DESIGN OF CAPACITY-APPROACHING PROTOGRAPH-BASED

LDPC CODING SYSTEMS

by

Thuy Van Nguyen

APPROVED BY SUPERVISORY COMMITTEE:

---

Dr. Aria Nosratinia, Chair

---

Dr. John P. Fonseka

---

Dr. Kamran Kiasaleh

---

Dr. Murat Torlak

*Dedicated to my family.*

# DESIGN OF CAPACITY-APPROACHING PROTOGRAPH-BASED LDPC CODING SYSTEMS

by

THUY VAN NGUYEN, B.S., M.S.

DISSERTATION

Presented to the Faculty of

The University of Texas at Dallas

in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY IN

ELECTRICAL ENGINEERING

THE UNIVERSITY OF TEXAS AT DALLAS

December 2012

ACKNOWLEDGMENTS

I am very fortunate to have Dr. Aria Nosratinia as my adviser. His insightful thinking and endless enthusiasm helps change his graduate students' lives for the better, including mine. I am grateful for his continual patience and encouragement that helped me express and clarify my research ideas. In so many ways he has helped me to put my research into larger context by drawing out similarities and contrasts to other works, pointing out which problems are important and where the ideas can be applied. I am grateful for his emphasis on good writing and presentation. I will never forget how he often spent a great amount of time rewriting just one small paragraph for our papers. This dissertation would not have been possible without his guidance and support. I hope that I have been able to grasp his research philosophy: work on important problems, think big and verify all results. More importantly, I really hope that I have been able to acquire his optimism, determination, kindness and generosity of heart.

I would like to thank Dr. Dariush Divsalar, Jet Propulsion Laboratory, California Institute of Technology. Collaborating with him has been an exciting and rewarding research experience. I would also like to thank Dr. John P. Fonseka, Dr. Kamran Kiasaleh and Dr. Murat Torlak for agreeing to be members of my dissertation committee and all the valuable suggestions that make this dissertation better.

I am very thankful to all my current and former colleagues at the Multimedia Communication Laboratory for their friendship, including Yang Li, Ahmed Hesham Mehana, Mohamed Abou El-Seoud, Negar Bazargani, Youngsam Yoon, Xing Hua, Ramy M. Tannious, Mohamed Fadel, Ahmed Hindy and Ahmed Attial Abotabl. Their suggestions and discussions enlightened me and refined my understanding of the field. In addition, I would like to thank my Vietnamese friends in the Dallas area. We have spent precious moments together, and I am fortunate to know each of you.

I am also grateful to the VOSP program, Ministry of Education and Training, Vietnam for their financial support in the first two years of my Ph.D. studies.

Last but not the least I must express my deepest appreciation for the support and encouragement of my wife Hanh Do and my daughters Alice and Emily, as well as the trust and confidence of my parents, sisters, and brothers throughout my life. I am indebted to all of them for their loving support, sacrifices, and positive attitude that gave me the strength and confidence to follow my dreams. Without them I would be nothing. I dedicate this dissertation to the loving memory of my grandmother, Ho Thi Nguyen, whose role in my life was, and remains, immense.

October 2012

# PREFACE

This dissertation was produced in accordance with guidelines which permit the inclusion as part of the dissertation the text of an original paper or papers submitted for publication. The dissertation must still conform to all other requirements explained in the "Guide for the Preparation of Master's Theses and Doctoral Dissertations at The University of Texas at Dallas." It must include a comprehensive abstract, a full introduction and literature review, and a final overall conclusion. Additional material (procedural and design data as well as descriptions of equipment) must be provided in sufficient detail to allow a clear and precise judgment to be made of the importance and originality of the research reported.

It is acceptable for this dissertation to include as chapters authentic copies of papers already published, provided these meet type size, margin, and legibility requirements. In such cases, connecting texts which provide logical bridges between different manuscripts are mandatory. Where the student is not the sole author of a manuscript, the student is required to make an explicit statement in the introductory material to that manuscript describing the student's contribution to the work and acknowledging the contribution of the other author(s). The signatures of the Supervising Committee which precede all other material in the dissertation attest to the accuracy of this statement.

DESIGN OF CAPACITY-APPROACHING PROTOGRAPH-BASED

LDPC CODING SYSTEMS

Publication No. _____

Thuy Van Nguyen, Ph.D.
The University of Texas at Dallas, 2012

Supervising Professor: Dr. Aria Nosratinia

In this dissertation, a systematic framework is proposed to design practical protograph-based low-density parity check (LDPC) coding schemes that address simultaneously several important issues: structured coding that permits easy design, low encoding complexity, embedded structure for convenient adaptation to various channel conditions, and performance close to capacity with a reasonable block length. This dissertation consists of four closely inter-related parts. In the first part, the design of rate-compatible protograph codes for the hybrid automatic repeat request protocol is presented. A high-performance family of protograph codes that has the iterative decoding threshold within a gap of a fraction of dB to capacity in the AWGN channel over a wide range of rates is reported. In the second part, protograph-based LDPC coding schemes are designed for half-duplex relay channels. A simple new methodology for evaluating the end-to-end error performance of relay coding systems is then developed and used to highlight the performance of the proposed codes. In the third part, a general mapping method is devised for using protograph-based LDPC codes in bit-interleaved coded modulation. The reported coding scheme operates close to the coded modulation capacity. In the fourth part, the design of rate-compatible protograph codes in

inter-symbol interference channels is proposed. The design problem is non-trivial due to the joint design of structured LDPC codes and the state structure of ISI channels using the BCJR equalizer. High-performance protograph-based LDPC codes that have iterative thresholds close to i.u.d capacity of ISI channels are reported.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

# CHAPTER 1
# INTRODUCTION

## 1.1 Background

Since the 1948 celebrated results of Shannon [1], one of the most important questions in the coding theory has been to design practical coding schemes to achieve the Shannon capacity [2]. Low density parity check (LDPC) codes, proposed by Gallager [3] in 1963, were a notable early result in this direction, were forgotten for almost 30 years, and then regained attention in late 1990's [4, 5]. A few notable activities in the interim did exist, chief among them the works of Zyablov and Pinsker [6] in 1975, Tanner [7] in 1981, and Margulis [8] in 1982. By now it is known that LDPC codes can operate within a fraction of a dB to capacity of many channels, such as additive white Gaussian noise (AWGN) channels [9, 10], fading channels [11], etc.

The new wave of interest in LDPC codes produced several deep and important results in quick succession. Among them the concept of *degree distributions* first appeared in [12] and code ensemble analysis [9]. These results showed the superiority of irregular LDPC codes over the regular LDPC codes originally investigated by Gallager. The behavior of iterative decoding was characterized by the notion of *density evolution* [13, 9], and later through its variations such as the EXIT chart [14].

Through these advances, the capacity-approaching qualities of the LDPC codes were firmly established through codes designed with random structures and long block lengths. This, however, was not the end of the story from an engineering viewpoint. Random-like LDPC codes are not easy to encode, because their generator matrix is in general not sparse. Therefore, a search continued for efficient LDPC-like coding structures that can provide at the same time excellent performance and small complexity. Divsalar et al. [15] in 1998

introduced "repeat-accumulate" (RA) codes as a simple "turbo-like" structure. Then Lin et al. [16] showed that good LDPC codes can be built from finite-geometry. In 2003, protograph-based LDPC codes were proposed [17] as a method of constructing large tanner graphs systematically from a smaller protograph. In 2004, Richardson and Urbanke [18] introduced multi-edge-type LDPC codes as a generalization of regular and irregular LDPC codes, and it became evident that protographs are a special case of multi-edge-type graphs. Protograph codes have been shown to achieve the capacity of the AWGN channel closely, e.g. [19, 20] and reference therein.

### 1.2 Motivation and Objectives

The protograph family of LDPC codes have among them a few excellent codes (e.g. the AR4JA codes [20]), and new and exciting directions such as convolutional LDPC codes [21] which have surfaced that are closely related to protographs. However, on the question of protograph code designs, the picture has been incomplete: the design of protograph codes prior to the work in this dissertation had been limited to manual design via trial-and-error. This has several drawbacks: first, without a comprehensive computer search in the code space, it is not possible to know whether the best code within a certain class has been found. Second, it is not possible to design larger protographs reliably by trial and error (there are too many variables), and larger protographs are sometimes necessary for good performance and especially for producing nested codes. The central contribution of this dissertation is to produce a systematic method for designing protograph codes that lends itself to computer search, which has resulted in uncovering superior previously-unknown codes, and also has led to efficient code designs for channels other than the point-to-point memoryless AWGN channel.

The overall objective of the work in this dissertation has been to produce efficient methods for the design of protographs and corresponding protograph codes. This design task, in general, involves the characterization of desirable code properties such as growth of minimum distance and near-capacity threshold in terms of constraints on the structure of the

protograph. It has been also been an objective of this dissertation to use this systematic design procedure to discover excellent protograph codes for a variety of applications. For many of these applications, protograph LDPC codes did not exist prior to this dissertation.

### 1.3 Contributions and Outline

Chapter 2 provides definitions and background of protograph-based LDPC codes as well as their performance measures that will be used in this dissertation.

Chapter 3 produces a systematic framework to design a good protograph in the AWGN channel. Based on this framework, a high-rate nested family of protograph codes is designed with rates from 1/2 to 9/10. We also design a rate-compatible family by extension with applications in hybrid automatic repeat request (HARQ) systems, covering a wide range of rates from 0.33 to 0.9. The new nested and rate-compatible families of protograph codes not only inherit the advantages of protograph codes, namely low encoding complexity and efficient decoding algorithms, but also have very good performance with iterative decoding thresholds that are close to their capacity limits.

Chapter 4 studies the problem of designing protograph codes over relay channels with the decode-and-forward protocol. Two coding strategies for the relay channels are considered: the bilayer-lengthened and bilayer-expurgated structures. We present capacity-approaching protograph-based LDPC coding schemes using these bilayer structures for the half-duplex relay channels. The new structured relay coding schemes not only have low encoding/decoding complexity but also the embedded structure for convenient adaptation for various channel conditions. Additionally, the application of the coding structure to multi-relay networks is demonstrated. Finally, we also develop a new methodology for evaluating the end-to-end error performance of relay coding systems that is used to highlight the performance of the proposed codes.

Chapter 5 studies the problem of designing high order coded modulation schemes over Rayleigh fading channels. We present a simple method involving the concatenation of binary

protograph codes and a specially designed mapping. The proposed method is general and applies to any modulation. We calculate the iterative decoding thresholds of the protograph codes while mapped to higher order modulations. This coded modulation framework can support not only multiple rates but also adaptive modulation. We discover families of protograph codes that achieve a threshold within a gap of approximately $0.2 - 0.4$ dB of the BICM capacity limit across a wide range of rates and modulations.

Chapter 6 studies the problem of designing structured LDPC codes in inter-symbol interference (ISI) channels. This problem is non-trivial due to the joint design of structured (protograph-based) LDPC codes and the state structure of ISI channels. A method of computing the iterative decoding threshold of a concatenated system between the LDPC decoder and the BCJR equalizer is proposed. This allows us to design nested and rate-compatible protograph codes that can operate closely to the independent and uniformly distributed (i.u.d.) capacity of ISI channels. Numerical results are provided to support our analysis.

# CHAPTER 2
# PROTOGRAPH-BASED LDPC CODES

This chapter is dedicated to reviewing background material regarding LDPC codes and protograph-based LDPC codes, including the introduction of performance measures used in this dissertation.

## 2.1 LDPC Codes

LDPC codes are defined via the column null-space of a sparse $m \times n$ binary matrix $H$ called a parity check matrix, i.e., $cH^t = 0$. The common notation used in this dissertation includes the following: $H^t$ is a transposed matrix, $m$ and $n$ are the number of rows and columns of $H$, respectively, and $c$ is a binary row vector indicating a codeword of length $n$. The matrix $H$ is sparse if the number of 1's in each row or column is small compared with $m$ or $n$, respectively.

An LDPC code can be fully represented either by a matrix $H$ or via an equivalent bipartite graph (also known as Tanner graph) as shown in Figure 2.1. In the figure there are

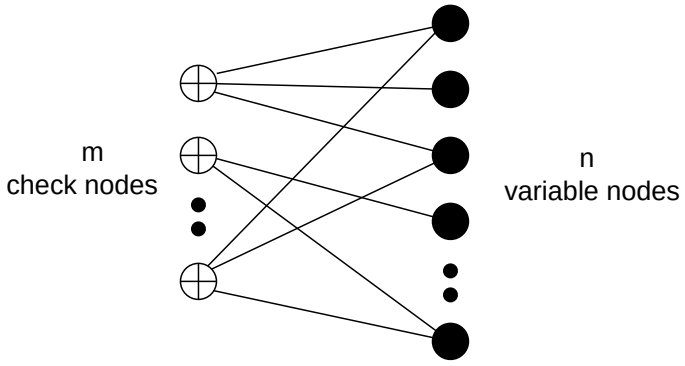


Figure 2.1. Tanner graph of a general LDPC code

two types of nodes: $n$ dark circles represent $n$ variable nodes (or $n$ columns of $H$), $m$ plus circles represent $m$ check nodes (or $m$ rows of $H$), and edges that connect variable nodes to check nodes correspond to the 1's in $H$.

The number of edges connecting to a node is called the degree of the node. An LDPC code is called *regular* if all its variable nodes share the same degree, and all the check nodes share the same degree. An LDPC code is *irregular* if it is not regular. Since LDPC codes have sparse parity check matrices, they facilitate efficient iterative decoding (e.g. the iterative sum-product algorithm) which allows building low complexity decoders. Even though iterative decoding algorithms are sub-optimal, they often provide near-optimal performance [22, 23]. The iterative decoding process improves the log-likelihood ration (LLR) of the coded bits in a step-by-step process.

A general tool to analyze LDPC codes using the iterative sum-product decoder is the density evolution (DE) algorithm [9]. The DE algorithm analyzes the evolution of probability density of the LLRs throughout the iterative decoding process for long codewords and large number iterations. DE analysis highlights the conditions of the channel parameter (e.g. noise power or SNR) that are required for error-free performance of LDPC iterative decoder. This level of noise or SNR beyond which the iterative decoder fails is called the iterative decoding threshold of an LDPC code. A good irregular LDPC code is designed by optimizing the threshold, itself accomplished by selecting the appropriate degree distribution for the code [9]. An optimized rate-1/2 irregular LDPC code [24] has been reported with an iterative decoding threshold within 0.0045 dB to capacity.

The above reported irregular LDPC codes are non-structured or random codes. These random codes often require high encoding complexity,[1] preventing them from being used in practical systems. There have been many research attempts to design good structured LDPC codes [23] having cyclic or quasi-cyclic (QC) structures. These structures facilitate low encoding complexity with simple feedback shift registers [25].

[1]Because the generator matrix of an LDPC code is in general not sparse.

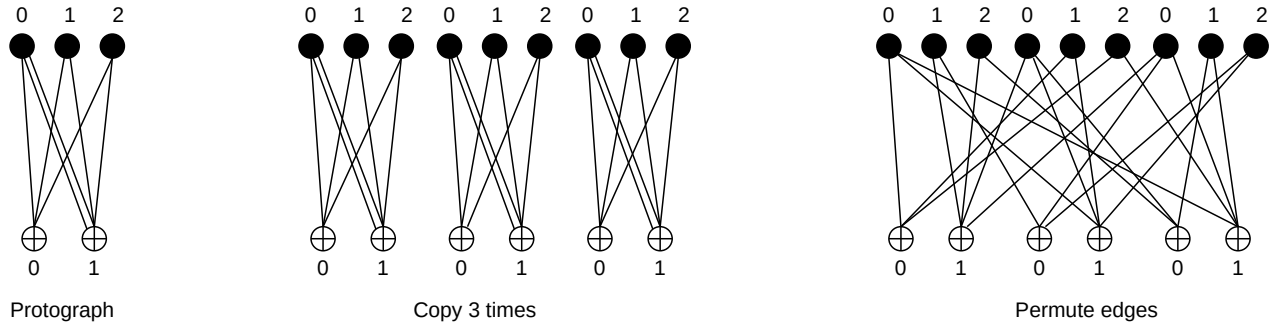


Figure 2.2. Copy and permute operation for a protograph to generate larger graphs

In the following, we present one special QC code, namely protograph-based LDPC codes, which are the main topics of this dissertation.

## 2.2 Protograph-Based LDPC Codes

A protograph [17] is a Tanner graph with a relatively small number of nodes, connected by a small number of edges, allowing parallel edges between two nodes. A protograph is typically represented by a protomatrix, i.e., a matrix whose entries indicate the number of edges connecting the respective variable and check nodes. Therefore, unlike the parity check matrix, the protomatrix is in general non-binary.

