved serial concatenation of an outer LDPC code with an inner marker code and a layered space-time code. Specifically, the information bits are first encoded by an LDPC code, then marker bits are periodically inserted, e.g., we insert a two-bit marker "01" after every 10 LDPC-coded bits. Assume that the resulting bit sequence is of length T. This sequence is first modulated using BPSK, i.e., $\mathbf{x}_1^T = \{x_k\}_{k=1}^T$, and then converted into two parallel subsequences, each with length $T/2$. Due to the i.i.d. deletions, random symbols get deleted resulting in a total number of R_1 symbols in the first subsequence $\hat{\mathbf{x}}_1$ and R_2

symbols in the second subsequence $\hat{\mathbf{x}}_2$, where R_1 and R_2 are random variables (both binomial with parameters $T/2$ and P_d). The received signals at the two receive elements are

$$\begin{aligned} \mathbf{y}_1 &= \hat{\mathbf{x}}_1 + \alpha\hat{\mathbf{x}}_2 + \mathbf{z}_1 \ , \\ \mathbf{y}_2 &= \alpha\hat{\mathbf{x}}_1 + \hat{\mathbf{x}}_2 + \mathbf{z}_2 \ , \end{aligned} \tag{6.2}$$

where $\mathbf{z}_1$, $\mathbf{z}_2$ are independent white Gaussian noise sequences with zero mean and variance σ^2. Note that $\hat{\mathbf{x}}_1$ and $\hat{\mathbf{x}}_2$ may be of different lengths, i.e., $R_1 \neq R_2$, therefore, we define their vector sum as

$$\text{if } \ \mathbf{a}+\mathbf{b}=\mathbf{c} \ , \quad \text{then } \ c_k = \begin{cases} a_k + b_k & \text{if } k \leq \min(|\mathbf{a}|, |\mathbf{b}|) \\ a_k & \text{if } |\mathbf{b}| < k \leq |\mathbf{a}| \\ b_k & \text{if } |\mathbf{a}| < k \leq |\mathbf{b}| \end{cases} ,$$

where k and $|\cdot|$ represent the element index and the length of the vector, respectively. An alternative system model is also considered in this chapter (Section 6.3.2), where instead of choosing one LDPC code and a serial-to-parallel converter, we select two separate LDPC codes with the same length to be transmitted as the two parallel bit streams.

As for the 2-user Gaussian multiple access channel with deletions, the only difference from the previous case is that the information bits from the two users are encoded by separate LDPC codes of the same length. Assume that each sequence $\mathbf{x}$ is of length T, under the assumption of equal power allocation for the two users and block fading, the received signal is expressed as

$$\mathbf{y} = \alpha_1\hat{\mathbf{x}}_1 + \alpha_2\hat{\mathbf{x}}_2 + \mathbf{z}, \tag{6.3}$$

where α_1, α_2 are the channel coefficients, and $\mathbf{z}$ is the vector of independent

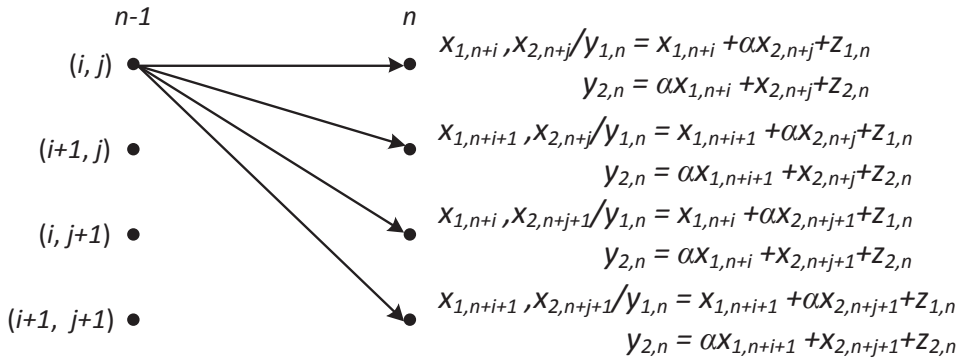

Figure 6.2: Example of state transitions on the trellis diagram.

Gaussian noise term with zero mean and a variance of σ^2. Clearly, a single element receiver is assumed.

6.3 Detection Algorithms for MIMO Channels with Deletions

In this section, we propose two solutions for the channel detection algorithm at the bit level, which generates soft information on the transmitted bits, i.e., the log-likelihood ratio $\log\left\{\frac{P(x_k=1|\mathbf{y}_1,\mathbf{y}_2)}{P(x_k=-1|\mathbf{y}_1,\mathbf{y}_2)}\right\}$ for $k \in \{1,\ldots,T\}$. The first one is the MAP detector, which jointly achieves synchronization and channel detection. The second one provides a low-complexity alternative, which utilizes the idea of interference cancellation in the decoding of the layers of the MIMO channel[1].

6.3.1 Joint MAP Detection Algorithm

The joint MAP detector generates soft information for all the LPDC-coded bits based on the received signals $\mathbf{y}_1 = \{y_{1,n}\}_{n=1}^{R}$ and $\mathbf{y}_2 = \{y_{2,n}\}_{n=1}^{R}$, where $R = \max(R_1, R_2)$. We define the state of the trellis to be $s_n = (d_{1,n}, d_{2,n})$, where $d_{1,n}$ denotes the total number of deletions for the first stream of bits at time n (i.e., by the time the n-th symbols in $\mathbf{y}_1$ and $\mathbf{y}_2$ are received) and $d_{2,n}$

[1]In this case, we assume separate LDPC codes for the two elements at the transmitter.

represents the total number of deletions for the second stream. An example of state transitions is shown in Fig. 6.2.

Defining $\mathbf{Y}_1^R = [\mathbf{y}_1; \mathbf{y}_2]$, we first express the forward and backward quantities as

$$\alpha_n(s_n) = P\big(s_n = (d_{1,n}, d_{2,n}), \mathbf{Y}_1^n\big), \tag{6.4}$$

$$\beta_n(s_n) = P\big(\mathbf{Y}_{n+1}^R | s_n = (d_{1,n}, d_{2,n})\big), \tag{6.5}$$

which can be calculated by means of the following recursion:

$$\begin{aligned}
\alpha_n(s_n) &= P\big(s_n = (d_{1,n}, d_{2,n}), \mathbf{Y}_1^n\big) \\
&= \sum_{s_{n-1}} P\big(s_{n-1} = (d_{1,n-1}, d_{2,n-1}), s_n, \mathbf{Y}_1^{n-1}, \mathbf{Y}_n\big) \\
&= \sum_{s_{n-1}} P\big(s_n, \mathbf{Y}_n | s_{n-1} = (d_{1,n-1}, d_{2,n-1})\big)\alpha_{n-1}(s_{n-1}),
\end{aligned} \tag{6.6}$$

$$\begin{aligned}
\beta_n(s_n) &= P\big(\mathbf{Y}_{n+1}^R | s_n = (d_{1,n}, d_{2,n})\big) \\
&= \sum_{s_{n+1}} P\big(\mathbf{Y}_{n+1}, \mathbf{Y}_{n+2}^R, s_{n+1} = (d_{1,n+1}, d_{2,n+1}) | s_n\big) \\
&= \sum_{s_{n+1}} P\big(s_{n+1} = (d_{1,n+1}, d_{2,n+1}), \mathbf{Y}_{n+1} | s_n\big)\beta_{n+1}(s_{n+1}),
\end{aligned} \tag{6.7}$$

where $\mathbf{Y}_n = [y_{1,n}, y_{2,n}]^T$ is the n-th column in $\mathbf{Y}$. By exploiting the "frame synchronization" assumption [72], the forward recursion can be initialized by setting

$$\alpha_0(s_0) = \begin{cases} 1, & \text{if } s_0 = (0,0), \\ 0, & \text{else.} \end{cases} \tag{6.8}$$

Similarly for the backward recursion, we have

$$\beta_R(s_R) = \begin{cases} 1, & \text{if } s_R = (T/2 - R_1, T/2 - R_2), \\ 0, & \text{else.} \end{cases} \tag{6.9}$$

Define $\gamma_n(s_{n-1}, s_n) = P\big(s_n = (d_{1,n}, d_{2,n}), \mathbf{Y}_n | s_{n-1} = (d_{1,n-1}, d_{2,n-1})\big)$. It is straightforward to show that

$$\gamma_n(s_{n-1}, s_n) = P(\mathbf{Y}_n | s_{n-1}, s_n) P(s_n | s_{n-1}), \tag{6.10}$$

where

$$P(s_n | s_{n-1}) = P_d^{d_{1,n}+d_{2,n}-d_{1,n-1}-d_{2,n-1}} (1 - P_d)^2, \tag{6.11}$$

if $d_{1,n} \geq d_{1,n-1}$, $d_{2,n} \geq d_{2,n-1}$, and 0 otherwise. Also

$$\begin{aligned} P\big(\mathbf{Y}_n | s_{n-1}, s_n\big) = & \sum_{i,j=\pm 1} P(\mathbf{Y}_n | x_{1,n+d_{1,n}} = i, x_{2,n+d_{2,n}} = j) \\ & \cdot P(x_{1,n+d_{1,n}} = i) P(x_{2,n+d_{2,n}} = j) \\ = & \frac{1}{2\pi\sigma^2} \sum_{i,j=\pm 1} \exp\left\{ -\frac{(y_{1,n} - (i + \alpha j))^2}{2\sigma^2} \right\} P_1(i) \\ & \cdot \exp\left\{ -\frac{(y_{2,n} - (\alpha i + j))^2}{2\sigma^2} \right\} P_2(j), \end{aligned} \tag{6.12}$$

where $P_1(i) = P(x_{1,n+d_{1,n}} = i)$ and $P_2(j) = P(x_{2,n+d_{2,n}} = j)$. They are set to 0 or 1 for the marker bits and 0.5 for the LDPC-coded bits.