A protograph code (an equivalent LDPC code) is a larger derived graph constructed by applying a "copy-and-permutation" operation on a protograph. The protograph is copied $N$ times, then a large LDPC code graph is obtained by permuting $N$ variable-to-check pairs (edges), corresponding to the same edge type of the original protograph. A simple example of a protograph is shown in Figure 2.2. This graph consists of 3 variable nodes and 2 check nodes that are interconnected by 7 different edge (variable-to-check) types. The derived graph is constructed by replicating the protograph 3 times and permuting variable-to-check pairings within the same edge type of the protograph. For example, the protomatrix of protograph in Figure 2.2 is in the following form

$$H_{proto} = \begin{pmatrix} 1 & 1 & 1 \\ 2 & 1 & 1 \end{pmatrix} \tag{2.1}$$

where the rows and columns represent the check nodes and variable nodes in the graph respectively. The $(i, j)$ element of the protomatrix indicates the number of parallel edges that connect the check node $i$ and the variable node $j$. The protograph code graph has $N$ times as many nodes as the protograph, but compared with the protograph it has the same rate as well as the same degree distribution for both variable and check nodes. Protograph LDPC codes are a subclass of multi-edge type LDPC codes [18] where each edge of a protograph is one edge type. Protograph codes can achieve capacity-approaching performance with low encoder complexity as well as fast decoding [26, 20].

## 2.3 Performance of LDPC Codes

The Shannon capacity of the binary-input AWGN channel determines the maximum rate supported at a given SNR. The BI-AWGN capacity does not have a closed-form expression, but regardless it is a monotonic function that can be inverted, so that at a given rate one may find the minimum SNR that supports that rate. This is often how Shannon capacity is displayed in the context of coding: as a SNR limit for each rate.

Figure 2.3 shows a typical behavior for an LDPC code in the AWGN channel. An LDPC code has an iterative decoding threshold that is above the capacity limit. If the code threshold is close to capacity, we say that the code is capacity-approaching. The performance of LDPC codes has two regions, the waterfall and error floor regions. The waterfall region indicates the regime of operation where error rate reduces significantly when SNR increases, i.e., a sharp slope. At higher SNR, some codes experience a slower slope for error probability that is due either to reaching the limit imposed by the minimum distance of the code, or to the limits of iterative decoding performance due to graph imperfections.

It is challenging to design an LDPC code that has good performance in both the waterfall and error floor regions. From Figure 2.3, the waterfall performance can be improved by

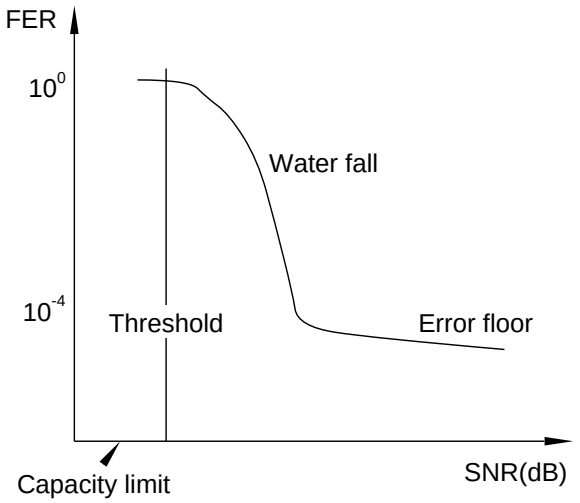


Figure 2.3. A general performance curve of an LDPC code

optimizing the code threshold based on density evolution. That is equivalent to shift the error performance curve to the left, close to the capacity limit. The error floor performance has proved difficult to characterize explicitly: it depends on graph imperfections of finite-length codes due to non-codewords such as stopping set and trapping sets as discussed in [27] and references therein.

Improving the error floor performance of LDPC codes is critical sine many systems, such as data-storage devices and optical communication systems, require extremely low error rates. There are two main strategies to solve this problem, i.e. novel LDPC code designs and decoder-based algorithms. The former [20] involves a careful design so that the code structure avoids graph imperfections of short cycles that form trapping sets [27]. The latter [23] involves designing modified iterative decoders so that they are able to recover from trapping sets.

In this dissertation, we focus the LDPC code design with two main parameters to control these performance regions: the iterative decoding threshold which affects the waterfall performance, and the linear minimum distance growth property which affects the error floor performance.

### 2.3.1 Iterative Decoding Thresholds of Protograph Codes

As mentioned in Section 2.2, a protograph code has the same degree distribution as that of the protograph. Thus, the iterative decoding threshold of a protograph-based LDPC code is the same as that of the protograph; this reflects the minimum channel SNR that supports reliable iterative decoding of asymptotically large LDPC codes built from the protograph. The decoding threshold can be computed by using density evolution (DE) [9]. The density evolution algorithm tracks the distribution of messages exchanged in the Tanner graph, but it is computationally cumbersome, especially for our purposes because we intend to use it inside an optimization loop as explained in the next chapter. There are approximate techniques that reduce the complexity of the calculation of the decoding threshold: the extrinsic information transfer (EXIT) chart [14] and reciprocal channel approximation (RCA) [20]. The EXIT chart and its variants are more popular methods for fast calculation of threshold, and in our dissertation we follow that route.

To consider further the threshold calculation using the EXIT chart, we need to make a note of a property of protograph codes. The permutation step that builds the LDPC code from its protograph only permutes the edges among the same node from different copies of the protograph. Therefore, even though an LDPC code is built that has the same degree distribution as the protograph, the edges have more structure than implied by the degree distribution, and that structure is enforced by the protograph. To put this in yet another way, it is possible to have two LDPC codes possessing the same degree distribution that arise from two different protographs; these two LDPC codes may not have the same decoding threshold (because a protograph does not allow complete randomization). As a result, the decoding threshold of a protograph code is a function of both the degree distribution as well as its protograph protomatrix.

Because of this dependency the general EXIT chart, which is designed for generic LDPC ensembles, cannot be accurately applied to protograph codes: it can neither account for degree-1 variables nor the dependency on the protomatrix as mentioned above [28]. To solve these problems, Liva and Chiani [28] proposed the PEXIT method. The PEXIT algorithm

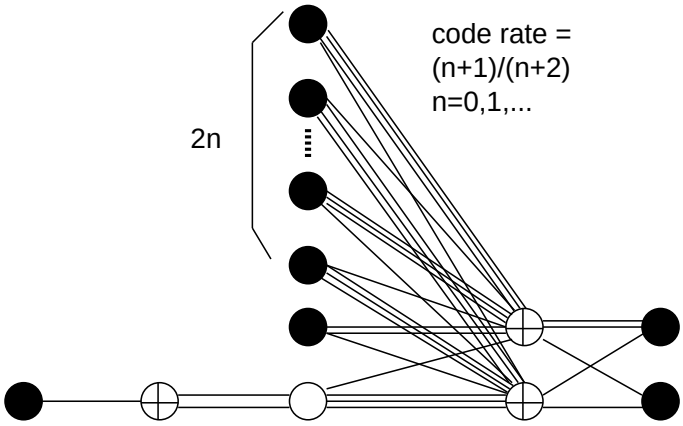


Figure 2.4. The AR4JA family of protograph with rates 1/2 and higher

computes the iterative threshold of a protograph based on its edge degree distribution (via its protomatrix) rather than node degree distribution as in general EXIT chart [14]. The PEXIT method will be used throughout this dissertation. For completeness, a summary of the PEXIT algorithm and the two-step lifting method to design protograph codes using the PEG algorithm are given in the appendix.

**The AR4JA Codes**

In the following, we will provide a description of AR4JA codes reported in [20] as reference codes in this dissertation. The family of AR4JA protograph codes are plotted in Figure 2.4. In this figure, the dark circles represent transmitted variable nodes, the white circle is a punctured node and the circles with a plus sign are parity check nodes. The graph contains $4+2n$ transmitted variable nodes and 3 check nodes that is equivalent to code rate $R = \frac{n+1}{n+2}$, with $n = 0, 1, \ldots$. The rate-1/2 AR4JA protograph ($n = 0$) has a protomatrix in the following form

$$H_{1/2} = \begin{pmatrix} 1 & 2 & 0 & 0 & 0 \\ 0 & 3 & 1 & 1 & 1 \\ 0 & 1 & 2 & 2 & 1 \end{pmatrix} \tag{2.2}$$

Table 2.1. Thresholds of AR4JA code family using PEXIT chart

| Code Rate | RCA Thres. (dB)[20] | PEXIT Thres. (dB) | Capacity Thres. (dB) | PEXIT gap to cap. |
|---|---|---|---|---|
| 1/2 | 0.628 | 0.6337 | 0.187 | 0.4467 |
| 2/3 | 1.450 | 1.4256 | 1.059 | 0.3666 |
| 3/4 | 2.005 | 1.9707 | 1.626 | 0.3447 |
| 4/5 | 2.413 | 2.387 | 2.040 | 0.3470 |
| 5/6 | 2.733 | 2.5924 | 2.362 | 0.2304 |
| 6/7 | 2.993 | 2.8935 | 2.625 | 0.2685 |

The other codes in this family with rate $R = \frac{n+1}{n+2}$, have protomatrices in the form of

$$H_{\frac{n+1}{n+2}} = \left(\begin{array}{c|cc} & 0 & 0 \\ H_{\frac{n}{n+1}} & 3 & 1 \\ & 1 & 3 \end{array}\right) \tag{2.3}$$

where $n = 1, 2, \ldots$. The nested codes with rate of $\frac{n+1}{n+2}$ are built by extending from the rate-1/2 code with the protomatrix of Equation. (2.2).

Thresholds of the AR4JA family of protograph codes [20] plotted in Figure 2.4 are shown in Table 2.1. For comparison purposes, the performance of protographs using RCA [20] is also given. As seen in the table, the AR4JA codes can perform within a gap of 0.5 dB to capacity in AWGN channel.

### 2.3.2 Linear Minimum Distance Growth Property

The previous section discusses the protograph threshold calculations that are used to design a code that that has good waterfall performance. However, it is not as easy to handle the error floor performance since the error floor involves graph imperfections due to non-codewords such as stopping sets and trapping sets [27].

Finding trapping sets has complexity that grows with codeword blocklength. However, experiments show that if the minimum distance of an LDPC code grows linearly with codeword blocklengths, the LDPC code will asymptotically have no error floors[2] [29]. We also

[2]If a random permutation per each edge of its protograph is used.

show [30] that if the protograph codes satisfy the minimum distance growth ($d_{min}$) property, the minimum size of small trapping sets of these codes also grows with the code blocklength. Since the error floor performance depends mostly on the small trapping sets [27], the code that satisfies the $d_{min}$ property can have no error floor if the code blocklength is large enough. The linear minimum distance growth property is determined via calculating the ensemble weight enumerator for an LDPC code built from a protograph [20]. The AR4JA codes represented in the previous subsection also have the linear minimum distance property [20].

# CHAPTER 3
# THE DESIGN OF RATE-COMPATIBLE PROTOGRAPH LDPC CODES

A rate-compatible family of codes allows coding across a range of rates using a common encoder/decoder infrastructure [31, 32, 33, 34, 35, 36, 37, 38]. This allows convenient matching of the code rate to channel conditions and also provides an efficient realization for incremental hybrid automatic repeat request (HARQ). In HARQ, upon an unsuccessful reception, the transmitter transmits incremental parity bits to allow the decoder to decode the data with a lower-rate (more powerful) code. This requires a rate-compatible family of codes.

The most common way of generating a rate-compatible family of codes is by puncturing, i.e., starting with a low-rate mother code and then selectively discarding some of the coded bits (parity bits) to arrive at higher rate codes [32, 33, 34]. This approach is simple but is not free of problems [39]. Specifically, (1) the mother code is optimally designed for low rates, so higher-rate punctured codes have iterative decoding thresholds with a wider gap to capacity than that of the mother code; (2) the optimal low rate code structure and puncturing patterns are designed separately, which is suboptimal. Even though it has been shown that puncturing can theoretically achieve the same gap to capacity as the mother code, in practice puncturing has increased the gap significantly [40, 32].

In this chapter, we propose a simple, systematic procedure to search for good nested protograph LDPC codes that have a low decoding threshold (close to the capacity limit) and also a minimum distance that grows linearly with block-length [20, 3], a necessary condition for avoiding an error floor. A protograph code is an LDPC code that can be constructed from a small protomatrix with a few elements [17]. We start by producing nested protograph codes where the parity check matrix of higher rate code is a lengthened version of the lower rate one. This lengthening structure is motivated by [20] but the resulting codes are better than those reported in [20]. This family is suitable for applications where adaptive

coding and/or unequal error protection is required as well as low complexity. However, code lengthening changes the information block size, thus the resulting codes are not useful for some applications, e.g., HARQ.

To achieve full rate-compatibility, we use another approach involving code extension. Code extension starts with a high-rate code (a daughter code), then lower rate codes are obtained by extending the parity check matrix of the daughter code [39, 41, 42, 35]. Most existing extension-based LDPC codes [39, 41, 35] are designed as rate-compatible irregular LDPC codes with highly optimized framework and unstructured design that does not promote low-complexity encoding. In contrast, the proposed rate-compatible protograph-based codes can achieve very good thresholds with low encoding complexity allowed by circulant permutations [20, 26].

Several protograph-based rate-compatible codes have appeared in the literature [43, 20, 44, 45, 46]. While some of these codes have reasonably good performance, the designs are by trial-and-error and thus time consuming. Some designs manually manipulate within a given structure [20, 19] or use node splitting and edge growth techniques [20, 44]. But without a comprehensive search one may easily miss good codes in this class, in fact this chapter shows the protograph codes mentioned above can be improved upon. Furthermore, manual design makes it impractical to use larger protographs that are needed for implementing a wide range of rates, and are helpful for improved performance. Also noteworthy is [42] which produced rate-compatible protograph codes based on node-splitting and attaching additional accumulators, but [42] only produces rates higher than 1/2 and furthermore the resulting iterative decoding threshold gaps to capacity are higher than that of the codes reported in this chapter.

To summarize, the main contribution of this chapter is a simple method to design rate-compatible codes, within a wide range of rates, for adaptive coding and HARQ applications. Examples of rate-compatible codes are produced with rates from 0.32 to 0.88. These codes have iterative decoding thresholds within 0.2 dB of capacity and a linear minimum distance growth property [20]. As a by-product of our main results, we also produce nested protograph

codes that do *not* have uniform information size. The rates of these codes range from 1/2 to 9/10 and have thresholds within 0.2 dB of capacity. Although our protograph designs are constrained within a nested structure, the resulting protograph codes still perform close to capacity uniformly and have the lowest iterative decoding thresholds among structured codes with linear minimum distance growth property reported so far in the literature.

## 3.1 Properties of a Good Protograph

### 3.1.1 Low Iterative Decoding Threshold

It is well known that optimized degree distribution of unstructured LDPC code ensemble should contain at least one very high degree variable nodes and a substantial fraction of degree-2 variable nodes [9]. This also applies to protograph structures [20, 17]. Protograph-based iterative decoding threshold can be further improved by including a few degree-1 variable nodes (usually called a pre-coder) and a very high degree variable node that is optionally punctured. This construction appeared in the accumulate-repeat-accumulate code [19]. The role of degree-1 variable nodes was also observed by Richardson in multi-edge type codes [18]. To summarize, the combination of one or more degree-1 variable node, one very high degree variable node and several degree-2 variable nodes are likely to lead to a good protograph.

### 3.1.2 Linear Minimum Distance Growth

Linear minimum distance growth is a desired property of an LDPC code, which can be verified by computing the asymptotic ensemble weight enumerator [20]. It is known that the LDPC codes with variable node degrees greater than or equal to 3 automatically have such a property [9]. However, as seen in the previous subsection, a good protograph usually includes degree-2 and even degree-1 variable nodes. In order to include degree-2 variables within the graph structure, Divsalar, et al. [20] proposed a check node splitting method that allows to build a graph with degree-2 from a graph having variables that have higher degree ($\geq 3$) while still preserving its linear minimum distance growth property. This technique

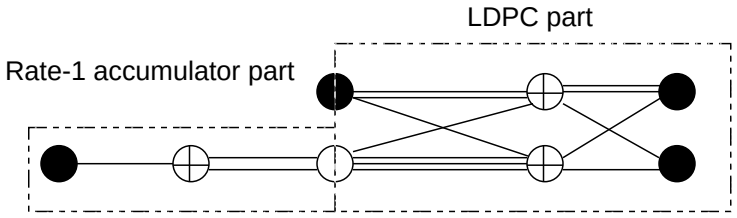


Figure 3.1. The rate 1/2 AR4JA protograph as a concatenation of a pre-coder and a protograph-based LDPC code

requires the maximum number of degree-2 variables in the protograph to be limited by total number of checks minus 1, and also there should be neither parallel edges nor cycles among these degree-2 nodes [29].

### 3.1.3 Protograph Structure with Degree-1 Variable Nodes

Degree-1 variable nodes (accumulators) are pre-coders that improve the iterative decoding threshold. We divide the protograph into two sub-graphs: One sub-graph contains the degree-1 variables, the corresponding check nodes, and any variables connected to these check nodes. The other subgraph contains all variables with degree greater than 1. The latter subgraph is LDPC-like, since good random LDPC codes do not have degree-1 variable nodes. The two subgraphs have in common certain variables of degree $> 1$ that are connected to the check node(s) associated with degree-1 node(s) (see for example Figure 3.1).

Since the degree-1 variables do not affect the growth of minimum distance with codeword length [20], any discussion regarding linear growth of minimum distance applies only to the LDPC subgraph.

As an example, the structure of the rate-1/2 AR4JA code from [20] is shown in Figure 3.1 in which dark circles are transmitted variable nodes, the white circle is un-transmitted (punctured) variable node and plus circles are check nodes. It is a concatenation of a pre-coder (degree-1 part or a rate-1 accumulator) with an LDPC subgraph. The protomatrix of the

rate-1/2 AR4JA protograph is

$$H_{1/2} = \left(\begin{array}{c|cccc} 1 & 2 & 0 & 0 & 0 \\ \hline 0 & 3 & 1 & 1 & 1 \\ 0 & 1 & 2 & 2 & 1 \end{array}\right) \tag{3.1}$$

Using the PEXIT technique, we find this rate-1/2 code has a threshold of 0.6337 dB.

The LDPC protograph subgraph, i.e., the two rows in the right-bottom part of (3.1), consists of only two check nodes. Following the linear minimum distance criteria, there is only one degree-2 variable node allowed in the protograph structure.

Although it is a good code, the AR4JA code has not been designed by a systematic search of the design space and therefore it can be improved upon. In the following, we outline a method for a systematic and efficient search of the design space that finds better codes.

### 3.2 Design Method

Based on the discussion above, our design procedure is:

1. Start with a desired code rate, determine number of check and variable nodes.
2. Impose empirical constraints for good threshold (see Section 3.1.1.)
3. Impose constraints for linear minimum distance growth (see Section 3.1.2.)
4. Among the graphs satisfying the constraints, select the graph with the lowest iterative decoding threshold.

The rate of the code does not uniquely determine the number of check and variable nodes, rather, if the number of check and variable nodes is respectively $n_c$ and $n_v$, assuming $n_p$ punctured variable nodes, $R = \frac{n_v - n_c}{n_v - n_p}$. So for the same rate, we may have larger or smaller protographs. Larger protographs of the same rate may yield better thresholds because of the larger search space and the design flexibility. However, as protographs grow bigger there

is a point of diminishing returns for the optimization of threshold and the search algorithm becomes more complex. Furthermore, if the overall codeword length is constant, a larger protograph means that the repetition factor of the protograph will be smaller, which restricts the design of the interleaver and makes it more difficult to avoid an error floor due to trapping sets.

With all that in mind, consider the following example for the design of a rate-1/2 protograph code. We begin with a simple protograph structure which has 7 variable nodes (of which the third node is punctured) and 4 check nodes. We shall see that this choice of the number of check and variable nodes leads to a threshold that is within a small fraction of a dB to capacity, therefore a larger protograph is not needed.[1] The rate of the proposed protograph is $R = \frac{7-4}{7-1} = \frac{3}{6} = \frac{1}{2}$. We include a punctured variable node since it has been observed that punctured (un-transmitted) variable nodes can improve the performance [18, 44, 20]. We represent the protograph by its $4 \times 7$ protomatrix. This protomatrix contains 28 elements each indicating how many parallel edges connect the respective check node (row) and variable node (column). Optimization over these 28 variables is computationally complex and the finding of the optimum in a high-dimensional space is difficult, therefore adjustments to this optimization problem are necessary to make a practical solution viable.

As discussed earlier, we institute one degree-1 variable (one column of weight-1), and one degree-2 variable node (column of weight-2). Therefore the protomatrix is:

$$H_{1/2}^{search} = \left(\begin{array}{c|cccccc} 1 & 0 & y_1 & y_2 & y_3 & y_4 & y_5 \\ \hline 0 & 1 & x_1 & x_4 & x_7 & x_{10} & x_{13} \\ 0 & 1 & x_2 & x_5 & x_8 & x_{11} & x_{14} \\ 0 & 0 & x_3 & x_6 & x_9 & x_{12} & x_{15} \end{array}\right)_{4\times 7} \tag{3.2}$$

The variables $x_i$ and $y_j$ designate the remainder of the protograph to be designed. The overall code is effectively the concatenation of an LDGM code (first row and column) with an LDPC code (columns 2 through 7, rows 2 through 4). The linear minimum distance

[1] Experiments showed that smaller protographs were not satisfactory.

growth only involves the LDPC part of (3.2), therefore all the comments in the remainder of this paragraph are focused on this submatrix. As shown in [29], to have linear minimum distance growth, the number of degree-2 nodes in an LDPC protograph must be limited to the number of check nodes in the LDPC part minus 1, that is $3-1$. It is also known that degree-2 nodes cannot form a cycle among themselves. However, in our design, we only search for one degree-2 variable node since extensive experiments show that two degree-2 variable nodes will only give an inferior threshold.

Other variable nodes within the LDPC part of the matrix must have degree at least 3, therefore the sum of columns designated in $x_i$, except for up to two degree-2 nodes, must be 3 or higher.

To limit the search space, we constrain the maximum number of parallel edges in the protograph. Via experiments, we observed that increasing the number of parallel edges beyond a certain point is not useful. The reason is that subject to a given code length, increasing the number of parallel edges will increase the likelihood of short cycles. For the remaining nodes, we set the maximum number of parallel edges to 3.

These constraints reduce both the dimensionality of the search space as well as the breadth of the discrete search in each dimension, making a systematic search viable. Thus:

$$H'_{1/2} = \left(\begin{array}{c|cccccc} 1 & 0 & 2 & 0 & 0 & 1 & 0 \\ \hline 0 & 1 & 3 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 2 & 2 & 2 & 1 \\ 0 & 0 & 2 & 0 & 0 & 0 & 2 \end{array}\right)_{4\times 7} \tag{3.3}$$

The threshold of this code is 0.395 dB which shows a gap of 0.208 dB of capacity. The new protograph is shown in Figure 3.2. This code shows improvements over the AR4JA rate-1/2 code reported in [20]. However, it should also be mentioned that the AR4JA protograph has fewer nodes, and within the class of protographs with 5 variable nodes the AR4JA code remains very attractive.

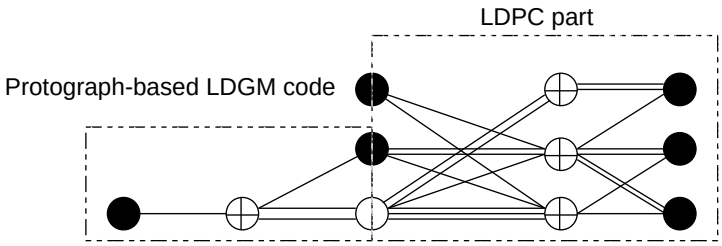


Figure 3.2. The new rate 1/2 protograph with a threshold of 0.395 dB

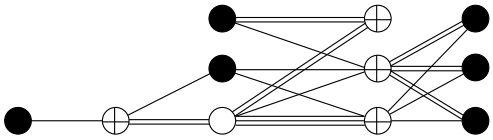

Figure 3.3. The rate-1/2 protograph with a threshold of 0.439 dB

If a degree-2 variable node is excluded, a slightly higher threshold protograph is obtained, having the proto-matrix

$$H_{1/2} = \begin{pmatrix} 1 & 1 & 2 & 0 & 0 & 0 & 0 \\ 0 & 1 & 3 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 2 & 2 & 2 & 1 \\ 0 & 0 & 2 & 0 & 0 & 0 & 2 \end{pmatrix} \tag{3.4}$$

The threshold of this code is 0.439 dB, a gap of 0.252 dB of capacity. This code is originally reported in [47] and plotted in Figure 3.3.

It is instructive to note that if we do not impose the linear minimum distance growth criteria, the optimization procedure yields a protograph with a lower threshold that has a protomatrix as follows

$$H_{1/2}^{opt} = \left(\begin{array}{c|cccccc} 1 & 0 & 2 & 0 & 0 & 1 & 0 \\ \hline 0 & 1 & 2 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 2 & 2 & 2 & 0 \\ \hline 0 & 0 & 3 & 1 & 1 & 1 & 1 \end{array}\right) \tag{3.5}$$

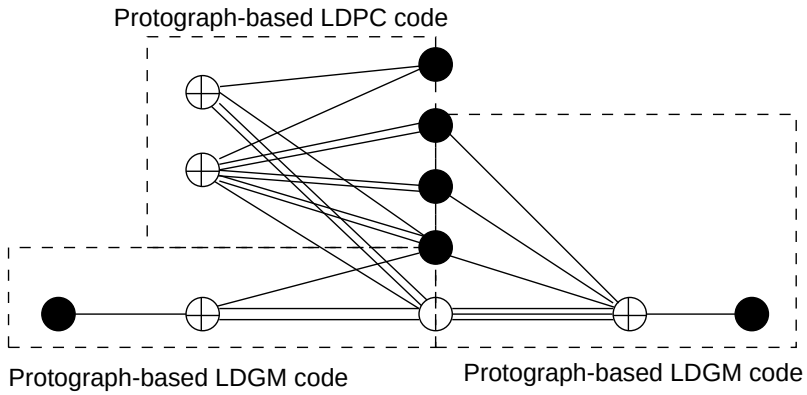


Figure 3.4. The new rate 1/2 protograph with a threshold of 0.250 dB

This code has a threshold of 0.250 with a gap of only 0.063 dB to capacity. This is the best iterative decoding threshold reported for a structured (non-random) code in the literature, but it does not mean it is a good code. In fact, this is the concatenation of two LDGM codes (the first column-row; and last column-row in (3.5)) with one LDPC code (corresponding to the middle sub-matrix of (3.5)) as plotted in Figure 3.4. The LDPC subgraph has too many degree-2 variable nodes (3 degree-2 variables and 2 checks) which violates the condition for linear minimum distance [20, 29]. Therefore the minimum distance of this code does not grow linearly with codeword length and it suffers from an error floor.

### 3.3 Design of High Rate Protograph Codes by Lengthening

So far we have concentrated on rate-1/2 protograph codes. Designing higher rate protographs is more difficult because the protomatrix contains many more elements than those of a rate-1/2 protograph. In this section, we propose a simple approach to overcome this difficulty. We produce codes whose iterative decoding thresholds are close to capacity, form a nested family of codes and offer a range of rates.

We construct a family of high-rate protograph codes from a low-rate protograph by code lengthening, i.e., by lengthening the parity check matrix of a lower-rate code (a base code)

in the form of

$$H_{high\ rate} = [H_1 H_e] \tag{3.6}$$

where $H_1$ is the parity check matrix of the low-rate code and $H_e$ is an extension matrix.

Several families of nested protograph codes based on code lengthening have been reported in the literature [20, 19] using node splitting and permutation of the edges in an ad-hoc manner that is not feasible to generalize to various mother codes and various rates. Therefore, there is motivation in devising a systematic approach for the design of nested protograph codes.

We will describe in the following an example for the design of high-rate codes with rate $R = \frac{n+1}{n+2}$, $n = 1, 2, \ldots$ as an extension of our optimized rate-1/2 protograph of (3.3) (from previous section). These codes have a minimum distance that grows linearly with code length. Similar to the previous section, we describe the search space for our code in the following way

$$H^{search}_{\frac{n+1}{n+2}} = \left(\begin{array}{c|ccc} & y_1 & y_2 & y_3 \\ \hline H'_{\frac{n}{n+1}} & x_1 & x_4 & x_7 \\ & x_2 & x_5 & x_8 \\ & x_3 & x_6 & x_9 \end{array}\right) \tag{3.7}$$

where variables $y_j$ in the first row correspond to the check node that connects to the degree-1 variable node. Variables $x_i$ in rows 2-4 constitute the extension to the LDPC subgraph. In order to preserve the linear growth of minimum distance for the new high rate codes, the column sums in the LDPC subgraph, namely sums of columns designated with variable $x_i$, should be 3 or higher. This constraint ensures that all variables in LDPC part of the extension structures have degree $\geq 3$.

Table 3.1. Threshold of the new protograph code family

| Code Rate | Protograph threshold (dB) | Capacity threshold (dB) | Gap to capacity |
|---|---|---|---|
| 1/2 | 0.395 | 0.187 | 0.208 |
| 2/3 | 1.181 | 1.059 | 0.122 |
| 3/4 | 1.701 | 1.626 | 0.075 |
| 4/5 | 2.112 | 2.040 | 0.062 |
| 5/6 | 2.445 | 2.362 | 0.083 |
| 6/7 | 2.702 | 2.625 | 0.077 |
| 7/8 | 2.930 | 2.845 | 0.085 |
| 8/9 | 3.123 | 3.042 | 0.081 |
| 9/10 | 3.288 | 3.199 | 0.089 |

We further simplify the problem by setting the maximum number of parallel edges to 2 (i.e. $x_i, y_j = 0, 1, 2$). We obtain:

$$
\begin{aligned}
H'_{2/3} &= \left(\begin{array}{c|ccc} H'_{1/2} & 0 & 1 & 1 \\ & 1 & 0 & 1 \\ & 2 & 2 & 2 \\ & 0 & 1 & 0 \end{array}\right) \qquad H'_{3/4} = \left(\begin{array}{c|ccc} H'_{2/3} & 0 & 0 & 1 \\ & 1 & 2 & 0 \\ & 2 & 1 & 2 \\ & 0 & 0 & 2 \end{array}\right) \\
H'_{4/5} &= \left(\begin{array}{c|ccc} H'_{3/4} & 0 & 1 & 2 \\ & 1 & 1 & 2 \\ & 2 & 2 & 1 \\ & 0 & 0 & 0 \end{array}\right) \qquad H'_{5/6} = \left(\begin{array}{c|ccc} H'_{4/5} & 0 & 0 & 0 \\ & 0 & 2 & 2 \\ & 2 & 1 & 1 \\ & 2 & 0 & 0 \end{array}\right) \\
H'_{6/7} &= \left(\begin{array}{c|ccc} H'_{5/6} & 0 & 1 & 2 \\ & 1 & 1 & 2 \\ & 2 & 2 & 2 \\ & 0 & 0 & 0 \end{array}\right) \qquad H'_{7/8} = \left(\begin{array}{c|ccc} H'_{6/7} & 0 & 0 & 1 \\ & 2 & 2 & 0 \\ & 1 & 2 & 1 \\ & 0 & 0 & 2 \end{array}\right) \\
H'_{8/9} &= \left(\begin{array}{c|ccc} H'_{7/8} & 0 & 0 & 1 \\ & 1 & 2 & 1 \\ & 2 & 1 & 2 \\ & 0 & 1 & 0 \end{array}\right) \qquad H'_{9/10} = \left(\begin{array}{c|ccc} H'_{8/9} & 0 & 0 & 2 \\ & 2 & 2 & 0 \\ & 1 & 1 & 2 \\ & 0 & 0 & 2 \end{array}\right)
\end{aligned}
\tag{3.8}
$$

The iterative decoding thresholds of these codes are given in Table 3.1. For rates $> 2/3$, the produced codes have decoding thresholds within 0.09 dB of capacity, and the rate-2/3 code has a threshold within 0.122 dB of capacity. This shows an improvement of 0.2 dB compared with the AR4JA family [20].

## 3.4 Design of Rate-Compatible Protograph Codes

Although we have successfully designed a family of nested codes in Section 3.3, the information block-lengths of the nested codes are not identical. The same is true of AR4JA codes. Thus, these codes are not truly rate-compatible and are unsuitable e.g. for HARQ applications. Several works in the literature design irregular LDPC rate-compatible codes using puncturing and code extension [39, 41, 35, 32, 33] whose code design requires painstaking optimization and furthermore their encoders are unstructured therefore computationally complex. Our design avoids the weaknesses of puncturing by using a code extension approach.

El-Khamy et al. [44] designs rate-compatible protograph LDPC codes by first extending and randomly pruning from an existing protograph to produce a low-rate code, and then obtaining other high-rate codes by puncturing this low-rate code. As mentioned in the beginning of this chapter, puncturing has several weaknesses and usually results in codes that have iterative decoding thresholds with a wider gap to capacity than that of the low-rate mother code. The weaknesses of puncturing can be avoided by using a code extension approach, as discussed below for our designs.

The new rate-compatible family of codes is constructed by extending the parity check matrix $H$ of a high-rate protograph code by an equal number of columns (variable nodes) and rows (check nodes), ensuring that the new code will have the same information block size (Figure 3.5). In this figure, the parity check matrix $H_{1/2}$ of the rate 1/2 code is extended to create $H_{1/3}$, which in turn is extended to obtain $H_{1/4}$. The zero submatrices ensure that

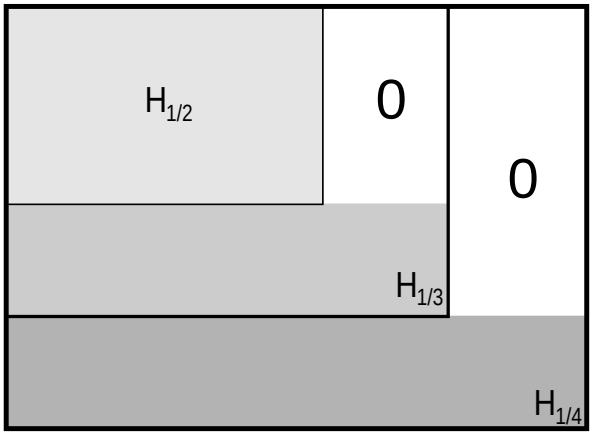


Figure 3.5. Rate-compatible parity check matrices by extension

the incremental variable nodes are indeed new parity bits determined only by the new parity check equations, thus guaranteeing that the high-rate code is nested inside the low-rate code.