Having the forward and backward quantities, it is easy to show the following equations:

$$\begin{aligned} P(\mathbf{Y}_1^R | x_k) &= \sum_{s_n} \sum_{s_{n-1}} P(s_n, s_{n-1}, \mathbf{Y}_1^{n-1}, \mathbf{Y}_n, \mathbf{Y}_{n+1}^R | x_k) \\ &= \sum_{s_n} \sum_{s_{n-1}} \alpha_{n-1}(s_{n-1}) P(s_n | s_{n-1}) P(\mathbf{Y}_n | s_{n-1}, s_n, x_k) \beta_n(s_n), \end{aligned} \tag{6.13}$$

where $n = k - d$, $d = d_{n,2}$ if $k > \frac{T}{2}$ and $d = d_{n,1}$ otherwise. Also, we have

$$P(\mathbf{Y}_n | s_{n-1}, s_n, x_k) = \begin{cases} \frac{1}{2\pi\sigma^2} \sum_{i=\pm 1} \exp\left\{ -\frac{(y_{1,n} - (i + \alpha x_k))^2}{2\sigma^2} \right\} \exp\left\{ -\frac{(y_{2,n} - (\alpha i + x_k))^2}{2\sigma^2} \right\} P_1(i), & \text{if } k > \frac{T}{2}, \\ \frac{1}{2\pi\sigma^2} \sum_{i=\pm 1} \exp\left\{ -\frac{(y_{1,n} - (x_k + \alpha i))^2}{2\sigma^2} \right\} \exp\left\{ -\frac{(y_{2,n} - (\alpha x_k + i))^2}{2\sigma^2} \right\} P_2(i), & \text{else.} \end{cases} \tag{6.14}$$

After obtaining $P(\mathbf{Y}_1^R|x_k)$, we can compute the LLR $\log\left\{\frac{P(x_k=1|\mathbf{Y}_1^R)}{P(x_k=-1|\mathbf{Y}_1^R)}\right\}$ (assuming equally likely LDPC-coded bits).

6.3.2 Separate Detection with Interference Cancellation

As simplifications of the joint detection algorithm, separate detection is possible among different receiver elements by utilizing two IC schemes, as described here.

6.3.2.1 IC with Non-Decoded Bits

Using the QR decomposition [81, 82], we can write the received sequences as

$$\mathbf{Y}_1^R = \begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \end{bmatrix} = \begin{bmatrix} 1 & \alpha \\ \alpha & 1 \end{bmatrix} \cdot \begin{bmatrix} \hat{\mathbf{x}}_1 \\ \hat{\mathbf{x}}_2 \end{bmatrix} + \begin{bmatrix} \mathbf{z}_1 \\ \mathbf{z}_2 \end{bmatrix} = \mathbf{QR} \begin{bmatrix} \hat{\mathbf{x}}_1 \\ \hat{\mathbf{x}}_2 \end{bmatrix} + \begin{bmatrix} \mathbf{z}_1 \\ \mathbf{z}_2 \end{bmatrix}, \quad (6.15)$$

where $\mathbf{R}$ is a 2×2 upper triangular matrix and $\mathbf{Q}$ is a 2×2 unitary matrix with $\mathbf{Q}^H\mathbf{Q} = \mathbf{I}_2$.

Let us left-multiply $\mathbf{Y}_1^R$ with $\mathbf{Q}^H$, which gives

$$\hat{\mathbf{Y}} = \mathbf{Q}^H\mathbf{Y} = \mathbf{R} \cdot \begin{bmatrix} \hat{\mathbf{x}}_1 \\ \hat{\mathbf{x}}_2 \end{bmatrix} + \hat{\mathbf{Z}}, \quad (6.16)$$

where $\hat{\mathbf{Z}} = \mathbf{Q}^H \begin{bmatrix} \mathbf{z}_1 \\ \mathbf{z}_2 \end{bmatrix}$. Since $\mathbf{R}$ is an upper triangular matrix, the second row in $\hat{\mathbf{Y}}$, $\hat{\mathbf{y}}_2$, solely consists of the transmitted sequence from the second layer. Therefore, the MAP detection algorithm in [72] can be directly applied to generate the soft information for the outer LDPC decoder[2].

For the first bit stream, estimates of $\hat{\mathbf{x}}_2$ (as hard decisions) are first obtained based on $\hat{\mathbf{y}}_2$. Then, they are multiplied by the off-diagonal coefficient

[2] A slight change in the algorithm is needed, since [72] only considers substitution-type errors instead of additive white Gaussian noise.

α and subtracted from $\mathbf{y}_1$. If all the decisions on $\hat{\mathbf{x}}_2$ are correctly made, the interference from the second stream is cancelled out and the same decoding procedure as in [72] can be performed.

6.3.2.2 IC with Decoded Bits

The interference cancellation scheme in the previous section does not utilize the error-correction capability offered by the outer LDPC decoder. In the following, we consider an IC scheme where the LDPC decoder is integrated into the process.

Let $\tilde{\mathbf{x}}_2$ be the final LDPC decoding output of the coded bits transmitted as the second stream, and $LLR(\mathbf{x}_2)$ be the vector of the corresponding log-likelihood ratios generated by the MAP detector. We group $\tilde{\mathbf{x}}_2$ into consecutive blocks, each consisting of all the LDPC-coded bits between two adjacent markers. The blocks of bits with successful transmissions, i.e., when no deletions occur, are marked "good", while the remaining blocks of bits are determined to be "contaminated" by deletions and are not used in the IC process. The positions of deletion errors can be found by identifying when the LLR values are close to zero, e.g., by obtaining the average of absolute values of the LLRs for each block and making a decision on the presence of deletion errors by comparing this value to a pre-determined threshold. Recall that without deletion errors, the mean of the absolute LLR value is $2/\sigma^2$, and therefore, we set the threshold to be $\eta \cdot 2/\sigma^2$, where $0 < \eta < 1$. The positions of these blocks in the received sequence are then estimated by [9]

$$\hat{k} = \arg\max_{k} \alpha_n(k)\beta_n(k), \tag{6.17}$$

where $\alpha_n(k)$ and $\beta_n(k)$ are the forward/backward quantities defined in the FBA in Chapter 3, n is the index of the last bit of the previous block in the

transmitted sequence and k is the index of the corresponding received bit[3]. Finally, the sequence to be subtracted from $\mathbf{y}_1$ is generated by substituting the bits in the hard decision of $\hat{\mathbf{x}}_2$ with the "good" bits from $\tilde{\mathbf{x}}_2$ (the starting positions are estimated using (6.17)).

When $\tilde{\mathbf{x}}_1$ is obtained, this information can also be utilized to cancel the interference for the second bit stream. The same procedure follows and iterative decoding between the first and the second data streams can be performed.

6.4 Detection/Decoding Algorithms for Two-User Gaussian Multiple Access Channel with Deletions

In this section, instead of considering systems with multiple-transmit and multiple-receive elements, we study the case where two transmitters communicate with the same receiver in the presence of deletion errors. We first introduce a joint MAP detection algorithm which achieves channel detection and synchronization simultaneously. Then, we describe an iterative decoding algorithm utilizing the output from the MAP detector, through which information from the two users can be decoded separately.

6.4.1 Joint MAP Detection Algorithm

The joint MAP detector generates soft information for all the possible LDPC-coded bit pairs based on the received signal $\mathbf{y}$. We define the state of the trellis to be $s_n = (d_{1,n}, d_{2,n})$, where $d_{1,n}$ denotes the total number of deletions from the first user by time n (the n-th symbol in $\mathbf{y}$) and $d_{2,n}$ represents the total number of deletions from the second user.

[3]The definitions are different from the ones in Section 6.3.1.

We assume that the total number of deletions for both users are available at the receiver end, i.e., $D_1 = T - R_1$ and $D_2 = T - R_2$ are known to the receiver. Note that, for a particular bit in $\mathbf{x}_1$, there are $D_1 + D_2 + 1$ bits in $\mathbf{x}_2$ that may be received at the same time, and vice versa. Therefore, the soft output of the FBA should generate the probabilities

$$P(\mathbf{y}|x_{1,k}, x_{2,k-D_1}), P(\mathbf{y}|x_{1,k}, x_{2,k-D_1+1}), \ldots, P(\mathbf{y}|x_{1,k}, x_{2,k}),$$
$$P(\mathbf{y}|x_{1,k}, x_{2,k+1}), \ldots, P(\mathbf{y}|x_{1,k}, x_{2,k+D_2-1}), P(\mathbf{y}|x_{1,k}, x_{2,k+D_2}),$$

for the transmitted bits from the first user, and

$$P(\mathbf{y}|x_{1,k-D_2}, x_{2,k}), P(\mathbf{y}|x_{1,k-D_2+1}, x_{2,k}), \ldots, P(\mathbf{y}|x_{1,k}, x_{2,k}),$$
$$P(\mathbf{y}|x_{1,k+1}, x_{2,k}), \ldots, P(\mathbf{y}|x_{1,k+D_1-1}, x_{2,k}), P(\mathbf{y}|x_{1,k+D_1}, x_{2,k}),$$

for the transmitted bits from the second user.