In the interest of brevity, we only present one construction of a nested rate-compatible family of codes. However, the proposed method is completely general; we can start from any high-rate code to build a successive set of lower-rate codes. The starting point for the following construction is the rate-5/6 protograph code designed in the last section. Since an equal number of columns and rows are added to the protomatrix, the lower-rate codes obtained from the rate-5/6 code above have rates $R = \frac{19-4}{19-1+n} = \frac{15}{18+n}$, where $n$ is the number of checks and variables added. For each value of $n$, the new code will require a search. As a representative sample, we concentrate on the search for $n = 2$, which yields a rate $R = \frac{15}{20} = 0.75$ code. The search space for this new code is in the form

$$H^1_{0.75} = \left(\begin{array}{cccccccc|cc} & & & & & & & & 0 & 0 \\ & & & & & & & & 0 & 0 \\ & & & H'_{5/6} & & & & & 0 & 0 \\ & & & & & & & & 0 & 0 \\ \hline 0 & x_1 & y_1 & x_3 & . & . & . & x_{17} & 1 & 0 \\ 0 & x_2 & y_2 & x_4 & . & . & . & x_{34} & z & 1 \end{array}\right) \tag{3.9}$$

The problem is to find edges that connect the old graph (the rate-5/6 code) to 2 new check nodes and variable nodes. The unknown edges are represented with variables $x_i$, $y_j$ and $z$, which are divided into two types. The first edge type connects the variables of the old graph to new checks, denoted by $x_i$ and $y_j$, where variables $y_j$ is the number of edges connecting to the highest degree variable node in the old graph, and variables $x_i$ are the remaining edges connecting to the rest of old variables except degree-1 node. The other type consists of edges that connect the new variable nodes to the new check nodes, denoted by variable $z$ in the right-bottom corner sub-matrix. In order to maintain linear growth of minimum distance, more care is needed in this case since degree-2 variables might arise that form short cycles. Therefore we also further simplify the problem by setting this submatrix to be lower triangular.

The optimization process is simplified by further limiting the maximum number of parallel edges. For example, we can set $y_j$ to the values of $1, 2$, $x_i$ and $z_k$ to the values of $0, 1$. The resulting rate-0.75 protograph has a protomatrix given by:

$$H_{0.75} = \left(\begin{array}{cccccccccccccccccc|cc} 1 & 0 & 2 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 3 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 2 & 0 & 1 & 1 & 2 & 0 & 2 & 2 & 0 & 0 \\ 0 & 1 & 1 & 2 & 2 & 2 & 1 & 2 & 2 & 2 & 2 & 1 & 2 & 2 & 2 & 1 & 2 & 1 & 1 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 & 2 & 0 & 1 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \end{array}\right) \tag{3.10}$$

This protograph has an iterative decoding threshold of 1.706 dB with a gap of only 0.078 dB to capacity. Applying this procedure, we are able to generate a family of rate-compatible codes with rates ranging from 0.32 to 0.88, based on the rate-5/6 protograph. To get as many as lower rate codes as possible, we only add one variable node and one check node to the mother code. These codes have rates $R = \frac{15}{19-1+n}$, $n = 1, \ldots, 28$, i.e., rates from $\frac{15}{46} = 0.32$ to $\frac{5}{6} = 0.83$. The rate-0.88 (15/17) protograph is obtained by puncturing the degree-1 variable node in the rate-5/6 protograph structure. The protomatrix of the smallest rate ($R = 15/46$) code in this rate-compatible structure is given in Eq. (3.11), from which all other protomatrices can be deduced. This protograph has 47 variable and 32 check nodes with

Table 3.2. Threshold of rate-compatible protograph code family

| Code Rate | Protograph threshold (dB) | Capacity threshold (dB) | Gap to capacity |
|---|---|---|---|
| 15/17 | 3.137 | 2.945 | 0.192 |
| 5/6 | 2.445 | 2.362 | 0.083 |
| 15/19 | 2.046 | 1.948 | 0.098 |
| 3/4 | 1.706 | 1.628 | 0.078 |
| 15/21 | 1.434 | 1.364 | 0.070 |
| 15/22 | 1.208 | 1.143 | 0.065 |
| 15/23 | 1.027 | 0.968 | 0.059 |
| 15/24 | 0.878 | 0.816 | 0.062 |
| 3/5 | 0.785 | 0.682 | 0.103 |
| 15/26 | 0.706 | 0.559 | 0.147 |
| 5/9 | 0.548 | 0.471 | 0.077 |
| 15/28 | 0.432 | 0.357 | 0.075 |
| 15/29 | 0.383 | 0.265 | 0.118 |
| 1/2 | 0.278 | 0.188 | 0.090 |
| 15/31 | 0.198 | 0.109 | 0.089 |
| 15/32 | 0.132 | 0.050 | 0.082 |
| 15/33 | 0.080 | -0.014 | 0.094 |
| 15/34 | 0.032 | -0.070 | 0.102 |
| 3/7 | 0.002 | -0.118 | 0.116 |
| 15/36 | -0.067 | -0.170 | 0.103 |
| 15/37 | -0.106 | -0.217 | 0.111 |
| 15/38 | -0.144 | -0.256 | 0.112 |
| 15/39 | -0.171 | -0.295 | 0.124 |
| 15/40 | -0.231 | -0.334 | 0.103 |
| 15/41 | -0.259 | -0.375 | 0.116 |
| 15/42 | -0.316 | -0.410 | 0.094 |
| 15/43 | -0.357 | -0.432 | 0.075 |
| 15/44 | -0.382 | -0.469 | 0.087 |
| 1/3 | -0.402 | -0.501 | 0.099 |
| 15/46 | -0.421 | -0.532 | 0.111 |

the highest degree variable node is punctured. The thresholds of the new rate-compatible codes are given in Table 3.2. As seen in the table, these codes have thresholds uniformly within 0.2 dB of their capacity limits. All the codes have linear minimum distance growth property [20].

### 3.5 Simulation Results

In this chapter, a protograph code is derived from our proposed protograph in two lifting steps. First, the protograph was lifted by a factor of 4 using the progressive edge growth (PEG) algorithm [48] in order to remove all parallel edges. Then based on desired information block-length, the resulting graph was lifted again using PEG to determine a circulant permutation of each edge class in order to maximize the girth of an overall bipartite graph.

In our nested codes of Section 3.3, the parity check matrix for the lower-rate code can be obtained by removing certain columns from the parity check matrix of the higher rate code, and this produces economies in the design of the decoders. In fact, it is enough to design a decoder for the largest rate code (9/10). To decode the lower-rate codes, the missing coded bits are replaced by erasure at the decoder. In the same manner, the rate-compatible codes generated in Section 3.4 only need the decoder of the lowest rate. Other higher-rate codes are decoded by replacing missing parity bits by erasure at the decoder. The fixed-point iterative sum-product decoder used for protograph codes is described in the appendix.

The performances of our nested high-rate codes with rates 1/2, 2/3, 3/4, 5/6, 7/8 and 9/10 over the binary-input AWGN channel are shown in Figure 3.6. All codes are simulated with the information block-length of 16k. The nested high-rate protographs given in Section 3.3 are first lifted by a factor of 4. This results in larger protographs that are lifted again with factors of 1365, 683, 455, 273, 195 and 152, respectively. For the rate-compatible protographs given in Section 3.4, we only need to lift the largest protograph (rate-15/46), by factor of 4 and then 273. Other high rate codes are obtained by removing the redundant parity bits. The performances of rate-compatible codes are shown in Figures 3.7 and 3.8.

$$H_{32\times 47} =$$

$$\begin{pmatrix}
1&0&2&0&0&1&0&0&1&1&0&0&1&0&1&2&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&1&3&1&1&1&0&1&0&1&1&2&0&1&1&2&0&2&2&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&1&1&2&2&2&1&2&2&2&2&1&2&2&2&1&2&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&2&0&0&0&2&0&1&0&0&0&2&0&0&0&2&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&1&0&0&0&1&0&0&0&0&1&1&1&1&1&1&1&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&2&0&0&0&0&0&0&1&0&1&0&0&1&0&0&1&1&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&2&1&0&0&0&0&0&1&0&1&0&0&0&1&0&1&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&2&0&0&0&0&0&0&1&0&1&0&0&1&0&0&0&0&1&1&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&2&0&0&1&0&0&0&1&0&0&0&0&1&0&0&1&0&0&0&1&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&2&0&0&0&0&0&0&1&0&0&0&1&0&0&0&0&0&0&0&0&1&1&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&1&1&0&0&0&0&0&0&1&0&1&0&0&1&1&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&1&0&0&0&0&1&0&1&0&0&0&0&1&1&0&1&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&2&0&0&0&1&0&0&1&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&2&0&0&0&0&0&0&0&0&0&0&1&0&0&0&1&0&0&0&0&1&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&2&0&1&0&0&0&0&1&0&1&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&2&0&0&0&0&0&0&0&0&1&0&0&0&0&1&0&0&1&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&2&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&1&0&0&1&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&2&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&1&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&2&0&0&0&0&0&0&0&1&0&0&0&0&0&1&0&0&0&0&0&1&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&2&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&0&0&0&1&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&2&0&1&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0\\
0&0&1&0&0&0&0&0&0&0&1&1&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&2&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&1&0&1&0&1&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&2&0&1&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&2&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&2&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&1&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0\\
0&0&2&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0\\
0&0&1&0&1&0&1&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0\\
0&0&1&0&1&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0\\
0&0&2&0&1&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0\\
0&0&2&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0
\end{pmatrix}$$

(3.11)

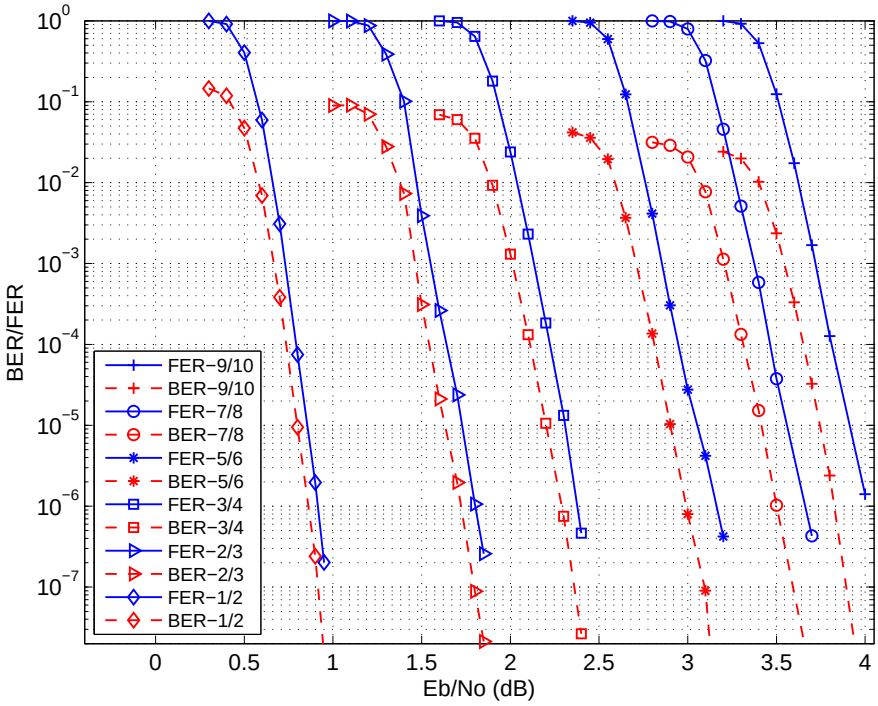


Figure 3.6. Performance of nested codes of Section 3.3.

The resulting protograph codes have the girth $\geq 8$ except the protograph codes with rates 7/8, 9/10 and 15/46 where the girth is 6. The equivalent parity check matrices of all protograph codes have full rank except the rate-1/2 protograph plotted in Figure 3.2 whose the parity check matrix has a row null space of dimension one. Thus, the rate loss is negligible. The maximum number of iterations is set to 200. No error floors were observed up to FER of $10^{-6}$. The practical performance at $FER = 10^{-6}$ shows a gap of less than 0.6 dB to threshold and 0.8 dB to capacity, thus the waterfall regions are steep.

Figure 3.9 shows a comparison between our rate-1/2 code (protomatrix deduced from Equation (3.11)) and the rate-1/2 AR4JA code reported in CCSDS standard [49]. The new code outperforms the AR4JA code by approximately 0.2 dB. A comparison is also shown in the same figure with the rate-1/2 LDPC code reported in the DVB-S2 standard for video broadcasting [50]. The standard provides the code structure as a concatenation of BCH and LDPC codes, and reports the code performance with the information block-length of 32k. Even with a smaller block-length, the new code outperforms the standard code.

## 3.6 Conclusion

This chapter presents a simple approach for constructing rate-compatible LDPC codes based on protographs which perform within 0.2 dB of their capacity limits. The rate-compatible structure has the advantage of low encoding complexity and efficient decoding algorithm with one hardware implementation for all members of the code family.

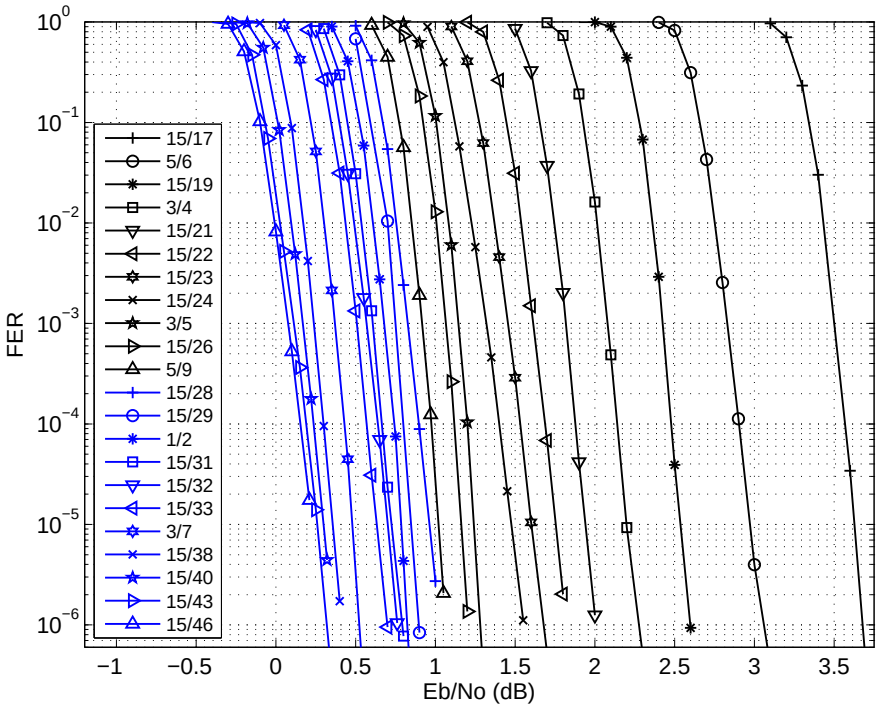


Figure 3.7. FER performance of rate-compatible codes of Section 3.4.

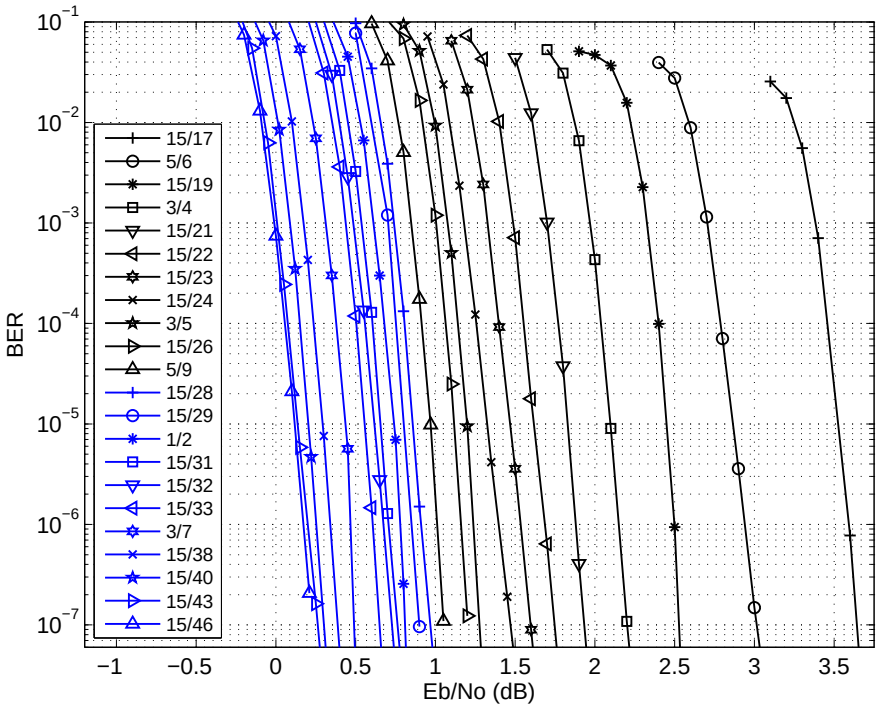


Figure 3.8. BER performance of rate-compatible codes of Section 3.4.