We first define the forward and backward quantities as

$$\alpha_n(s_n) = P\big(s_n = (d_{1,n}, d_{2,n}), \mathbf{y}_1^n\big), \tag{6.18}$$

$$\beta_n(s_n) = P\big(\mathbf{y}_{n+1}^R | s_n = (d_{1,n}, d_{2,n})\big), \tag{6.19}$$

which can be calculated by means of the following recursion

$$\begin{aligned}\alpha_n(s_n) &= P\big(s_n = (d_{1,n}, d_{2,n}), \mathbf{y}_1^n\big) \\ &= \sum_{s_{n-1}} P\big(s_n, y_n | s_{n-1}\big)\alpha_{n-1}(s_{n-1} = (d_{1,n-1}, d_{2,n-1})),\end{aligned} \tag{6.20}$$

and

$$\begin{aligned}\beta_{n-1}(s_{n-1}) &= P\big(\mathbf{y}_n^R | s_{n-1} = (d_{1,n-1}, d_{2,n-1})\big) \\ &= \sum_{s_n} P\big(s_n = (d_{1,n}, d_{2,n}), y_n | s_{n-1}\big)\beta_n(s_n),\end{aligned} \tag{6.21}$$

where y_n is the n-th symbol in $\mathbf{y}$. By exploiting the "frame synchronization" assumption, the forward recursion can be initialized by setting

$$\alpha_0(s_0) = \begin{cases} 1, & \text{if } s_0 = (0,0), \\ 0, & \text{else.} \end{cases} \tag{6.22}$$

Similarly for the backward recursion,

$$\beta_R(s_R) = \begin{cases} 1, & \text{if } s_R = (R_1, R_2), \\ 0, & \text{else.} \end{cases} \tag{6.23}$$

Define $\gamma_n(s_{n-1}, s_n) = P\big(s_n = (d_{1,n}, d_{2,n}), y_n | s_{n-1} = (d_{1,n-1}, d_{2,n-1})\big)$, $d_{1,n} \geq d_{1,n-1}$ and $d_{2,n} \geq d_{2,n-1}$. It is straightforward to show that

$$\gamma_n(s_{n-1}, s_n) = P(y_n | s_{n-1}, s_n) P(s_n | s_{n-1}), \tag{6.24}$$

where $P(s_n|s_{n-1})$ can be obtained from (6.11), and

$$\begin{aligned} P\big(y_n | s_{n-1}, s_n\big) &= \sum_{i,j=\pm 1} P(y_n | x_{1,n+d_{1,n}} = i, x_{2,n+d_{2,n}} = j) \\ &\qquad \cdot P(x_{1,n+d_{1,n}} = i) P(x_{2,n+d_{2,n}} = j) \\ &= \sum_{i,j=\pm 1} F(y_n, i, j) P(x_{1,n+d_{1,n}} = i) P(x_{2,n+d_{2,n}} = j), \end{aligned} \tag{6.25}$$

where the $F(\cdot)$ function is slightly different from the previous case, and it is given by

$$F(y, x_1, x_2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{ -\frac{(y - (\alpha_1 x_1 + \alpha_2 x_2))^2}{2\sigma^2} \right\}. \tag{6.26}$$

Having the forward and backward quantities, it is easy to show the following equalities

$$P(\mathbf{y}_1^R|x_{1,k}, x_{2,k+j_1}) = \sum_{s_{n-1}} \sum_{d_n=0}^{D_1} \alpha_{n-1}(s_{n-1}) P(s_n|s_{n-1}) F(y_n, x_{1,k}, x_{2,k+j_1})$$
$$\beta_n(s_n = (d_n, d_n + j_1)), \tag{6.27}$$

$$P(\mathbf{y}_1^R|x_{1,k+j_2}, x_{2,k}) = \sum_{s_{n-1}} \sum_{d_n=0}^{D_2} \alpha_{n-1}(s_{n-1}) P(s_n|s_{n-1}) F(y_n, x_{1,k+j_2}, x_{2,k})$$
$$\beta_n(s_n = (d_n + j_2, d_n)), \tag{6.28}$$

where $j_1 \in \{-D_1, \ldots, D_2\}$, $j_2 \in \{-D_2, \ldots, D_1\}$ and $n = k - d_n$.

6.4.2 Iterative Decoding with Outer LDPC Decoders

Having the channel detector output

$$\mathbf{L}(x_{1,k}) = \left[P(\mathbf{y}|x_{1,k}, x_{2,k-D_1}), \ldots, P(\mathbf{y}|x_{1,k}, x_{2,k}), \ldots, P(\mathbf{y}|x_{1,k}, x_{2,k+D_2})\right], \tag{6.29}$$

$$\mathbf{L}(x_{2,k}) = \left[P(\mathbf{y}|x_{1,k-D_2}, x_{2,k}), \ldots, P(\mathbf{y}|x_{1,k}, x_{2,k}), \ldots, P(\mathbf{y}|x_{1,k+D_1}, x_{2,k})\right], \tag{6.30}$$

$\forall k \in \{1, \ldots, T\}$, we are able to perform iterative decoding on a 2-user MAC factor graph [83]. Let $m_{vs}^{(1)}[k]$ and $m_{vs}^{(2)}[k]$ be the decoding outputs (log-likelihood ratios) for the k-th bit from the first and second LDPC decoders, respectively, and define

$$\mathbf{P}_{k,m}^{(1)} = \begin{bmatrix} P(\mathbf{y}|x_{1,k} = 0, x_{2,k+m} = 0) \\ P(\mathbf{y}|x_{1,k} = 0, x_{2,k+m} = 1) \\ P(\mathbf{y}|x_{1,k} = 1, x_{2,k+m} = 0) \\ P(\mathbf{y}|x_{1,k} = 1, x_{2,k+m} = 1) \end{bmatrix}, \tag{6.31}$$

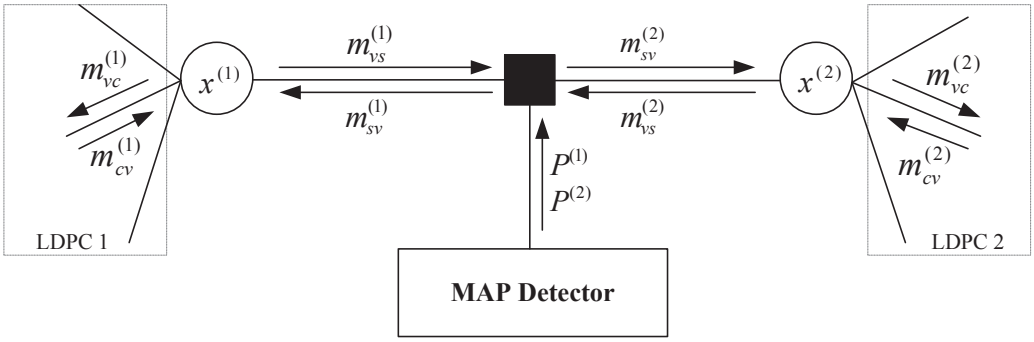

Figure 6.3: Factor graph for 2-user LDPC-MAC.

$$\mathbf{P}_{k,m}^{(2)} = \begin{bmatrix} P(\mathbf{y}|x_{1,k+m}=0, x_{2,k}=0) \\ P(\mathbf{y}|x_{1,k+m}=1, x_{2,k}=0) \\ P(\mathbf{y}|x_{1,k+m}=0, x_{2,k}=1) \\ P(\mathbf{y}|x_{1,k+m}=1, x_{2,k}=1) \end{bmatrix}. \tag{6.32}$$

The updated message to be passed to the i-th LDPC decoder, $m_{sv}^{(i)}[k]$, is calculated as

$$m_{sv}^{(1)}[k] = \sum_{m=-D_1}^{D_2} \log \frac{\mathbf{P}_{k,m}^{(1)}(4)\exp(m_{vs}^{(2)}[k+m]) + \mathbf{P}_{k,m}^{(1)}(3)}{\mathbf{P}_{k,m}^{(1)}(2)\exp(m_{vs}^{(2)}[k+m]) + \mathbf{P}_{k,m}^{(1)}(1)}, \tag{6.33}$$

$$m_{sv}^{(2)}[k] = \sum_{m=-D_2}^{D_1} \log \frac{\mathbf{P}_{k,m}^{(2)}(4)\exp(m_{vs}^{(1)}[k+m]) + \mathbf{P}_{k,m}^{(2)}(3)}{\mathbf{P}_{k,m}^{(2)}(2)\exp(m_{vs}^{(1)}[k+m]) + \mathbf{P}_{k,m}^{(2)}(1)}, \tag{6.34}$$

where $\mathbf{P}_{k,m}^{(1)}(i)$ and $\mathbf{P}_{k,m}^{(2)}(j)$ are the i-th and j-th elements in (6.31) and (6.32), respectively.

A detailed joint factor graph for the 2-user LDPC MAC in given in Fig. 6.3. The factor graphs of the two LDPC decoders are connected by the state check node (denoted as the black box in the figure), in which updated soft information $m_{sv}^{(1)}$ and $m_{sv}^{(2)}$ are generated based on (6.33) or (6.34). Depending on the order of the information exchange, serial or parallel scheduling for iterative decoding can be performed [83]. For serial scheduling, the decoding process starts with the LDPC decoder corresponding to the first user. With

no information from the second LDPC decoder ($m_{vs}^{(2)}[k] = 0$, $\forall k$), the soft information being passed to the first LDPC decoder is generated (from (6.33)) at the state check node. After performing the SPA, $m_{vs}^{(1)}$ is generated, and based on (6.34), the new decoding iterations for the second user start. In serial scheduling, a decoding round for one user is not initialized until the decoding procedure for the other user is completed. As for the case of parallel scheduling, the decoding iterations for two users are activated simultaneously.

6.5 Simulation Results

In this section, we first consider an example of the proposed coding scheme with the aim of demonstrating reliable communication through the MIMO deletion channel. In particular, we utilize an LDPC code of length 3001 and rate 0.667 and two marker codes with rates $r_M = 23/25$ and $48/50$, obtained by inserting a two-bit marker "01" every 23 or 48 LDPC-coded bits. Using the joint detection algorithm, in Fig. 6.4, we plot the error-rate performance for different P_d and α values, where the SNR is defined as $1/\sigma^2$. It is clear that when the deletion rate is high and the marker code rate is not sufficiently low, there exists an error floor, which indicates that deletion errors are not fully-corrected. When the α value is decreased, e.g., the ITI from the adjacent track is less severe, the performance improves dramatically. The last comment is that when $\alpha = 0$, the MIMO channel degrades to a conventional AWGN channel (as can be inferred from (6.2)), which offers the same result as for the case of the cascade of an i.i.d. deletion channel and an AWGN channel.

Fig. 6.5 compares the BER performance for different interference cancellation schemes over the MIMO deletion channel. We focus on the case of $P_d = 0.001$ and $r_M = 48/50$, and we set $\eta = 0.6$. We observe that there is

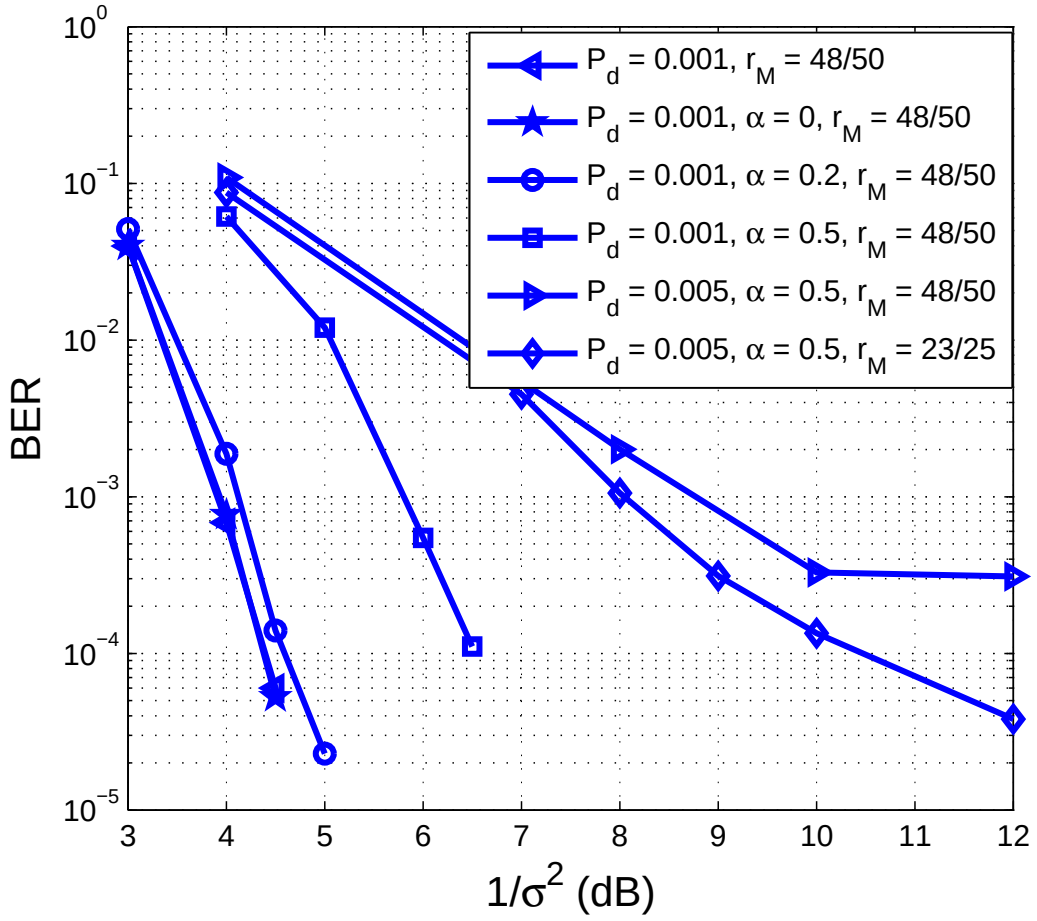

Figure 6.4: Error rate performance with various P_d, α and marker code rate.

a noticeable gap between the joint detection algorithm in Section 6.3.1 and the IC scheme in Section 6.3.2.1 (about 1dB for the first layer and 1.5dB for the second layer). We can narrow this gap by applying the scheme introduced in Section 6.3.2.2. It is also shown in the figure that, for this example, the performance improves with further iterations and the bit error rate finally converges to the joint detection result. However, we also observe an error floor, which indicates existence of persistent errors when determining the locations of deletions. This problem may be alleviated by using a lower rate marker code, at the expense of a larger overhead.

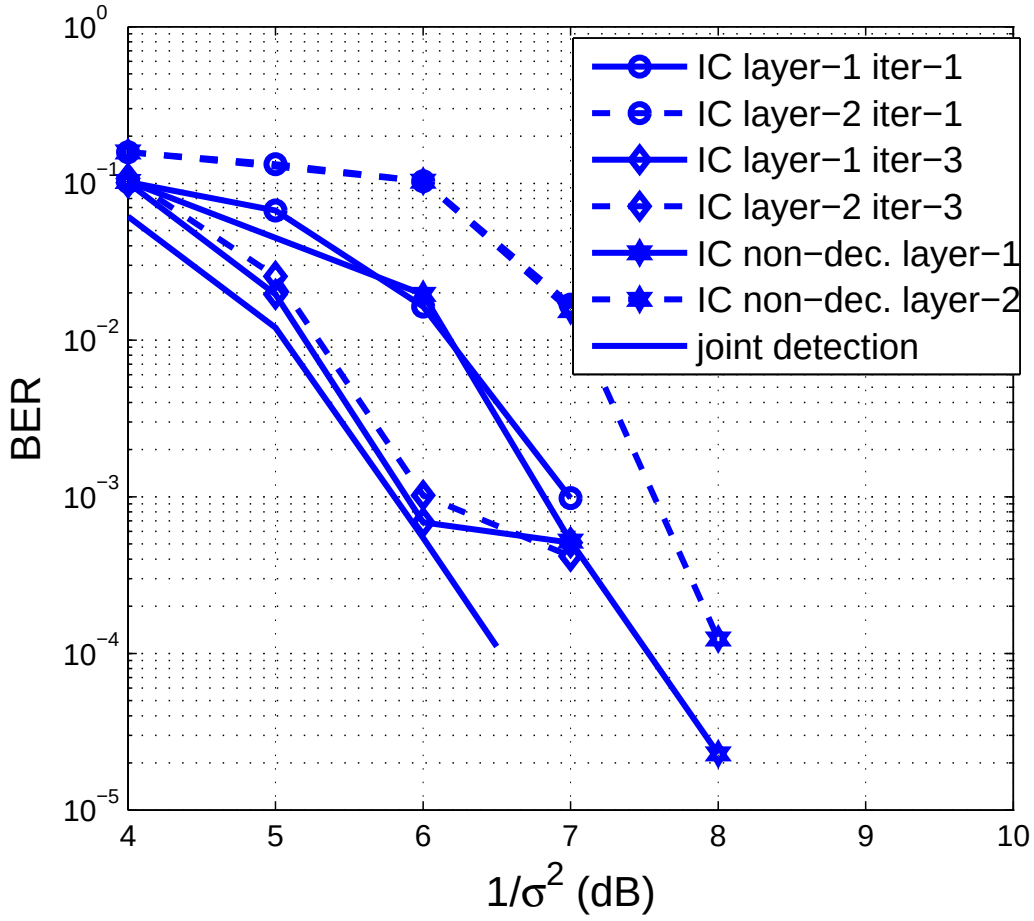

Figure 6.5: Error rate performance for different IC schemes.

For the case of the multiple access channel, we utilize an LDPC code of length 1920 and rate 0.333 and marker codes with rates $r_M = 20/22$, obtained by inserting the two-bit marker "01" every 20 LDPC-coded bits. Again, we set the SNR to be $1/\sigma^2$ and let $P_d = 0.001$. In Fig. 6.6, we plot the error-rate performance for different α values. Clearly, a more distinctive channel condition, i.e., a larger difference between two α values, leads to a better performance yet more obvious gap between the two users.

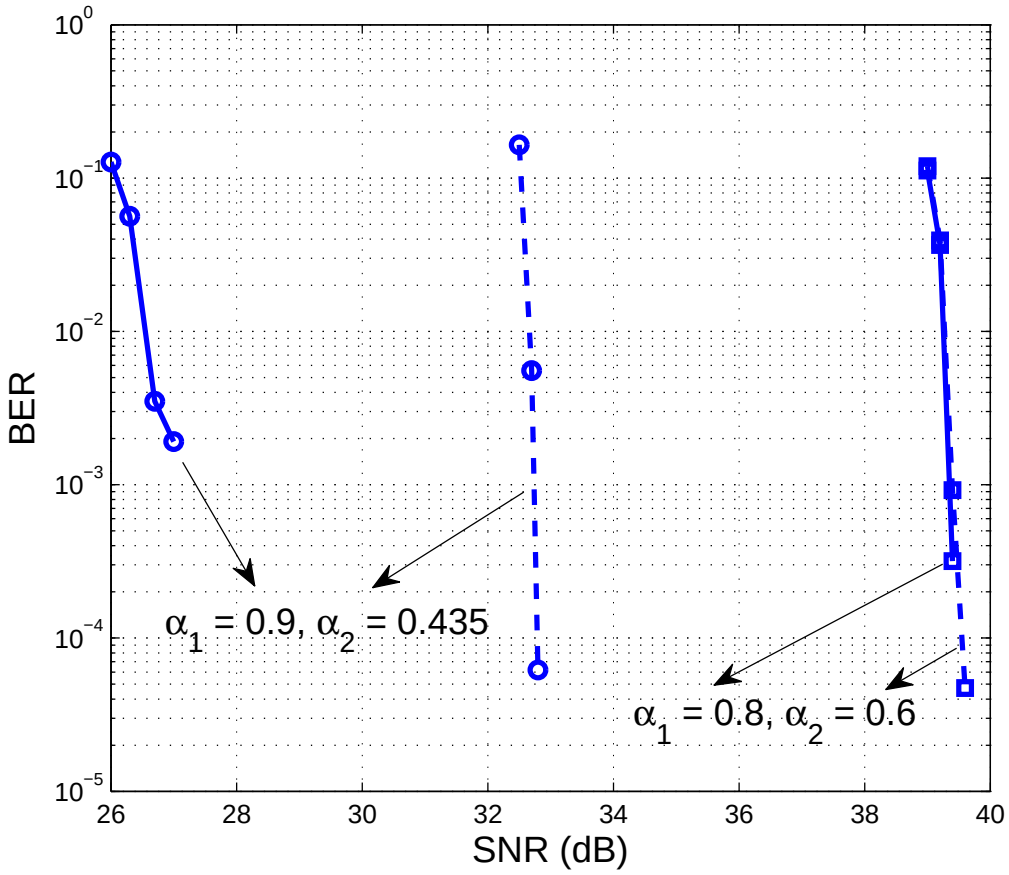

Figure 6.6: Error rate performance for different channel coefficients.