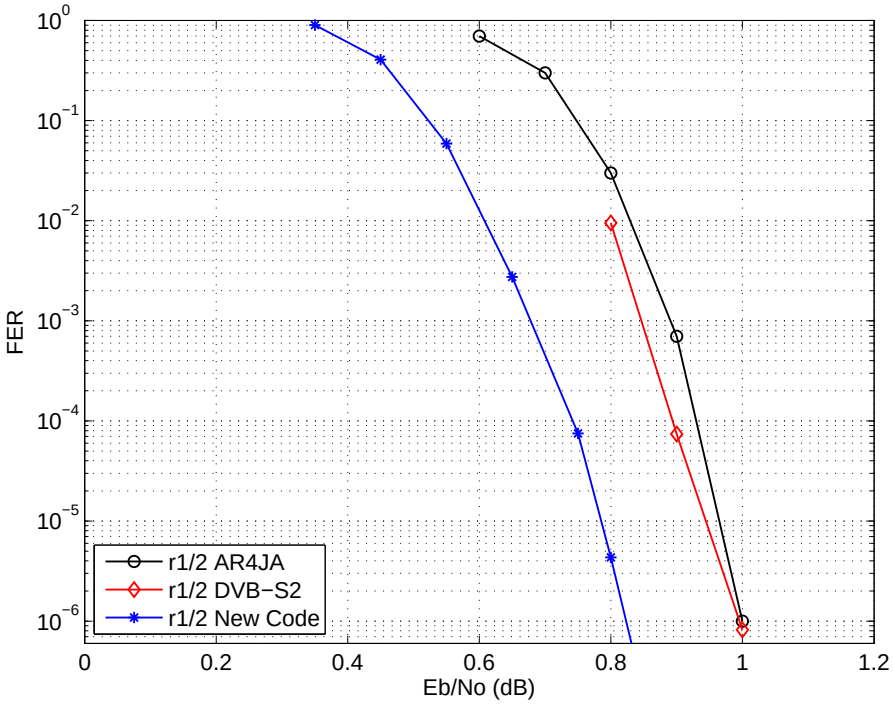


Figure 3.9. Comparison of the rate-1/2 code produced in Section 3.4 and AR4JA, both with information block-length of $16k$, and DVB-S2 code with information block-length of $32k$.

# CHAPTER 4
# BILAYER PROTOGRAPH CODES FOR HALF-DUPLEX RELAY CHANNELS

To harvest the cooperative gain promised by information theoretic results [51, 52], we are in need of coding schemes that can approach the fundamental limits of relay channels. Among the early coding results for the relay channel were distributed coded diversity schemes [53, 54, 55], which proposed convolutional and Turbo codes for the fading channel under cooperation. Later on, Duman et al. [56, 57] proposed turbo codes for the decode-forward relay channel in half-duplex and full-duplex modes. There is a large body of work focused on designing LDPC codes for the relay channel [58, 59, 60, 61, 62, 63, 64]. These LDPC code designs mainly depend on irregular LDPC codes and use density evolution (or related) techniques to search for optimized irregular LDPC ensembles operating at two different rates. Razaghi and Yu [61] produce bilayer LDPC code structures for the decode-and-forward (DF) scheme. This approach has been refined in other works including [58, 59]. Other examples of LDPC codes for the half-duplex relay channel include [60, 62, 63, 64].

It has been known that the basic problem of coding for the decode-forward relay channel in the power-limited regime can be reduced to the following: a code and its sub-code must be designed simultaneously that operate with two different rates at two different SNRs at the relay and the destination respectively. Despite significant progress, existing LDPC codes must be painstakingly optimized to match to a set of channel conditions, many of them do not offer easy encoding, and lack a structure that facilitates the optimization of design. This chapter proposes a class of LDPC relay codes that address three important issues in an integrative manner: low encoding complexity, modular structure allowing for easy design, and rate-compatibility so that the code can be easily matched to a variety of channel

conditions without extensive re-optimization. In addition, the proposed codes offer excellent performance.

In this chapter, we concentrate on the decode-and-forward protocol. For practical purposes, we further limit ourselves to the half-duplex relaying where the relay cannot transmit and receive at the same time. The main contribution of this chapter is a coding scheme for the relay channel using protographs [17] that are built with bilayer expurgated and bilayer-lengthened structures [61]. A protograph code [17] is an LDPC code that can be constructed by extension from a small bipartite graph called a protograph, whose graph topology is represented by a so-called proto-matrix. Protograph codes can achieve very good thresholds with low encoder complexity as well as fast decoding [26, 20]. The proposed approach offers flexibility in designing a family of rate-compatible embedded codes for relay channels. These embedded codes allow a coding scheme whose rate can easily be adapted to channel conditions, and are thus suitable for designing multi-relay coding systems. We also introduce a methodology to evaluate the end-to-end error performance of relay coding schemes, and demonstrate the end-to-end performance of our proposed codes.

## 4.1 Background

### 4.1.1 System Model

A half-duplex single-relay channel is shown in Figure 4.1. For the moment we ignore the coding scheme shown in this figure to introduce the system model. $X_i$ and $W_i$ denote the transmitted signals from the source (S) and the relay (R) while $Y_i$ and $V_i$ denote the received signals at the destination (D) and relay respectively. Subscript $i = 1$ denotes the broadcast mode which is active for a fraction $t$ of the transmission interval, and subscript $i = 2$ denotes the multiple access mode which is active for a fraction $t' = 1 - t$ of the transmission interval. The received signals are:

$$V_1 = h_{SR}X_1 + N_{R_1}$$

$$Y_1 = h_{SD}X_1 + N_{D_1}$$

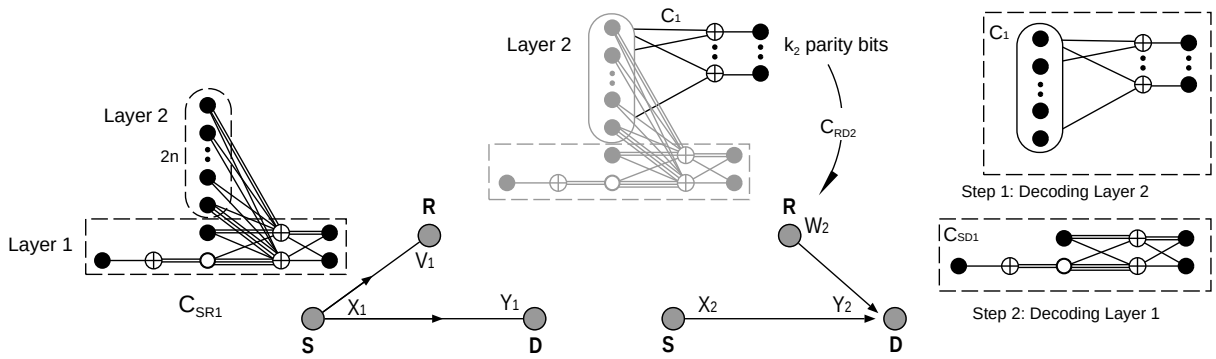


Figure 4.1. Bilayer-lengthened protograph design for half-duplex relay with $R_{SD_1} = \frac{1}{2}$ and $R_{SR_1} = \frac{n+1}{n+2}$, $n = 1, 2, \ldots$

$$Y_2 = h_{SD}X_2 + h_{RD}W_2 + N_{D_2} \tag{4.1}$$

where $h_{SD}$ and $h_{RD}$ are the S-to-D and R-to-D channel coefficients respectively, $N_{R_1}$, $N_{D_1}$ and $N_{D_2}$ are the noise samples at the relay and the destination in the first and second time slots, respectively. All noise samples are assumed to be Gaussian with zero mean and unit variance. $P_{S_1} = E[X_1^2]$ represents the source transmission power in the first time slot (duration $t$). Similarly $P_{S_2}$ and $P_{R_2}$ represent the source and relay transmission powers in the relayed time slot (duration $t' = 1 - t$). We also define $SNR_{SR}$, $SNR_{SD}$ and $SNR_{RD}$ as signal-to-noise ratios by the relay and destination during the first time slot, and by the destination during the second time slot, respectively.

### 4.1.2 Coding for Half-Duplex Relay Channels

In the first time slot, a source sends a high-rate $C_{SR_1}$ code to a relay and a destination, using a rate that is decodable at the relay. In the second time slot, the relay transmits additional mutual information to help the destination decode, producing an effectively lower-rate $C_{SD_1}$.[1] Thus the relay coding problem consists of a simultaneous design of two codes

[1]In our formulation the source may also be active in the second time slot, but it will be constrained to synchronously transmit the same signal/codebook as the relay. Thus in this work the distinction between orthogonal and non-orthogonal transmission is only in the

that operate at two different SNRs $SNR_{SR} > SNR_{SD}$, such that one is a subset of the other. This problem has been attempted a number of times by forceful optimization, but a more streamlined approach is now available via bilayer expurgated and bilayer-lengthened LDPC structures [61]. In the bilayer LDPC structure, either the bit nodes or the check nodes are divided into two sets (called layers) and the graph, although still a bipartite graph, is re-drawn with two rows of check nodes sandwiching the bit nodes, or vice versa [61, Figure 3 and 6]. This provides a convenient way to build and illustrate subcodes, which we use for our purposes. For a more comprehensive treatment and justification of the bilayer structure please see [61, 63].

## 4.2 Design of Bilayer Protograph Codes for Relay Channels

### 4.2.1 Bilayer Lengthened Structure

A bilayer lengthened structure for the relay channel [61] is shown in Figure 4.1 where a high-rate code is constructed by adding variable nodes to the graph of a low-rate code. Geometrically, the overall graph contains two layers (sets) of variable nodes and one set of check nodes. One layer of variables connecting to the checks forms a graph of a lower-rate code, and the entire graph constitutes a high-rate code. The corresponding parity check matrix has the form:

$$H_{SR_1} = [H_{SD_1} H_e] \tag{4.2}$$

where $H_{SD_1}$ is the parity matrix of the lower-rate code which is equivalent to the sub-code (Layer 1 or $C_{SD_1}$) and $H_e$ is the extension matrix representing Layer 2. Layers 1 and 2 combined create a capacity-approaching code for the source-relay link ($C_{SR_1}$) as shown in Figure 4.1. The relay decodes this codeword, then protects the codeword bits in Layer 2 via $k_2$ parity bits, and the parities are transmitted to the destination with $C_{RD_2}$ at the second time slot. The destination, using these $k_2$ parity bits, can reliably detect the bits in Layer

received SNR during relay time slots. Since we parameterize our analysis by received SNR without any direct reference to source and relay powers, both orthogonal and (synchronous) non-orthogonal transmissions are subsumed in the following developments.

2, therefore they can be eliminated from the code graph. The remaining graph contains just Layer 1, which constitutes a capacity approaching code at the SNR of the SD link, which is decodable at the destination.

We will describe in the following an example for the design of the bilayer lengthened codes with rates $R = \frac{n+1}{n+2}$, $n = 1, 2\ldots$, as an extension of the rate-1/2 protograph of Eq. (3.4). For each new code, three variable nodes (or columns) are added, as follows:

$$H_{\frac{n+1}{n+2}} = \left(\begin{array}{c|ccc} & y_1 & y_2 & y_3 \\ \hline H_{\frac{n}{n+1}} & x_1 & x_4 & x_7 \\ & x_2 & x_5 & x_8 \\ & x_3 & x_6 & x_9 \end{array}\right) \tag{4.3}$$

where $x_i$ and $y_i$ variables denote the number of parallel edges in the extension graph, to be determined. Variables $y_i$ in the first row correspond to the check node that connects to the degree-1 variable node. Variables $x_i$ in rows 2-4 correspond to the remaining check nodes. In order to preserve the linear growth of minimum distance for the bilayer codes, the column sums in rows 2-4, namely sums of columns designated with variables $x_i$, should be 3 or higher [29].

We further simplify the problem by setting the maximum number of parallel edges to 2 (i.e. $x_i, y_i = 0, 1, 2$). These constraints limit the range of parameters thus simplifying the optimization. Our cost function is the threshold which is calculated via the PEXIT method. For this example, the best threshold is given by:

$$H_{2/3} = \left(\begin{array}{c|ccc} & 0 & 1 & 1 \\ H_{1/2} & 1 & 1 & 1 \\ & 2 & 1 & 2 \\ & 0 & 1 & 0 \end{array}\right) \qquad H_{3/4} = \left(\begin{array}{c|ccc} & 0 & 0 & 2 \\ H_{2/3} & 2 & 2 & 0 \\ & 1 & 1 & 2 \\ & 0 & 0 & 1 \end{array}\right)$$

$$H_{4/5} = \left(\begin{array}{c|ccc} H_{3/4} & 0 & 1 & 2 \\ & 1 & 2 & 2 \\ & 2 & 1 & 1 \\ & 0 & 0 & 0 \end{array}\right) \qquad H_{5/6} = \left(\begin{array}{c|ccc} H_{4/5} & 0 & 0 & 1 \\ & 2 & 2 & 0 \\ & 1 & 1 & 2 \\ & 0 & 0 & 2 \end{array}\right)$$

$$H_{6/7} = \left(\begin{array}{c|ccc} H_{5/6} & 0 & 0 & 1 \\ & 1 & 2 & 1 \\ & 2 & 1 & 2 \\ & 0 & 1 & 0 \end{array}\right) \qquad H_{7/8} = \left(\begin{array}{c|ccc} H_{6/5} & 0 & 0 & 2 \\ & 2 & 2 & 0 \\ & 1 & 1 & 2 \\ & 0 & 0 & 2 \end{array}\right)$$

$$H_{8/9} = \left(\begin{array}{c|ccc} H_{7/8} & 0 & 0 & 0 \\ & 0 & 1 & 2 \\ & 2 & 2 & 1 \\ & 1 & 1 & 0 \end{array}\right) \qquad H_{9/10} = \left(\begin{array}{c|ccc} H_{8/9} & 0 & 0 & 2 \\ & 1 & 2 & 0 \\ & 2 & 1 & 2 \\ & 0 & 0 & 2 \end{array}\right)$$

The iterative decoding thresholds of these codes calculated by PEXIT technique [28] are given in Table 4.1. For rates $> 2/3$, the produced codes have iterative decoding thresholds within 0.1 dB of the capacity, and the rate-2/3 code has a threshold within 0.152 dB.