6.6 Chapter Summary

In this chapter, we have described channel models suitable for applications employing multiple transmit/receive elements and allowing multiple users with imperfect synchronization. For the 2×2 MIMO deletion channel, we have developed two detection algorithms operating at the bit level. The first detection algorithm is based on the MAP criterion while the second approach utilizes the IC scheme for the detection of layered space-time codes, which offers a low-complexity solution. We have also described a detection/decoding scheme for a 2-user Gaussian multiple access channel with i.i.d. deletion errors. By

generating a vector of soft information for each transmitted bit from each user and introducing an iterative decoding algorithm (between two LDPC decoders), we have shown that the information from two users can be successfully decoded simultaneously, as confirmed by the simulation results.

Chapter 7

SUMMARY AND CONCLUSIONS

Insertion and deletion errors represent some of the most important channel impairments and exist in many applications including high data rate wireless communications, bit-patterned media recoding systems, and so on. Many problems including finding tight upper and lower bounds on the channel capacity and developing good error correcting codes are extremely challenging even for some seemingly simple cases.

In this dissertation, we have considered practical coding schemes over i.i.d. insertion/deletion channels, i.e., using an outer LDPC code concatenated with an inner marker code. We first described the bit-level MAP decoding algorithm for Gallager's insertion/deletion channel model. For single-pass decoding, through numerical mutual information analyses, we developed a technique that allows us to optimize the marker code based on the ultimate rate achievable by the concatenated scheme. Moreover, we presented a new symbol-level detection algorithm, which has been proved to outperform the standard bit-level one in terms of achievable rates. We also considered a multi-pass decoding with information exchange between the inner detector and the outer decoder, and showed that by choosing good variable and check node degree distributions, LDPC codes designed for insertion and deletion channels offer better error correcting capabilities than those optimal for the AWGN-only channels. Simulation results related to practical LDPC codes showing clear performance gains were provided for both cases under consideration.

We have also studied segmented deletion channels introduced in [17]. In this line of work, we first gave an argument that such channels are information

stable, and their channel capacity exists. Then, we introduced several capacity upper and lower bounds in an attempt to understand the channel capacity behavior. The results indicate that when the deletion probability is near zero or near unity (for each segment), the upper and lower bounds behave similarly and the results very close to the capacity. However, there is a wide-range of deletion probabilities where they are far apart, hence there is room for further improvement in terms of obtaining tighter capacity bounds. In addition to the information theoretic analysis of the channel, we also considered a practical channel coding approach with suitable channel detection algorithms. Different MAP based channel synchronization algorithms operating at the bit level and at the symbol level were introduced, and their detection complexities were compared. Using the same method for the case of i.i.d. insertion/deletion channels, we designed specific LDPC codes for segmented deletion channels which provide better performance than the one optimized for the AWGN channels. Simulation results clearly show the advantages of the proposed approach. In particular, for the entire range of deletion probabilities less than unity, the proposed approach offers a significantly larger transmission rate than the only other alternative solution of the zero-error codes designed in [17].

As another contribution, we described the problem of coding over insertion/deletion channel with ISI. We designed a MAP detection algorithm at the bit level based on a modification of the trellis diagram used in [18], which jointly achieves equalization for the ISI channel and synchronization for the insertion/deletion channel. Then, as an alternative to the joint MAP detection, we introduced several low-complexity solutions. We utilized the stack, M- and T- algorithms implemented as simplifications of the full complexity forward backward algorithm and also considered a separate channel detection scheme,

i.e., concatenation of an equalizer for the ISI channel and a MAP detector for synchronization purposes. We showed that these approaches greatly reduce the decoding complexity, especially for channels with long memory or for high insertion/deletion rates, at the expense of reduced decoding performance.

Finally, we focused on the case where multiple antennas and multiple users are present. We first provided the system model and the motivation for studying these channels. We designed two detection algorithms for a 2×2 MIMO deletion channel, one is the full-complexity bit level MAP detector and the other explores suitable interference cancellation schemes. For a 2-user multiple access channel in the presence of deletion errors, we introduced a joint detection algorithm (with modifications of the MAP detector for the 2×2 MIMO deletion channel) and an iterative decoding strategy which enables information exchange between the two LDPC decoders. Through illustrative simulation results, we showed that reliable communications over these two channels are possible.

To summarize, in this dissertation, we have studied an important channel impairment, namely, presence of insertion and deletion errors in communication systems. We focused on finding an efficient (low detection/decoding complexity) yet good channel codes (with achievable rates close to the the capacity (bounds)) when synchronization errors are present. We demonstrated that by utilizing a marker code concatenated with an LDPC code, reliable communication is possible for (segmented) insertion/deletion channels with substitution errors, ISI, multiple antennas and multiple users.

Based on our findings, we can cite several possible future research directions. First of all, in all the results, we assume that perfect channel information is available at the receiver, i.e., P_i, P_d and P_s is accurately known.

It would be interesting to study the impact of imperfect knowledge of the insertion/deletion probabilities on the detection/decoding performance. Furthermore, in this work, we focus on the use of marker codes for synchronization, while on the other hand, watermark codes have been proposed as an alternative solution [9]. A better performance may be obtained if the ideas of watermark and marker codes can be combined together. Extensions of the work on segmented deletion channels are also possible, i.e., more general cases can be taken into account, such as the presence of insertion errors and multiple synchronization errors per segment instead of only one. Note that in this dissertation, LDPC code designs are only done for the i.i.d. insertion/deletion channel and the segmented deletion channel. Same method can be applied to insertion/deletion channels with ISI, MIMO deletion channels and multiple access channels with synchronization errors. Last but not the least, when additional constraints on the insertions/deletions are enforced, e.g., a minimum gap between adjacent errors is K bits/symbols, or a presence of feedback loop, e.g., the transmitter knows some knowledge about the insertion/deletion realization, a better coding/decoding schemes may be designed.

REFERENCES

[1] R. L. White, R. M. H. New, and R. F. W. Pease, "Patterned media: A viable route to 50 Gbit/in^2 and up for magnetic recording?" *IEEE Trans. on Magnetics*, vol. 33, no. 1, pp. 990–995, Jan. 1997.

[2] X. Zhang, M. Tao, and C. S. Ng, "Utility-based wireless resource allocation for variable rate transmission," *IEEE Trans. on Wireless Communications*, vol. 7, no. 9, pp. 3292–3296, Sep. 2008.

[3] P. A. H. Bours, "Constructing of fixed-length insertion/deletion correcting runlength-limited code," *IEEE Trans. on Information Theory*, vol. 40, pp. 1841–1856, Nov. 1994.

[4] R. M. Roth and P. H. Siegel, "Lee-metric BCH codes and their application to contrained and partial-response channels," *IEEE Trans. on Information Theory*, vol. 40, pp. 1083–1096, Jul. 1994.

[5] E. A. Ratzer, "Error-correction on non-standard communication channels," Ph.D. dissertation, University of Cambridge, Sep. 2003.

[6] ——, "Marker codes for channels with insertions and deletions," *Annals of Telecommunications*, vol. 60, pp. 29–44, Jan. 2003.

[7] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. New York: John Wiley & Sons, Inc., 1991.

[8] L. R. Bahl and F. Jelinek, "Decoding for channels with insertions, deletions, and substitutions with applications to speech recognition," *IEEE Trans. on Information Theory*, vol. 21, pp. 404–411, Jul. 1975.

[9] M. C. Davey and D. J. Mackay, "Reliable communication over channels with insertions, deletions and substitutions," *IEEE Trans. on Information Theory*, vol. 47, no. 2, pp. 687–698, Feb. 2001.

[10] S. ten Brink, G. Kramer, and A. Ashikhmin, "Design of low-density parity-check codes for modulation and detection," *IEEE Trans. on Communications*, vol. 52, no. 4, pp. 670–678, Apr. 2004.

[11] M. Franceschini, G. Ferrari, and R. Raheli, "Does the performance of LDPC codes depend on the channel?" *IEEE Trans. on Communications*, vol. 54, no. 12, pp. 2129–2132, Dec. 2006.

[12] N. Varnica and A. Kavćić, "Optimized low-density parity-check codes for partial response channels," *IEEE Commun. Letters*, vol. 7, no. 4, pp. 168–170, Apr. 2003.

[13] M. Franceschini, G. Ferrari, and R. Raheli, "EXIT chart-based design of LDPC codes for inter-symbol interference channels," in *Proc. IST Summit 2005*, Dresden, Germany, Jun. 2005.

[14] H. Saeedi and A. H. Banihashemi, "Design of irregular LDPC codes for BIAWGN channels with SNR mismatch," *IEEE Trans. on Communications*, vol. 57, no. 1, pp. 6–11, Jan. 2009.

[15] R. L. Dobrushin, "Shannon's theorems for channels with synchronization errors," *Problems of Information Transmission*, vol. 3, no. 4, pp. 11–26, 1967.

[16] Y. Kanoria and A. Montanari, "On the deletion channel with small deletion probability," in *Proc. IEEE International Symposium on Information Theory*, Austin, USA, Jun. 2010, pp. 1002–1006.

[17] Z. Liu and M. Mitzenmacher, "Codes for deletions and insertion channels with segmented errors," *IEEE Trans. on Information Theory*, vol. 56, no. 1, pp. 224–232, Jan. 2010.