### 4.2.2 Design of Bilayer Expurgated Protograph Codes

A bilayer expurgated structure for the relay channel [61] is shown in Figure 4.2 where a low-rate code is constructed by adding check nodes to the graph of a high-rate code. The bilayer graph in this case contains two layers (sets) of check nodes and one set of variable nodes (unlike the bilayer lengthened structure). The variable nodes together with one layer of check nodes form the graph of a high-rate code, and the entire graph constitutes a low-rate code. The corresponding parity check matrix has the form

$$H_{SD_1} = \begin{bmatrix} H_{SR_1} \\ H_e \end{bmatrix} \tag{4.4}$$

where $H_{SR_1}$ is the parity check matrix of a high-rate LDPC capacity-approaching code for the source-relay link (representing the sub-code $C_{SR_1}$ of Layer 1), and $H_e$ is the extension matrix

Table 4.1. Thresholds of proposed protograph codes

| Code Rate | Protograph threshold (dB) | Capacity threshold (dB) | Gap to capacity |
|---|---|---|---|
| Bilayer lengthened design | | | |
| 1/2 | 0.439 | 0.187 | 0.252 |
| 2/3 | 1.223 | 1.059 | 0.164 |
| 3/4 | 1.720 | 1.626 | 0.094 |
| 4/5 | 2.136 | 2.040 | 0.096 |
| 5/6 | 2.455 | 2.362 | 0.093 |
| 6/7 | 2.718 | 2.625 | 0.093 |
| 7/8 | 2.941 | 2.845 | 0.099 |
| 8/9 | 3.125 | 3.042 | 0.083 |
| 9/10 | 3.295 | 3.199 | 0.096 |
| Bilayer expurgated design | | | |
| 3/4 | 1.720 | 1.626 | 0.094 |
| 2/3 | 1.182 | 1.059 | 0.123 |
| 7/12 | 0.809 | 0.590 | 0.219 |
| 1/2 | 0.420 | 0.187 | 0.233 |
| 5/12 | 0.144 | -0.185 | 0.329 |
| 1/3 | -0.263 | -0.497 | 0.234 |

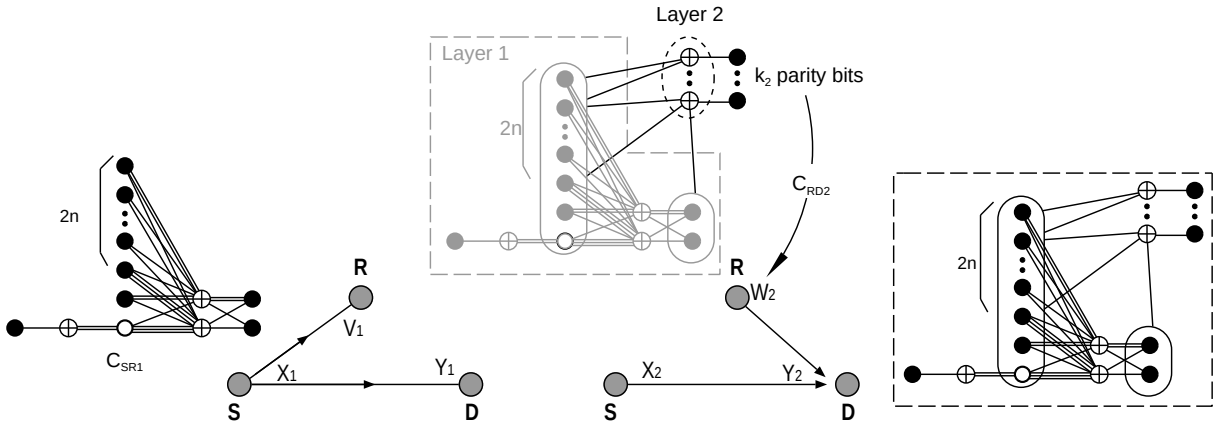


Figure 4.2. Bilayer expurgated protograph design for half-duplex relay with $R_{SD_1} = \frac{N-k_1-k_2}{N}$ and $R_{SR_1} = \frac{N-k_1}{N}$, $N$ is the number of variable nodes

representing Layer 2. Layers 1 and 2 together create a capacity-approaching code $C_{SD_1}$ for the source-destination link as shown in Figure 4.2. The source transmits a $C_{SR_1}$ codeword. The relay, after decoding the source codeword, produces $k_2$ additional "parity" bits using the extension matrix $H_e$, re-encodes these $k_2$ bits with codebook $C_{RD_2}$ and transmits to the destination. At the destination, the $k_2$ parity bits that are reliably detected essentially provide $k_2$ additional check values on the source codeword from $C_{SR_1}$. This is equivalent to decoding a $C_{SD_1}$ codeword (with a lower rate) at the SNR of the source-destination link.

We now describe an example for the design of bilayer expurgated codes. In the interest of brevity, we only present one construction of bilayer codes. However, the proposed method is completely general; we can start from any high-rate code to build a family of expurgated codes. The starting point for the following construction is the rate-3/4 protograph code designed in the last subsection. The rate-3/4 protograph contains 13 variable nodes (one of them punctured) and 4 check nodes. Low-rate bilayer codes expurgated from this code have rates $R = \frac{13-4-n}{13-1}$, where $n$ is the number of checks added. For each value of $n$, the new code will require a search. As a representative sample, we concentrate on the search for $n = 1$, which yields a rate-2/3 code. The search space for this new code is in the form

$$H_{2/3}^{search} = \left(\begin{array}{c} H_{3/4} \\ \hline 0 \quad y \quad x_1 \quad x_2 \quad . \quad . \quad . \quad x_{11} \end{array}\right) \tag{4.5}$$

where $y$ variable denotes the number of parallel edges connecting to the punctured variable node and $x_i$ variables denote the number of parallel edges connecting to remaining variable nodes except the degree-1 variable node. Having a degree-1 variable node improves the iterative decoding threshold of the code [20].

The optimization process is simplified by further limiting the maximum number of parallel edges. For example, we can limit $y \in \{1, 2\}$ and $x_i \in \{0, 1, 2\}$. Proto-matrices of new expurgated protographs are given by

$$H_{2/3}^{e} = \left(\begin{array}{c} H_{3/4} \\ \hline 0 \quad 1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 1 \quad 0 \quad 0 \quad 1 \quad 2 \end{array}\right)$$

$$H_{7/12}^{e} = \left(\begin{array}{c} H_{2/3}^{e} \\ \hline 0 \quad 1 \quad 0 \quad 0 \quad 1 \quad 0 \quad 0 \quad 1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 2 \end{array}\right)$$

$$H_{1/2}^{e} = \left(\begin{array}{c} H_{7/12}^{e} \\ \hline 0 \quad 2 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 0 \quad 1 \quad 0 \quad 0 \end{array}\right)$$

$$H^e_{5/12} = \left(\begin{array}{cccccccccccc} & & & & & & H^e_{1/2} & & & & & \\ \hline 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \end{array}\right)$$

$$H^e_{1/3} = \left(\begin{array}{cccccccccccc} & & & & & & H^e_{7/12} & & & & & \\ \hline 0 & 2 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right)$$

The iterative decoding thresholds of these codes are given in Table 4.1. The produced codes all have decoding thresholds within 0.23 dB of the capacity except the rate-7/12 one which has a decoding threshold within 0.33 dB of the capacity.

## 4.3 Numerical Results

### 4.3.1 Coding Thresholds

We design nested bilayer lengthened and expurgated protograph codes for the half-duplex relay channel, which can operate at any two rates with iterative decoding thresholds close to capacity, as shown in Table 4.1. For comparison, the thresholds of bilayer lengthened and bilayer expurgated *irregular* LDPC codes proposed in the literature are shown in Table 4.2. All codes use a rate pair ($R_{SD_1} \simeq 0.5, R_{SR_1} \simeq 0.7$). Our proposed bilayer lengthened and bilayer expurgated codes are better than the codes proposed in [64] and within 0.1 dB of codes reported in other works[2]. Thus, a simple design approach has yielded a family of codes that offer a performance comparable with other highly-optimized bilayer LDPC codes, while

[2]The rate of [61, 64] is slightly less than 1/2, thus slightly skewing the comparison in their favor.

Table 4.2. Comparison of thresholds ($R_1 \simeq 0.5, R_2 \simeq 0.7$)

| | Lengthened | | | | Expurgated | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | [61] | [64] | [58] | Our code | [61] | [59] | [58] | [64] | Our code |
| $\text{Gap}_1$ | 0.164 | 0.3854 | 0.1039 | 0.239 | 0.514 | 0.258 | 0.284 | 0.6323 | 0.233 |
| $\text{Gap}_2$ | 0.120 | 0.1758 | 0.0945 | 0.152 | 0.084 | 0.084 | 0.084 | 0.215 | 0.123 |

offering important advantages in terms of rate-compatibility, low encoding complexity, and the fast decoding properties of protograph codes.

### 4.3.2 Simulation Results

Our protograph codes are derived from protographs in two lifting steps. First, the protograph is lifted by a factor of 4 using the progressive edge growth (PEG) algorithm [48] in order to remove all parallel edges. Then, a second lifting using the PEG algorithm was performed to determine a circulant permutation of each edge class that would yield the desired code block length.

In our nested bilayer lengthened codes of Section 4.2.1, the parity check matrix for the lower-rate code can be obtained by removing certain columns from the parity check matrix of the higher rate code, and this produces economies in the design of the decoders. In fact, it is enough to design a decoder for the largest rate code (9/10). To decode the lower-rate codes, the missing coded bits are replaced by erasures at the decoder. In the same manner, the bilayer expurgated codes generated in Section 4.2.2 only need the decoder of the lowest rates (1/3). Other higher-rate codes are decoded by ignoring redundant checks at the common decoder.

The performances of our bilayer lengthened protograph codes with rates 1/2, 2/3 and 3/4 over a binary-input additive white Gaussian noise (BI-AWGN) are shown in Figure 4.3. In addition, performances of bilayer expurgated protographs with rates 1/3, 5/12, 1/2, 2/3 and 3/4 are shown in Figure 4.4. All codes are simulated with the information block-length of 16k. To construct codes with this 16k information block-length, the protographs for the lengthened codes are first lifted by a factor of 4, and subsequently again by factors 1365,

683, and 455, for codes of rate 1/2, 2/3, and 3/4 respectively. A similar two-step lifting is used for the expurgated codes.

The decoder is a standard message passing decoder where the maximum number of iterations is set to 200. LLR clipping and other decoding parameters are according to [65]. No error floors were observed up to WER of $10^{-6}$. The gap to capacity of these codes are within 0.7 dB of their iterative decoding threshold and within 1 dB of their capacity limits at $WER \approx 3 * 10^{-6}$.

We now consider a half-duplex relay channel with information block-length of 16380 bits. We assume the SR link supports rate $R_{SR_1} = 3/4$ and the SD link supports rate $R_{SD_1} = 1/2$. For this example, the time division $t = 0.75$ is chosen for the source and relay. The rate needed in the RD link is $R_{RD_2} = 3/4$ [63, Eq. 5]. This is the same as the rate of SR code but with a different codeword-length. Under these conditions, the achievable rate of the relay channel is 0.5625 [63, Eq. 3].

We now consider the bilayer expurgated coding scheme (Figure 4.2) with information blocklength 16380. $C_{SR_1}$ is implemented by the rate-3/4 protograph given in Section 4.2.1. The code $C_{RD_2}$, which protects additional parities generated at the relay, has information blocklength 5460. $C_{RD_2}$ is constructed from the same protograph as $C_{SR_1}$, but with a shorter length. The destination decodes a $C_{SR_1}$ codeword plus these additional parity values, which constitutes a codeword of $C_{SD_1}$ in the same manner as [61]. The WER and BER performances of these codes are shown in Figure 4.5. The shorter block length of the rate-3/4 RD code has resulted in a wider waterfall region, highlighting a weakness of the successive decoding approach at smaller block lengths.

The bilayer lengthened structure involves four codes (Figure 4.1). Using the above relay channel parameters, $C_{SR_1}$ has the information block-length 16380 and codeword block length 21840. Each codeword of $C_{SR_1}$ is composed of a $C_{SD_1}$ codeword with blocklength 10920 and an extension of 10920 bits derived via the extension matrix $H_e$ from Eq. (4.2). The relay, after decoding and recovering the codeword, multiplies the extension bits by the parity check matrix of a rate-1/2 code $C_1$ to calculate a syndrome of length 5460. $C_1$ is lifted from the

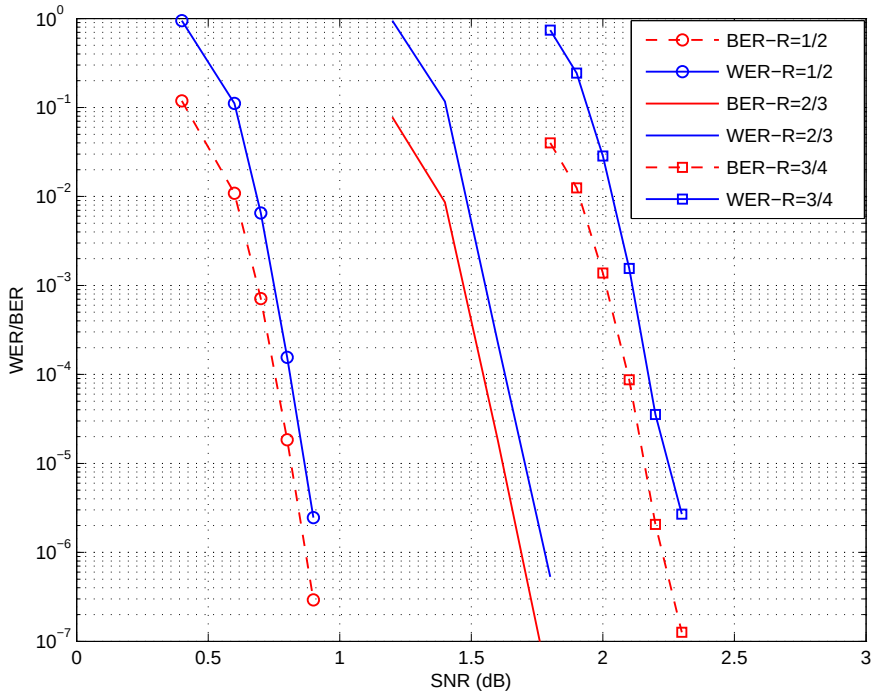


Figure 4.3. Performance of rate 1/2, 2/3, 3/4 bilayer lengthened protograph codes designed in Section 4.2.1 with information blocklength of 16k

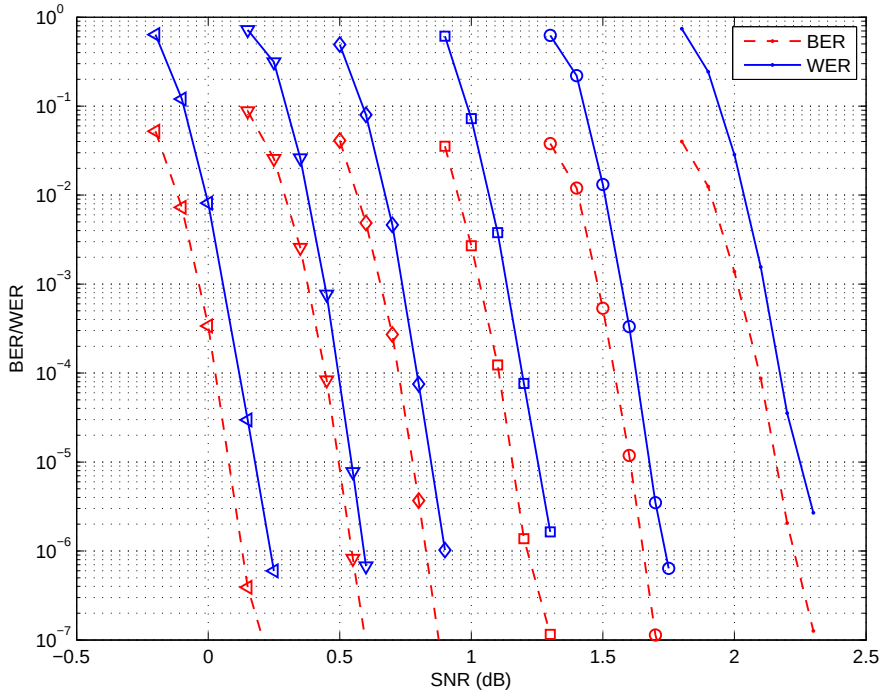


Figure 4.4. Performance of rate 1/3, 5/12, 1/2, 7/12, 2/3, 3/4 bilayer expurgated protograph codes designed in Section 4.2.2 with information block-length of 16k

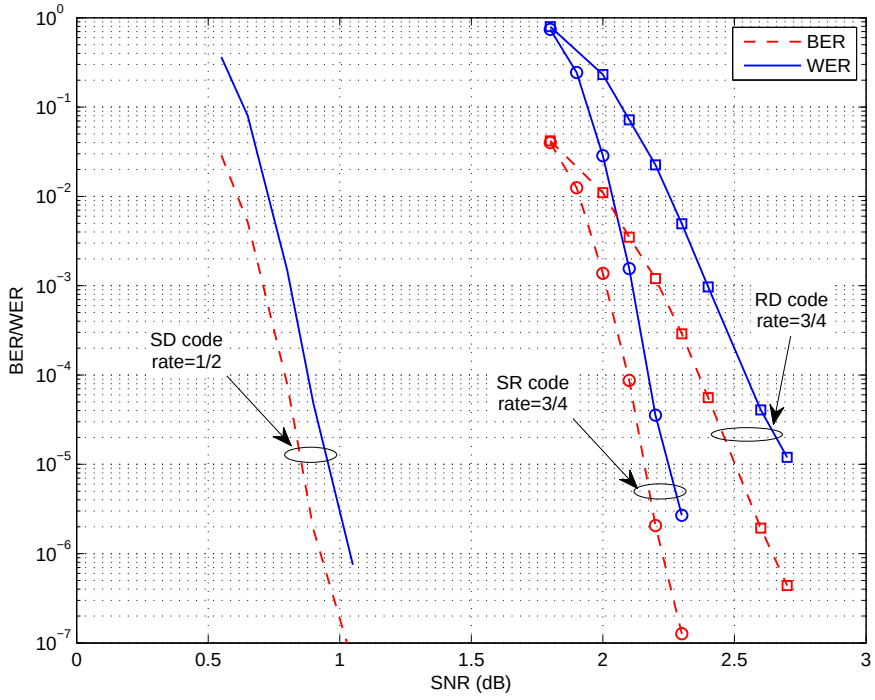


Figure 4.5. Performance of component relay codes: $R_{SD_1}$=1/2, $R_{SR_1}$=3/4 and $R_{RD_2} = 3/4$ using bilayer expurgated structure in Section 4.2.2 with $\alpha = 1.4$ dB and $\beta = 1.6$ dB

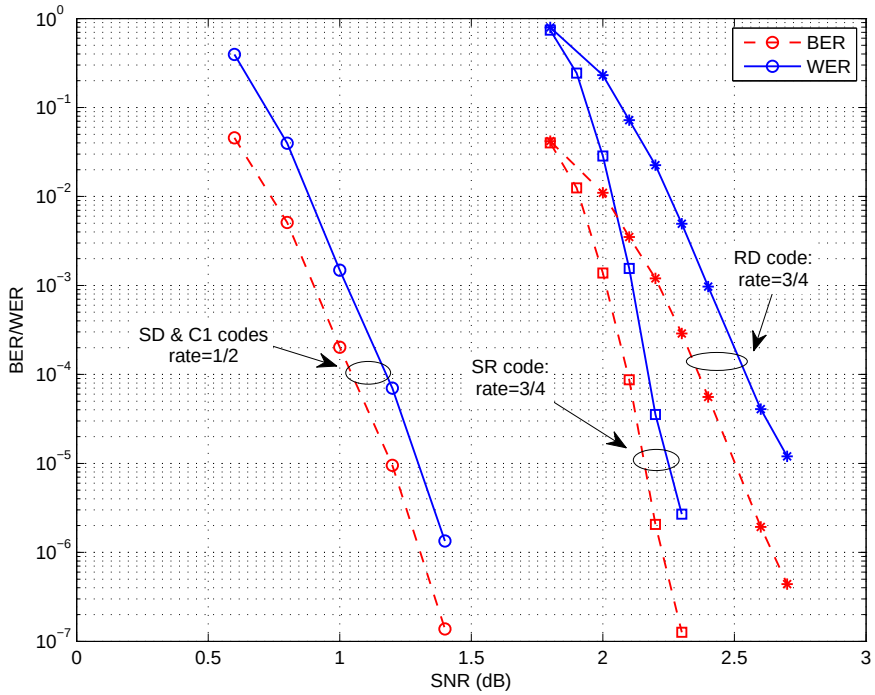


Figure 4.6. Performance of component codes for a relay channel: $R_{SD_1}$=1/2, $R_{SR_1}$=3/4, $R_{C_1} = 1/2$ and $R_{RD_2} = 3/4$ using bilayer lengthened structure in Section 4.2.1 with $\alpha = 1.4$ dB and $\beta = 1.6$ dB

rate-1/2 protograph given in Eq. (3.4). The syndrome is then encoded by $C_{RD_2}$, transmitted and decoded at the destination to recover the syndrome. $C_{SR_1}$ and $C_{RD_2}$ are similar to their counterparts in the bilayer expurgated scheme. Also, $C_{SD_1}$ and $C_1$ have the same code structure and blocklength. The WER and BER performances are shown in Figure 4.6. Performance of $C_{SD_1}$ in the bilayer-lengthening relay structure is worse than in the bilayer expurgated structure because of its shorter blocklength.