[18] J. Hu, T. M. Duman, M. F. Erden, and A. Kavćić, "Achievable information rates for channels with insertions, deletions and intersymbol interference with i.i.d. inputs," *IEEE Trans. on Communications*, vol. 58, no. 4, pp. 1102–1111, Apr. 2010.

[19] J. B. Anderson, "Limited search trellis decoding of convolutional codes," *IEEE Trans. on Information Theory*, vol. 35, no. 5, pp. 944–955, Sep. 1989.

[20] S. J. Simmons, "Breadth-first trellis decoding with adaptive effort," *IEEE Trans. on Communications*, vol. 38, no. 1, pp. 3–12, Jan. 1990.

[21] R. Sivasankaran and S. W. McLaughlin, "Twin-stack decoding of recursive systematic convolutional codes," *IEEE Trans. on Communications*, vol. 49, no. 7, pp. 1158–1167, Jul. 2001.

[22] R. G. Gallager, "Sequential decoding for binary channels with noise and synchronization errors," MIT Lincoln Lab., Tech. Rep., Oct. 1961.

[23] T. Gucluoglu and T. M. Duman, "Soft input soft output stack equalization for MIMO frequency selective fading channels," in *Proc. IEEE International Conference on Communications*, Seoul, S. Korea, May 2005, pp. 510–514.

[24] X. Li, M. Chen, and W. Liu, "Cooperative transmissions in wireless sensor networks with imperfect synchronization," in *Proc. Thirty-Eighth Asilomar Conference on Signals, Systems and Computers*, Binghamton, NY, USA, Nov. 2004, pp. 1281–1285.

[25] K. S. Zigangirov, "Sequential decoding for a binary channel with dropouts and insertions," *Problems of Information Transmission*, vol. 5, no. 2, pp. 17–22, 1969.

[26] M. Mitzenmacher, "Capacity bounds for sticky channels," *IEEE Trans. on Information Theory*, vol. 54, no. 1, pp. 72–77, Jan. 2008.

[27] L. Dolecek and V. Anantharam, "On communication over channels with varying sampling rate," in *2007 Information Theory and Applications (ITA) Workshops (at UCSD)*, La Jolla, CA, USA, Jan.–Feb. 2007.

[28] M. Mitzenmacher, "A survey of results for deletion channels and related synchronization channels," *Probability Surveys*, pp. 1–33, Jun. 2009.

[29] D. Fertonani and T. M. Duman, "Novel bounds on the capacity of the binary deletion channel," *IEEE Trans. on Information Theory*, vol. 56, no. 6, pp. 2753–2765, Jun. 2010.

[30] D. Fertonani, T. M. Duman, and M. F. Erden, "Bounds on the capacity of channels with insertions, deletions and substitutions," *IEEE Trans. on Communications*, vol. 59, no. 1, pp. 2–6, Jan. 2011.

[31] J. D. Ullman, "On the capabilities of codes to correct synchronization errors," *IEEE Trans. on Information Theory*, no. 1, pp. 95–105, Jan. 1967.

[32] S. Diggavi and M. Grossglauser, "On information transmission over a finite buffer channel," *IEEE Trans. on Information Theory*, vol. 52, no. 3, pp. 1226–1237, Mar. 2006.

[33] E. Drinea and M. Mitzenmacher, "On lower bounds for the capacity of deletion channels," *IEEE Trans. on Information Theory*, vol. 52, no. 10, pp. 4648–4657, Oct. 2006.

[34] ——, "Improved lower bounds for the capacity of i.i.d. deletion and duplication channels," *IEEE Trans. on Information Theory*, vol. 53, no. 8, pp. 2693–2714, Aug. 2007.

[35] R. E. Blahut, "Computation of channel capacity and rate distortion functions," *IEEE Trans. on Information Theory*, vol. 18, pp. 460–473, Jan. 1972.

[36] S. Arimoto, "An algorithm for calculating the capacity of an arbitrary discrete memoryless channel," *IEEE Trans. on Information Theory*, vol. 18, pp. 14–20, Jan. 1972.

[37] A. Kalai, M. Mitzenmacher, and M. Sudan, "Tight asymptotic bounds for the deletion channel with small deletion probabilities," in *Proc. IEEE International Symposium on Information Theory*, Austin, USA, Jun. 2010, pp. 997–1001.

[38] Y. Kanoria and A. Montanari, "Optimal coding for the deletion channel with small deletion probability," arXiv:1104.5546v1., Tech. Rep., Apr. 2011.

[39] A. R. Iyengar, P. H. Siegel, and J. K. Wolf, "Modeling and information rates for synchronization error channels," in *Proc. IEEE International Symposium on Information Theory*, St. Petersburg, Russi, Jul.–Aug. 2011, pp. 380–384.

[40] F. F. Sellers Jr., "Bit loss and gain correction code," *IRE Trans. on Information Theory*, vol. 8, no. 1, pp. 35–38, Jan. 1962.

[41] N. J. A. Sloane, "On single-deletion-correcting codes," in *Codes and Designs: Proc. Conf. Honoring Professor Dijen K. Ray-Chaudhuri on the Occasion of His 65th Birthday, Ohio State University*, May 2000, pp. 273–291.

[42] A. S. Helberg and H. C. Ferreira, "On multiple insertion/deletion correcting codes," *IEEE Trans. on Information Theory*, vol. 48, no. 1, pp. 305–308, Jan. 2002.

[43] L. Dolecek and V. Anantharam, "Using Reed-Muller RM $(1, m)$ codes over channels with synchronization and substitution error," *IEEE Trans. on Information Theory*, vol. 53, no. 4, pp. 1430–1443, Apr. 2007.

[44] L. McAven and R. Safavi-Naini, "Classification of the deletion correcting capabilities of Reed-Solomon codes of dimension 2 over prime fields," *IEEE Trans. on Information Theory*, vol. 53, no. 6, pp. 2280–2294, Jun. 2007.

[45] S. E. Tavares and M. Fukada, "Matrix approach to synchronization recovery for binary cyclic codes," *IEEE Trans. on Information Theory*, vol. 15, no. 1, pp. 93–101, Jan. 1969.

[46] G. Seguin, "On synchronizable binary cyclic codes," *IEEE Trans. on Information Theory*, vol. 21, no. 5, pp. 589–592, Sep. 1975.

[47] L. Cheng and H. Ferreira, "Rate-compatible path-pruned convolutional codes and their applications on channels with insertion, deletion and substitution errors," in *Proc. IEEE Information Theory Workshop 2005*, Rotorua, New Zealand, Aug. 2005, pp. 20–25.

[48] T. G. Swart and H. C. Ferreira, "Insertion /deletion correcting coding schemes based on convolution coding," *Electronics Letters*, vol. 38, no. 16, pp. 871–873, Aug. 2002.

[49] M. F. Mansour and A. H. Tewfik, "Convolutional codes for channels with substitutions, insertions, and deletions," in *Proc. IEEE Global Telecommun. Conf.*, vol. 2, Nov. 2002, pp. 1051–1055.

[50] V. I. Levenshtein, "Binary codes capable of correcting deletions, insertions and reversals," *Dokl. Akad. Nauk SSSR*, vol. 163, no. 4, pp. 845–848, 1965.

[51] H. C. Ferreira, T. G. Swart, and M. P. dos Santos, "Using parallel-interconnected Viterbi decoders to correct insertion/deletion errors," in *IEEE 7th AFRICON Conference*, vol. 1, 2004, pp. 341–344.

[52] L. Schulman and D. Zuckerman, "Asymptotically good codes correcting insertions, deletions, and transpositions," *IEEE Trans. on Information Theory*, vol. 45, no. 7, pp. 2552–2557, Nov. 1999.

[53] J. Chen, M. Mitzenmacher, C. Ng, and N. Varnica, "Concatenated codes for deletion channels," in *Proc. IEEE International Symposium on Information Theory*, Yokohama, Japan, 2003, p. 218.

[54] T. G. Swart and H. C. Ferreira, "Insertion/deletion correcting coding schemes based on convolution coding," *IEEE Electronics Letters*, vol. 38, no. 16, pp. 871–873, Aug. 2002.

[55] C. E. Shannon, "A mathematical theory of communication," *Bell System Technical Joural*, vol. 27, pp. 379–423, Jul. 1948.

[56] R. Gallager, "Low-density parity-check codes," *IRE Trans. on Information Theory*, vol. 8, no. 1, pp. 21–28, Jan. 1962.

[57] D. J. Mackay, "Good error correcting codes based on very sparse matrices," *IEEE Trans. on Information Theory*, vol. 45, pp. 399–431, Mar. 1999.

[58] L. Shu and D. J. C. Jr., *Error Control Coding, second edition.* Upper Saddle River, NJ: Prentice Hall, Inc., 2004.

[59] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks*, vol. 38, pp. 393–422, 2002.

[60] H. J. Richter, A. Y. Dobin, O. Heinonen, K. Z. Gao, R. J. M. Veerdonk, R. T. Lynch, J. Xue, D. Welle, P. Asselin, M. F. Erden, and R. M. Brocki, "Recording on bit-patterned media at densities of 1 Tb/in^2 and beyond," *IEEE Trans. on Magnetics*, vol. 42, no. 10, pp. 2255–2260, Oct. 2006.

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

[62] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. on Information Theory*, vol. 47, pp. 498–519, Feb. 2001.

[63] A. Kavćić, X. Mao, and M. Mitzenmacher, "Binary intersymbol interference channels: Gallager codes, density evolution, and code performance bounds," *IEEE Trans. on Information Theory*, vol. 40, no. 7, pp. 1636–1652, Jul. 2003.

[64] D. Arnold, H. Loeliger, P. Vontobel, A. Kavćić, and W. Zeng, "Simulation-based computation of information rates for channels with memory," *IEEE Trans. on Information Theory*, vol. 52, no. 8, pp. 3498–3508, Aug. 2006.

[65] E. Drinea and A. Kirsch, "Directly lower bounding the information capacity for channels with i.i.d. deletions and duplications," *IEEE Trans. on Information Theory*, vol. 56, no. 1, pp. 86–102, Jan. 2010.