Although the example above was coached in terms of orthogonal transmissions, it also applies to a non-orthogonal system with correspondingly lower source and relay powers (see footnote 1).

### 4.3.3 End-to-End Performance of Relay Coding Systems

Many works in contemporary literature represent the performance of relay channel codes by illustrating two component codes: the source-relay code, and the source-destination code. The price of this simple approach is that one must either ignore or assume perfect the relay-destination code. We believe there is merit in providing a comprehensive end-to-end performance because the relay-destination code operates at a smaller block length, thus it has a wider waterfall and may very well be the bottle neck for the entire system, therefore providing the thresholds or even simulations for only the two other sub-codes may not always be fully illuminating of the overall performance.

A complete representation of the error performance of a coding scheme for a relay channel would show error as a function of the three SNRs of its constituent channels. But this requires a four-dimensional plot which has limited practical utility. We construct a simpler error plot by assuming

$$SNR_{SR} = SNR_{SD} + \alpha \qquad SNR_{RD} = SNR_{SD} + \beta \tag{4.6}$$

where $\alpha$ and $\beta$ are fixed constants. The affine relationship of the SNRs allows them to be displayed on the same axis, producing a simple end-to-end error plot which can be thought of as a 2-dimensional slice of the general four-dimensional error characterization of the relay channel. $\alpha$ and $\beta$ can be carefully chosen (as shown below) to generate a reasonable and insightful overall representation of the performance of the relay channel that does not ignore or trivialize any of the parameters of the system.

To simplify the simulations, we develop a tight bound on the end-to-end error based on component errors in the system. For the purposes of exposition we concentrate on the expurgated-bilayer codes (Figure 4.2), where three types of errors may happen in the system: $E_R$ is the error event at the relay, $E_{RD}$ is the error in the decoding of the "extra parity" arriving from the relay to the destination, and finally, $E_D$ is the error event at the destination in the final decoding of the source message, and the complement of event $E$ is shown with

$\overline{E}$. The bound is shown in Eq. (4.9).

$$P_e = P(E_D|\overline{E_R})P(\overline{E_R}) + P(E_D|E_R)P(E_R) \tag{4.7}$$

$$= \left[P(E_D|\overline{E_{RD}}, \overline{E_R})P(\overline{E_{RD}}) + P(E_D|E_{RD}, \overline{E_R})P(E_{RD})\right]P(\overline{E_R}) + P(E_D|E_R)P(E_R) \tag{4.8}$$

$$\leq P(E_D|\overline{E_{RD}}, \overline{E_R}) + P(E_{RD}) + P(E_R) \tag{4.9}$$

It is easy to see the bound is tight because component codes are used in a regime where their errors are, conservatively, no more than $10^{-2} \sim 10^{-3}$, therefore with a very good approximation $P(\overline{E_R}) \approx 1$, $P(\overline{E_{RD}}) \approx 1$, $P(E_D|E_R) \approx 1$ and $P(E_D|E_{RD}, \overline{E}_R) \approx 1$.

We are now ready to calculate the values of $\alpha$ and $\beta$, which will determine how the individual code error characteristics are combined to produce the end-to-end performance. We are interested in cases where none of the component errors dominate the others, because when one of the link errors dominates, the performance will be essentially determined by the code on the dominating link, which is already known from the literature on point-to-point channels. So the interesting case is when no link errors dominate, i.e. the waterfall regions of the component curves coincide at their starting points (approximately the threshold).

The performance of two bilayer codes with a rate pair ($R_{SD} = 1/2, R_{SR} = 3/4$), $\alpha = 1.4$ dB, $\beta = 1.6$ dB and time sharing $t = 0.75$ are shown in Figure 4.7. To understand how far is the end-to-end performance of our codes from theoretical limits, we follow the convention of point-to-point channels and produce a value of SNR at which a random code of infinite length and rate similar to the code under study is capable of theoretically supporting error-free communication. Similarly to the point-to-point case, this involves inverting the capacity formula by inserting the rate and extracting the corresponding SNR. To produce a single SNR value, we use Eq. (4.6), assume that each of the component code is at the rate the mutual information the link supports, and use the relay mutual information formulas in [63]. Using these parameters, the relay channel of the above example has a throughput of 0.5625 and the code rates correspond to the limiting SNR of $SNR_{SD} = 0.225$ dB. As seen in this figure, the gap-to-capacity of bilayer expurgated and bilayer lengthened relay coding schemes

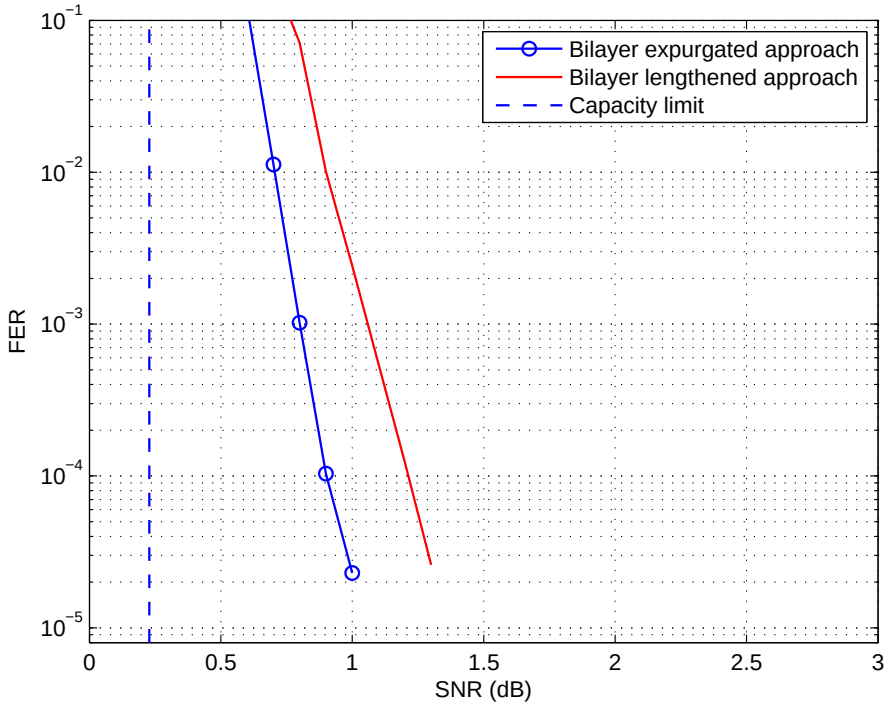


Figure 4.7. End-to-end performance bounds for relay coding schemes in Figure 4.6 and Figure 4.5 with $\alpha = 1.4$ dB and $\beta = 1.6$ dB

are about 0.7 dB and 1.2 dB respectively at FER of $2.10^{-5}$. The end-to-end error bound of the bilayer lengthened code is worse than that of the bilayer expurgated code because of the block-length issues. This phenomenon has also been reported by several previous works including [61, 64].

## 4.4 Design of Bilayer Codes for Two-Relay Channels

In a practical two-relay scenario (Figure 4.8) it is likely that the relays do not have precisely and deterministically identical channels to the source, destination, and each other. Therefore one relay is likely to be "stronger" and decode first, then this relay will be able to help another relay to decode, and then the two relays together will assist the destination.

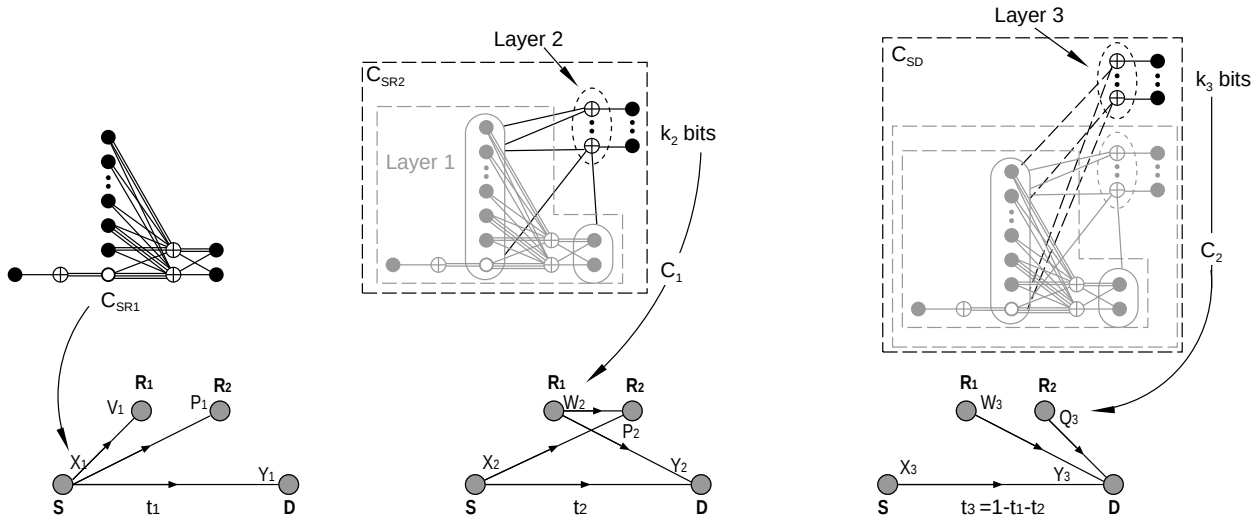


Figure 4.8. The expurgated coding structure for the half-duplex two-relay channel

We consider transmissions in three time slots for the source and two relays. The transmitted signals from the source (S) and two relays ($R_1$ and $R_2$) are denoted with $X$, $W$ and $Q$, respectively, and the received signals at the destination and two relays are denoted with $Y_i$, $V_i$ and $P_i$, respectively, where $i$ indicates the time slot index. In the first time slot, the source encodes its message using the code $C_{SR_1}$. The first relay decodes the source message, but the second relay and the destination cannot (yet). During the second time slot, the first relay generates $k_2$ parity bits using the sub-graph denoted Layer 2,[3] then encodes these parity bits using another LDPC code $C_1$. The second relay, after decoding the $k_2$ parity bits, decodes the source message. Then, the second relay computes $k_3$ parity bits using the subgraph denoted Layer 3, encodes the $k_3$ bits with another LDPC code $C_2$, and transmits to destination. The destination decodes $C_1$, $C_2$, and finally the source message with the help of $k_2 + k_3$ additional parity bits from two relays. The achievable rate using the above strategy is a special case of the achievable rate in [66].

For demonstration, consider an example where $R_1$, $R_2$ and $D$ can reliably receive source signals with a rate $R_{SR_1} = 3/4$, $R_{SR_2} = 7/12$ and $R_{SD} = 1/3$ respectively. The $R_1$-to-$D$

[3]Like the one-relay case, these bits are the syndrome of a parity check extension matrix.

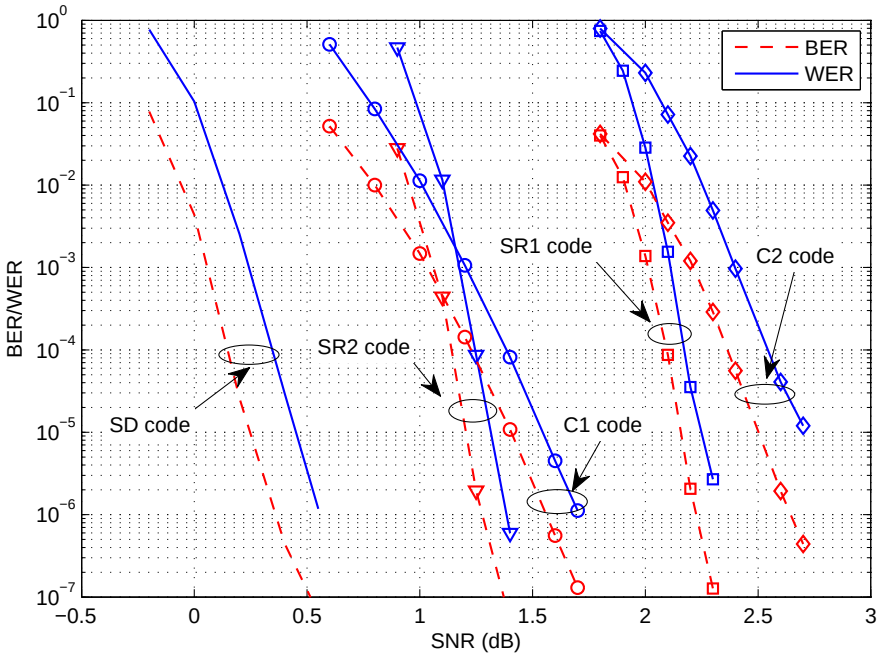


Figure 4.9. Performance of component codes used in a two-relay channel: $R_{SR_1} = 3/4$, $R_{SR_2} = 7/12$, $R_{SD} = 1/3$, $R_{C_1} = 1/2$ and $R_{C_2} = 3/4$. $C_{SR_1}$, $C_{SR_2}$ and $C_{SD}$ all have codeword blocklengths of 23660, while $C_1$ and $C_2$ have codeword blocklengths of 7280.

channel supports $R_{R_1D} = 1/2$. Assuming $t_1 = 0.6$, $t_2 = 0.2$ and $t_3 = 0.2$, the achievable rate of the two-relay channel is 0.45 [63]. The rate of $C_1$ and $C_2$ are 1/2 and 3/4, respectively. Information block length is 16380. $C_{SR_1}$, $C_{SR_2}$ and $C_{SD}$ are constructed from protographs given in Section 4.2.2. $C_{SR2}$ is rate-7/12 and $C_{SD}$ is rate-1/3, resulting in $k_2 = 3640$ and $k_3 = 5460$. Although the explanation of this example was in terms of orthogonal transmissions, the precise same example also applies to a non-orthogonal (beamforming) relay channel with correspondingly lower source and relay power.

$C_1$ arises from the protograph in Eq. (3.4) and $C_2$ is the same code as $C_{SR_1}$, but with a shorter blocklength. Figure 4.9 shows the performance of $C_{SR_1}$, $C_{SR_2}$, $C_{SD}$, $C_1$ and $C_2$, which operate within 0.6 dB, 0.7 dB, 0.9 dB, 1.5 dB and 1 dB respectively of their capacity limits at $WER = 10^{-5}$. As expected, $C_1$ is the bottleneck in this two-relay channel because of its short blocklength. We note that this gap is an outcome of a relatively short overall

block length and a successive decoding approach, both of which are practical considerations and can be relaxed.