[66] X. Li and J. A. Ritcey, "Bit-interleaved coded modulation with iterative decoding," *IEEE Commun. Letters*, vol. 1, no. 6, pp. 169–171, Nov. 1997.

[67] T. T. Kadota, "On the information stability of stationary ergodic processes," *SIAM Journal on Applied Mathematics*, vol. 26, no. 1, pp. 176–182, Jan. 1974.

[68] S. Verdu and T. Weissman, "The information lost in erasures," *IEEE Trans. on Information Theory*, vol. 54, no. 11, pp. 5030–5058, Nov. 2008.

[69] S. Arimoto, "An algorithm for calculating the capacity of an arbitrary discrete memoryless channel," *IEEE Trans. on Information Theory*, vol. 18, pp. 14–20, Jan. 1972.

[70] R. E. Blahut, "Computation of channel capacity and rate distortion functions," *IEEE Trans. on Information Theory*, vol. 18, pp. 460–473, Jan. 1972.

[71] P. Jacquet and W. Szpankowski, "Entropy computations via analytic depoissonization," *IEEE Trans. on Information Theory*, vol. 45, no. 4, pp. 1072–1081, May 1999.

[72] F. Wang, D. Fertonani, and T. M. Duman, "Symbol-level synchronization and LDPC code design for insertion/deletion channels," *IEEE Trans. on Communications*, vol. 59, no. 5, pp. 1287–1297, May 2011.

[73] A. A. AlRustamani, A. D. Damnjanovic, and B. R. Vojcic, "Turbo greedy multiuser detection," *IEEE J. Select Areas Commun.*, vol. 19, no. 8, pp. 1638–1645, Aug. 2001.

[74] A. R. Iyengar, P. H. Siegel, and J. K. Wolf, "Write channel model for bit-patterned media recording," *IEEE Trans. on Magn.*, vol. 47, no. 1, pp. 35–45, Jan. 2011.

[75] S. Karakulak, P. H. Siegel, J. K. Wolf, and H. N. Bertram, "A new read channel model for patterned media storage," *IEEE Trans. on Magn.*, vol. 44, no. 1, pp. 193–197, Jan. 2008.

[76] M. Keskinoz, "Two-dimensional equalization/detection for patterned media storage," *IEEE Trans. on Magnetics*, vol. 44, no. 4, pp. 533–539, Apr. 2008.

[77] J. Hu, T. M. Duman, E. M. Kurtas, and M. F. Erden, "Bit patterned media with written-in errors: Modelling, detection and theoretical limits," *IEEE Trans. on Magnetics*, vol. 43, no. 8, pp. 3517–3524, Aug. 2007.

[78] Z. Wu and J. M. Cioffi, "Low-complexity iterative decoding with decision-aided equalization for magnetic recording channels," *IEEE Journal on Selected Areas in Communications*, vol. 19, no. 4, pp. 699–708, Apr. 2001.

[79] J. Wozencraft and B. Reiffen, *Sequential decoding*. Cambridge: The MIT Press, 1961.

[80] S. R. C. Weiss and J. Hagenauer, "Sequential decoding using a priori information," *Electronics Letters*, vol. 32, no. 13, pp. 1190–1191, Jun. 1996.

[81] T. M. Duman and A. Ghrayeb, *Coding for MIMO Communication Systems*. John Wiley & Sons, 2007.

[82] S. Loyka and F. Gagnon, "Performance analysis of the V-BLAST algorithm: an analytical approach," *IEEE Trans. on Wireless Communications*, vol. 3, no. 4, pp. 1326–1337, Jul. 2004.

[83] A. Roumy and D. Declercq, "Characterization and optimization of ldpc codes for the 2-user gaussian multiple access channel," *EURASIP Journal on Wireless Communications and Networking*, vol. 2007, Jun. 2007.

Appendix A

PROOFS OF RESULTS FROM CHAPTER 4

A.1 Proof of Lemma 1

Proof of Lemma 1. Define $\mathbf{D}^n$ to be an n-bit vector that contains a 1 if and only if the corresponding bit in $\mathbf{X}^n$ is deleted. We have $H(\mathbf{D}^n) = \frac{n}{b}(P_d \log_2 b + H(P_d))$. With this definition, the random processes $\mathbf{D}$ is non-stationary which even though $\mathbf{X}$ is stationary and ergodic. In order to make it stationary, we let the "first" segment of the channel start at a random position which is uniformly chosen from $\{1, 2, \ldots, b\}$, which does not affect the capacity. To prove this, we write $P(D_t = d_0, D_{t+1} = d_1, \ldots, D_{t+\tau} = d_\tau) = \sum_{i=1}^{b} P(D_t = d_0, D_{t+1} = d_1, \ldots, D_{t+\tau} = d_\tau | A_t = i) P(A_t = i)$, where $A_t = i$ is the event that X_t is the i-th bit of a segment. Clearly, $P(A_t = i)$ equals 0 or 1 for different t values and $P(D_t = d_0, D_{t+1} = d_1, \ldots, D_{t+\tau} = d_\tau | A_t = i) \neq P(D_t = d_0, D_{t+1} = d_1, \ldots, D_{t+\tau} = d_\tau | A_t = j)$, $\forall i \neq j,\ \tau > 0$. Hence, $\mathbf{D}$ is not stationary. When the input sequence starts at a random position of the segment, the positions of all the segment boundaries become random variables and $P(A_t) = 1/b$. Consequently, we have

$$
\begin{aligned}
&P(D_t = d_0, D_{t+1} = d_1, \ldots, D_{t+\tau} = d_\tau) \\
&\qquad = \frac{1}{b} \sum_{i=1}^{b} P(D_t = d_0, D_{t+1} = d_1, \ldots, D_{t+\tau} = d_\tau | A_t = i) \\
&\qquad = \frac{1}{b} \sum_{i=1}^{b} P(D_{t'} = d_0, D_{t'+1} = d_1, \ldots, D_{t'+\tau} = d_\tau | A_{t'} = i) \\
&\qquad = P(D_{t'} = d_0, D_{t'+1} = d_1, \ldots, D_{t'+\tau}),
\end{aligned}
$$

where $t \neq t'$. Hence, $\mathbf{D}$ becomes stationary. It is easy to deduce that

$$
\begin{aligned}
H(\mathbf{Y}|\mathbf{X}^n) &= H(\mathbf{D}^n, \mathbf{Y}|\mathbf{X}^n) - H(\mathbf{D}^n|\mathbf{Y}, \mathbf{X}^n) \\
&= H(\mathbf{D}^n|\mathbf{X}^n) - H(\mathbf{D}^n|\mathbf{Y}, \mathbf{X}^n) \\
&= H(\mathbf{D}^n) - H(\mathbf{D}^n|\mathbf{Y}, \mathbf{X}^n) \ . \qquad \text{(A.1)}
\end{aligned}
$$

The exact evaluation of the term $H(\mathbf{D}^n|\mathbf{X}^n, \mathbf{Y})$ is troublesome; however, under the condition that P_d/b is small, it can be bounded.

The following arguments follow similar steps as in [16], which considers the case of i.i.d. deletions. Let $\hat{\mathbf{D}}^n$ be the vector obtained by flipping "1"s in $\mathbf{D}^n$ for two cases. First, when a particular run (consecutive bits of the same value) experiences deletion errors, which is referred to as the error run, and the number of deletions exceeds one, we flip all 1s in $\mathbf{D}^n$ which are associated with that error run. Secondly, when different error runs span the same segment, we flip all 1s in $\mathbf{D}^n$ which are associated with these error runs. One example is given as follows. Suppose we transmit a sequence 001 000 001 110 over a segmented deletion channel with $b = 3$, and receive 01000110. Obviously, one bit gets deleted from each segment resulting in a total number of 24 possible realizations of $\mathbf{D}$ (one of the two 0's gets deleted from the first segment, one of the three 0's gets deleted from the second segment, one of the two 0's gets deleted from the third segment, and one of the two 1's gets deleted from the last segment). Since the third bit run (five consecutive 0's) have two deletion errors and the forth bit run with only one error but share the same segment with another error run, we assume an auxiliary channel that generates 01000001110 and the corresponding $\hat{\mathbf{D}}$ can only be either 100 000 000 000 or 010 000 000 000 with equal probability. By doing so, we guarantee that every deletion error from this auxiliary channel belongs to a bit run with a single deletion and every bit from that run can be deleted with an equal probability.

The process $\hat{\mathbf{D}} = f(\mathbf{D}, \mathbf{X})$ is also stationary with $P(\hat{D}_i = 1)$ being upper bounded by P_d/b. A lower bound on $P(\hat{D}_i = 1)$ can be obtained as follows. Let l_0 be the length of a bit run which contains X_i and spans $(j-m_1)$-th to $(j+m_2)$-th segments. When $\hat{D}_i = 1$, the $(j-m_1)$-th to $(j+m_2)$-th segments will not experience deletion error except the j-th segment, to which X_i belongs. Also, any bit from a run which starts from the $(j+m_2)$-th segment or ends in the $(j-m_1)$-th segment will not be deleted. Let l_1 and l_2 be the run lengths of the run which ends in the $(j-m_1)$-th segment and starts from the $(j+m_2)$-th segment, respectively. There is only one deletion error in these segments and it has to be in the j-th segment. Therefore, considering the worst case scenario, we have,

$$\begin{aligned} P(\hat{D}_i = 1) &\geq \sum_{l_1,l_2=1}^{\infty} \frac{P_d}{b}(1-P_d)^{l_0+l_1+l_2+4(b-1)} P_L(l_1,l_2) \\ &\geq \frac{P_d}{b} - (l_0 + E[l_1] + E[l_2] + 4(b-1))P_d^2. \end{aligned}$$

For any input process with a finite average run length, we can write $P(\hat{D}_i = 1) \in (P_d/b - K^* l_0 P_d^2, P_d/b)$, where $K^* < \infty$ is a nonnegative integer.

With the above definition of $\hat{\mathbf{D}}$ and letting $\hat{\mathbf{Y}}$ to be the outcome of $\mathbf{X}^n$ corresponding to the deletion pattern $\hat{\mathbf{D}}^n$, it is clear that runs with length $l = 1$ do not contribute to $H\left(\hat{\mathbf{D}}^n | \mathbf{X}^n, \hat{\mathbf{Y}}\right)$. Furthermore, no run with more than one deletion can contribute to $H\left(\hat{\mathbf{D}}^n | \mathbf{X}^n, \hat{\mathbf{Y}}\right)$ as they all have been reversed. Therefore, only runs with length $l \geq 2$ and one deletion lead to a contribution of $\log_2 l$ to $H\left(\hat{\mathbf{D}}^n | \mathbf{X}^n, \hat{\mathbf{Y}}\right)$ since the deleted bit is uniformly chosen, which is guaranteed by the definition of $\hat{\mathbf{D}}$ and the channel model. Finally, we conclude that $H(\hat{\mathbf{D}}^n | \mathbf{x}^n, \hat{\mathbf{y}}) = \sum_{r \in \mathbb{R}} \log_2(l_r)$, where $\mathbb{R}$ is the set of runs on which deletions occur and l_r is the corresponding run length. Therefore, from [16],

for any stationary ergodic process such that $E[L_0 \log_2 L_0] < \infty$, we have

$$\lim_{n\to\infty} \frac{1}{n} H\left(\hat{\mathbf{D}}^n | \mathbf{X}^n, \hat{\mathbf{Y}}\right) = \frac{P_d}{b} E[\log_2 L_0] - \delta \ , \tag{A.2}$$

where $0 \le \delta \le K^* P_d^2 E[L_0 \log_2 L_0]$.

Define $\mathbf{Z} = \mathbf{D} \oplus \hat{\mathbf{D}}$, which represents the difference between $\mathbf{D}$ and $\hat{\mathbf{D}}$. The process $\mathbf{Z}$ is stationary with $z = P(\mathbf{Z}_i = 1) \le K^* E[L_0] P_d^2$. Note that $(\mathbf{X}^n, \hat{\mathbf{Y}}, \hat{\mathbf{D}}^n)$ is a function of $(\mathbf{X}^n, \mathbf{Y}, \mathbf{D}^n, \mathbf{Z}^n)$, we have $|H(\mathbf{X}^n, \mathbf{Y}, \mathbf{D}^n) - H(\mathbf{X}^n, \hat{\mathbf{Y}}, \hat{\mathbf{D}}^n)| = |H(\mathbf{X}^n, \mathbf{Y}, \mathbf{D}^n) - H(\mathbf{X}^n, \mathbf{Y}, \mathbf{D}^n, \mathbf{Z}^n)| = H(\mathbf{Z}^n | \mathbf{X}^n, \mathbf{Y}, \mathbf{D}^n) \le H(\mathbf{Z}^n)$. Same argument also holds for $|H(\mathbf{X}^n, \mathbf{Y}) - H(\mathbf{X}^n, \hat{\mathbf{Y}})|$. Therefore from [16], $|H(\mathbf{D}^n | \mathbf{X}^n, \mathbf{Y}) - H(\hat{\mathbf{D}}^n | \mathbf{X}^n, \hat{\mathbf{Y}})| \le 2H(\mathbf{Z}^n) \le 2nH(z)$ Hence, the following equation follows,

$$H(\mathbf{Y}|\mathbf{X}^n) = H(\mathbf{D}^n) - H(\hat{\mathbf{D}}^n | \mathbf{X}^n, \hat{\mathbf{Y}}) + n\delta' \ , \tag{A.3}$$

where $-2H(z) \le \delta' \le \delta + 2H(z)$. Combining (A.2) and (A.3), we obtain

$$\lim_{n\to\infty} \frac{1}{n} H\left(\mathbf{Y}|\mathbf{X}^n\right) = \frac{P_d}{b} \log_2 b + \frac{H(P_d)}{b} - \frac{P_d}{b} E[\log_2 L_0] + \delta' \ . \tag{A.4}$$

For the input process $\mathbf{X}^*$, it is easy to verify that $E[L_0 \log_2 L_0] < \infty$. In this case, $z = O(P_d^2)$, and therefore, $\delta' = O(P_d^{2-\epsilon})$ for any $\epsilon > 0$. Hence, from (A.4), the lemma is proved. □

A.2 Proof of Lemma 2

Proof of Lemma 2. Lemma 2 provides a lower bound on $\lim_{n\to\infty} \frac{1}{n} H\left(\mathbf{Y}|\mathbf{X}^n\right)$. Based on the result given in (A.4), the only work is to quantify the lower bounds on δ' and $E[\log_2 L_0]$ for any stationary ergodic process.

First of all, (A.3) states that $\delta' \ge -2H(z)$. From the proof of Lemma 1, we have $z = P(\mathbf{Z}_i = 1) \le K^* E[L_0] P_d^2$. According to [16] (Lemma IV.3), for any stationary ergodic process satisfying the condition $H(\mathbf{X}) > 1 - \left(\frac{P_d}{b}\right)^{1-\gamma}$

$(\gamma > 0)$, the mean of the bit perspective run length $E[L_0] \leq K'(1+\left(\frac{P_d}{b}\right)^{1/2-\epsilon'} L^*)$, $K' < \infty$ for any integer L^*. Combining the upper bound on z and $E[L_0]$, we conclude that $H(z) \leq K'' P_d^{2-\epsilon}(1 + P_d^{1/2} L^*)$ $\forall P_d < P_{d,0}$ and consequently $\delta' \geq -K' P_d^{2-\epsilon}(1 + P_d^{1/2} L^*)$ [16], where $K' < \infty$ is a positive integer. Also from [16] (Lemma IV.3), we have $|A - E[\log_2 L_0]| = O(P_d^{1/2-\epsilon} \log_2 L^*)$. Combining these results with (A.4), the lemma is proved. □

A.3 Proof of Lemma 3

Proof of Lemma 3. In this case, define $\hat{\mathbf{D}}^n$ to be generated by flipping the ones in $\mathbf{D}^n$ when the corresponding error run spans two segments, which is different from the one defined in the proof of Lemma 1. In order to obtain a stationary process $\hat{\mathbf{D}}$, we still let the first segment of the input process start at a random position which is uniformly chosen from $\{1, 2, \ldots, b\}$.

For any stationary and ergodic process $\mathbf{X}$, the starting point of a bit run is uniformly distributed within the segment[1]. Also, since the positions of the segment boundaries are random with a uniform distribution, the probability that the error run with length l_0 spans two segments is $\frac{l_0-1}{b}$, if we restrict the input process $\mathbf{X} \in S_b$, i.e., $l_0 \leq b$. Therefore, it is clear that $P(\hat{D}_i = 1) = \frac{P_d}{b}(1 - \frac{l_0-1}{b})$. Also, with the same definition of $\mathbf{Z}$ as in the proof of Lemma 1, we have $z = P(Z_i = 1) \leq b^{-2} E[L_0]$. Following the same steps of the proof in Lemma 1, we have, for any stationary ergodic process $\mathbf{X} \in S_b$ such that $E[L_0 \log_2 L_0] < \infty$,

$$\lim_{n\to\infty} \frac{1}{n} H\left(\hat{\mathbf{D}}^n | \mathbf{X}^n, \hat{\mathbf{Y}}\right) = \frac{P_d}{b} E[\log_2 L_0] - \frac{P_d}{b^2} E[(L_0 - 1)\log_2 L_0] \,. \qquad \text{(A.5)}$$

[1] To see this, let us first consider the case of $b = 2$ and suppose that the bit run starts at the first bit of the segment with probability p_1 and at the last bit of the segment with probability p_2. Clearly, $p_1 = p_1 \cdot p_{\text{even}} + p_2 \cdot p_{\text{odd}}$, where p_{even} and p_{odd} are the probabilities of the run length being an even or odd number, respectively. Since $p_{\text{even}} = 1 - p_{\text{odd}}$, we have $p_1 = p_2 = 0.5$. Extension to the general case is straightforward and the detailed proof is omitted.

Substituting (A.5) into (A.3), the following result appears under the same condition,

$$\lim_{n\to\infty}\frac{1}{n}H\left(\mathbf{Y}|\mathbf{X}^n\right)=\frac{P_d}{b}\log_2 b+\frac{H(P_d)}{b}-\frac{P_d}{b}E[\log_2 L_0]+\delta'\,, \tag{A.6}$$

where $-2H(z)\le\delta'\le\delta+2H(z)$, $\delta=b^{-2}E[L_0\log_2 L_0]$.

For the process $\mathbf{X}_b^*\in S_b$, $z\le b^{-2}E[L_0]=O(b^{-2})$, and therefore, $H(z)=O(b^{-2+\epsilon})$. Since $-2H(z)\le\delta'\le b^{-2}E[L_0\log_2 L_0]+2H(z)$ and it is easy to verify in this case $E[L_0\log_2 L_0]<\infty$, we conclude that $\delta'=O(b^{-2+\epsilon})$ for any $\epsilon>0$. Hence, (4.16) is proved.

To show (4.15), we follow the same rationale in the proof of of Lemma 2. since $z\le b^{-2}E[L_0]$ and for any stationary ergodic process satisfying the condition $H(\mathbf{X})>1-\left(\frac{P_d}{b}\right)^{1-\gamma}$ ($\gamma>0$), $E[L_0]\le\kappa'(1+\left(\frac{P_d}{b}\right)^{1/2-\epsilon'}b)$ (let $L^*=b$), we get $H(z)\le\kappa^\star b^{-2+\epsilon}(1+b^{1/2})\ \forall b>b_0$. Using the conclusion that $|A-E[\log_2 L_0]|=O(b^{-1/2+\epsilon})$ [16] (Lemma IV.3), the result follows.

□