**Remark 4.4.1** *In the multi-relay scenario, there are multiple options for the decoding and transmission at relays. For example, two relays can cooperatively beamform towards a destination. The required codes in this case are fundamentally similar to the one-relay channel, therefore we do not consider them separately in this chapter.*

**Remark 4.4.2** *In the multi-relay scenario the transmission time is divided multiple times which shortens the component block-lengths. Therefore we did not pursue a multi-relay generalization of the bilayer* lengthened *structure due to its susceptibility to short block length effects.*

**Remark 4.4.3** *Optimization of the time division to maximize the achievable rates has been pursued in chapters on the capacity of relay channel [67] and can be extended to the multi-relay scenario. Depending on the channel gains, the optimal length for one of the time slots may be zero, in which case one of the relays must be shut off.*

**Remark 4.4.4** *The developments throughout this chapter apply to either orthogonal relaying or to non-orthogonal relaying where concurrent transmissions use the same codebook.*

## 4.5 Discussion and Conclusion

This chapter presents a simple approach for constructing relay coding schemes based on bilayer lengthened and bilayer expurgated protograph codes which perform within a fraction of dB of the capacity. The proposed codes allow easy design, flexibility in matching to various relay channel conditions and low encoding complexity and can be extended to multi-relay networks. A framework for end-to-end performance evaluation of the relay codes is also provided.

Nested protograph codes have also been considered for the point-to-point channel [30], whose similarity with the present work is designing multi-component protograph codes. However, in relay channels the two component codes are sent by different transmitters, unlike the point-to-point case, and see different channels with different SNRs. To address the issues arising from the relay network configuration, a bilayer structure [61] for the overall code and a coset code at the relay was used, which have no counterpart in the point-to-point case [30].

# CHAPTER 5
# PROTOGRAPH-BASED CODED BICM

Bit-interleaved coded modulation (BICM) [68] is a technique that allows relatively simple design of bandwidth-efficient coded modulation systems. This joint design has proven to be powerful and capacity-approaching for fading channels with applications of low-density parity-check (LDPC) codes [69, 70, 71, 72]. However, many of the previous works design a particular code with a specific modulation scheme. It is desirable to find a general scheme that can support multiple rates and multiple modulation schemes within the context of BICM.

In this chapter, we address this question via a family of protograph-based LDPC codes to be mentioned shortly, and combining them with a simple and efficient mapping to high level modulation schemes i.e. QPSK, 8PSK and 16QAM in Rayleigh fading channels. We proceed by calculating their iterative decoding thresholds, a useful parameter when designing any capacity-approaching LDPC code. In the context of *Rayleigh fading*, this chapter for the first time presents the calculation of the threshold of protograph-based BICM.

In this chapter, we study two families of protograph codes: AR4JA protograph codes [20] described in Chapter 2 and our proposed protograph codes reported in Chaper 3, and map them to high-order modulations. We should mention that the protograph-based LDPC coded BICM using AR4JA protograph codes with 16QAM in AWGN channel was previously studied in [73, 74]; the advantage of our design process is that it is modular and easy to apply to other protographs and modulations. We propose here a general mapping scheme that maps any protograph structure with any modulation. The effectiveness of our method is demonstrated via the calculation of the iterative decoding thresholds in Rayleigh faded channels. We show that certain family of protograph-based coded modulation schemes under

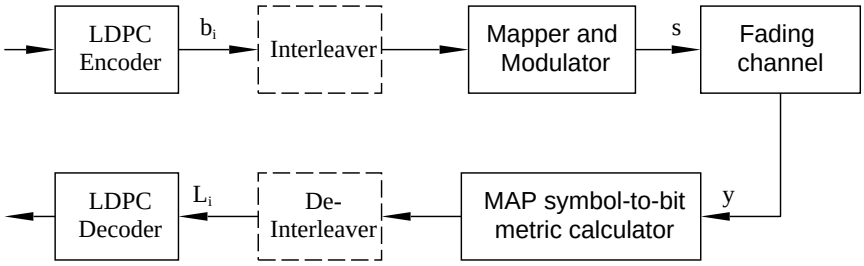


Figure 5.1. Block diagram of BICM system

studies in this chapter are operating within a gap of $0.2 - 0.4$ dB or less to their capacity limits.

## 5.1 System Model

A general BICM system is shown in Figure 5.1. The information sequence is encoded by an LDPC encoder to get a coded bit sequence. The coded sequence is then bit-interleaved before being sent to a modulator. The $M$-ary modulator maps $m = \log_2 M$ coded bits at a time to a complex symbol chosen from an $M$-ary constellation $\chi$. The discrete-time baseband channel model can be written as

$$y_t = hs_t + w_t \tag{5.1}$$

where $t$ is the discrete time index, $y_t$ is the received signal, $s_t$ is the transmitted symbol, $h = h_I + ih_Q$ is the zero mean complex Gaussian distributed fading coefficient with the variance of $1/2$ in each dimension and $w_t$ is a complex white Gaussian noise sample with the zero mean and the variance per dimension of $N_0/2$. In this chapter, we assume that the channel state information is available at the receiver.

The MAP symbol-to-bit metric calculator will compute the symbol-to-bit metrics based on the received symbol $y$ from the channel. These bit metrics are passed to a decoder, which employs the iterative belief propagation algorithm. Let $\chi_b^i$ be the subset of all the signal points $x \in \chi$ whole label has value $b \in \{0, 1\}$ in position $i$. The symbol-to-bit metric

computed by the MAP calculator at each time to feed into the iterative decoder is given by

$$L_i = L(b_i|y) = \ln \frac{\sum_{x \in \chi_0^i} \exp(-\frac{1}{2\sigma^2}\|y-x\|^2)}{\sum_{x \in \chi_1^i} \exp(-\frac{1}{2\sigma^2}\|y-x\|^2)} \tag{5.2}$$

for $i = 1, \ldots, m$. The BICM capacity [68], with perfect CSI and uniform inputs, is given by

$$C = m - \sum_{i=1}^{m} E_{b,y,h} \left[ \log_2 \frac{\sum_{x \in \chi} p_h(y|x)}{\sum_{x \in \chi_b^i} p_h(y|x)} \right] \tag{5.3}$$

In fact, BICM capacity is the sum of mutual information of $m$ parallel channels $I(L_i, b_i)$.

$$C = \sum_{i=1}^{m} I(L_i, b_i) \tag{5.4}$$

This mutual information $I(L, X)$ can be computed by using Monte Carlo simulation [70] as follows

$$I(L, X) = 1 - E[\log_2(1 + \exp(-xL))]) \tag{5.5}$$

$$= 1 - \frac{1}{N} \sum_{n=1}^{N} \log_2(1 + \exp(-x_n L_n) \tag{5.6}$$

## 5.2 Protograph-Based Coded BICM

### 5.2.1 Mapping Algorithm

As seen in the previous chapters, protograph codes can be represented by a small graph with only a few variable nodes and check nodes. For the purposes of BICM, the binary variable nodes of the code must be mapped to the bit-levels of the modulation. The iterative decoding threshold of the overall coded modulation scheme naturally depends in part on the protograph code, but also on the mapping between the code and the modulation levels.

Divsalar and Jones [73] proposed a mapping algorithm based on Variable Degree Matched Mapping (VDMM). The idea is to directly assign protograph variable nodes in proportion to mutual information of coded bits in the modulation symbol. This is a well-known water-filling problem. Following that idea, Jin et. al [74] carried out an exhaustive search of all possible

permutations of protograph variable nodes mappings to symbol bits, and find the optimal mapping pattern for the 16-QAM modulation whose constellation is shown in Figure 5.2, which gives an iterative decoding threshold improvement of 0.14 dB compared with that of the water-filling approach in [73].

In both [73] and [74] variable nodes in a protograph are directly assigned to the bits of a modulation symbol which is shown in Figure 5.3. This limits the coded modulation that can be designed with these methods. In this chapter, we propose a general method that can work for any protograph structure and modulation level and achieving BICM-liked performance. The basic idea of our technique is to use a two-stage lifting approach. Assuming that the desired modulation is $M$-ary and the original protograph has $n$ variable nodes, we start with a smaller lifting of the original protograph by a factor of $m = \log_2 M$ to a slightly larger protograph, using the PEG algorithm [48]. An example is shown in Figure 5.4, where we see an intermediate protograph constructed from four original AR4JA protographs. We now map this intermediate protograph to $n = 4$ modulation symbols as follows: each labeling bit position from all the $n$ modulation symbols will be mapped to the variable nodes of one of the planes in the intermediate protograph. For example, four $b_0$ bits from four modulation symbols are mapped to the top plane, four $b_1$ bits are mapped to the next plane, etc.

Now, the mapping between coding and modulation is complete within a relatively small "intermediate" protograph. The advantage of this intermediate protograph is that it is small enough to allow optimizations, but it also has enough degrees of freedom to provide a good mapping. Now this intermediate protograph is lifted (via a circulant matrix) to the expected codeword length to form a protograph-based LDPC code.

We would like to contrast the proposed method with VDMM in terms of flexibility. In VDMM as it is proposed, the coded bits in each protograph correspond to the symbol bits in one transmitted symbol. Thus, for example, a 4-variable-node protograph naturally corresponds to 16-QAM. With the proposed framework, however, it is very easy to use any protograph together with any modulation. All that is needed is to produce the right intermediate protograph.

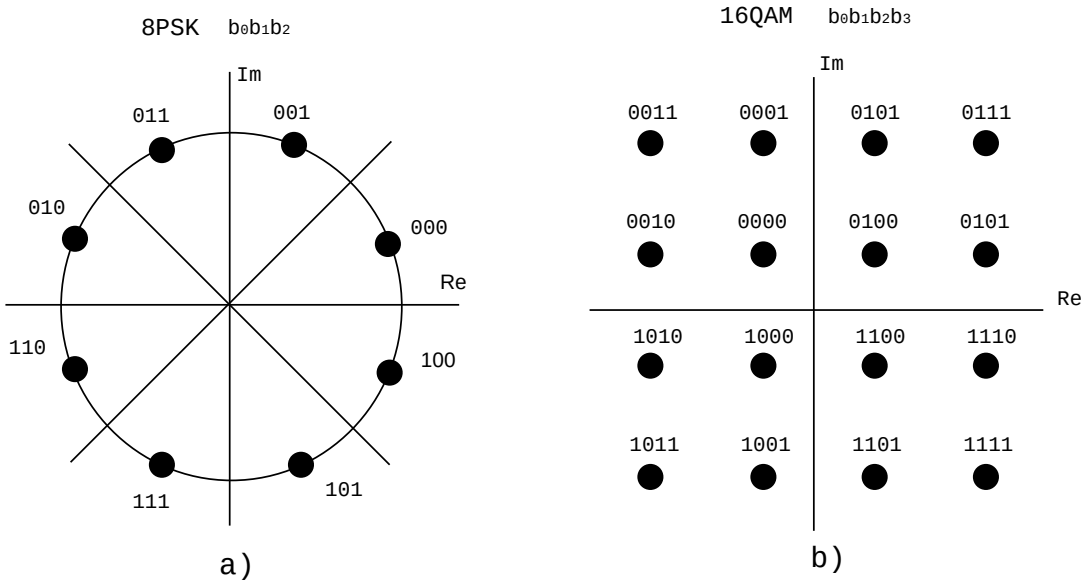


Figure 5.2. 16QAM and 8PSK with Gray labeling

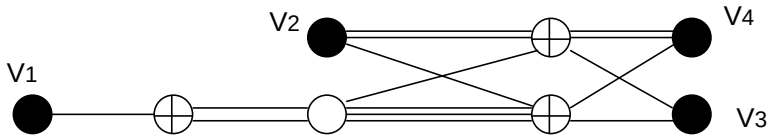


Figure 5.3. The rate-1/2 AR4JA protograph with 16QAM with mapping $\{b_0, b_1, b_2, b_3\} = \{V_2, V_4, V_1, V_3\}$ in Divsalar's paper and $\{b_0, b_1, b_2, b_3\} = \{V_1, V_2, V_3, V_4\}$

Table 5.1 compares the iterative decoding thresholds of our method compared with [73] and [74]. By necessity this table is small, as [73, 74], as mentioned above, naturally correspond to only one type of modulation.[1] The iterative decoding threshold of our mapping method is slightly larger than the brute-force optimized version of [74] and slightly smaller than that of [73]. But we note that the point of the proposed method is not to produce the absolutely smallest threshold for the specific case of $16 - QAM$, but rather to expand the horizon of available designs. In the sequel, we shall see the threshold gap to capacity for our method is very good over a large spectrum of rates and modulations.

[1]The iterative decoding thresholds given in Table 5.1 are calculated with simulations that use longer codewords than those reported in [74], and thus are slightly more accurate.

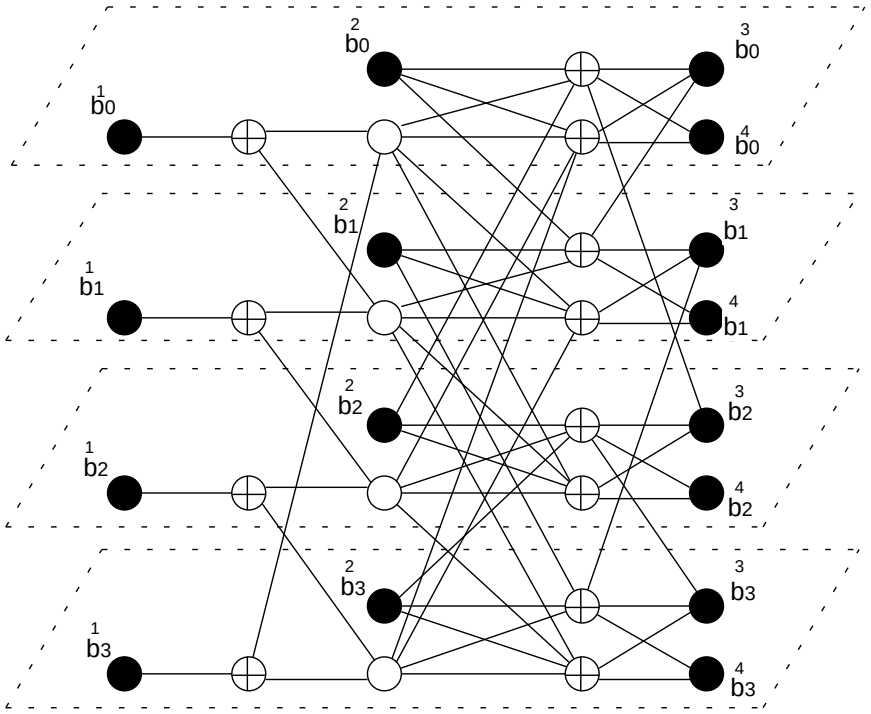


Figure 5.4. The rate-1/2 AR4JA protograph after lifting a factor of 4 with 16QAM

Table 5.1. Mapping of rate-1/2 AR4JA to 16QAM under AWGN

| Method | [73] | [74] | Our approach | capacity threshold |
|---|---|---|---|---|
| Threshold | 2.853 | 2.714 | 2.812 | 2.286 |

Table 5.2. Thresholds of AR4JA coded family in Rayleigh faded channels

| Code Rate | QPSK(dB) | | | 8PSK(dB) | | | 16QAM(dB) | | |
|---|---|---|---|---|---|---|---|---|---|
| | Thd. | cap. | gap | thd. | cap. | gap | thd. | cap. | gap |
| 1/2 | 2.326 | 1.852 | 0.474 | 4.019 | 3.388 | 0.631 | 4.783 | 4.129 | 0.654 |
| 2/3 | 4.242 | 3.660 | 0.582 | 6.118 | 5.598 | 0.520 | 6.992 | 6.219 | 0.773 |
| 3/4 | 5.458 | 4.944 | 0.514 | 7.546 | 6.972 | 0.574 | 8.401 | 7.665 | 0.736 |
| 4/5 | 6.598 | 5.966 | 0.632 | 8.666 | 7.985 | 0.681 | 9.364 | 8.679 | 0.685 |

Table 5.3. Thresholds of 395 coded family in Rayleigh faded channels

| Code Rate | QPSK(dB) | | | 8PSK(dB) | | | 16QAM(dB) | | |
|---|---|---|---|---|---|---|---|---|---|
| | Thd. | cap. | gap | thd. | cap. | gap | thd. | cap. | gap |
| 1/2 | 2.158 | 1.852 | 0.306 | 3.722 | 3.388 | 0.334 | 4.556 | 4.129 | 0.427 |
| 2/3 | 3.916 | 3.660 | 0.256 | 5.728 | 5.598 | 0.130 | 6.508 | 6.219 | 0.289 |
| 3/4 | 5.146 | 4.944 | 0.202 | 7.100 | 6.972 | 0.128 | 7.879 | 7.665 | 0.214 |
| 4/5 | 6.102 | 5.966 | 0.136 | 8.154 | 7.985 | 0.169 | 8.882 | 8.679 | 0.203 |

### 5.2.2 Iterative Decoding Thresholds of Protograph Coded BICM

The PEXIT algorithm to compute iterative decoding thresholds for protograph coded modulation is similar to the one proposed in [28] for AWGN channel, except the initialization step. Instead using exact equation as in initialization step in [28][Section III.C], we need to compute the mutual information for coded bits using Monte Carlo method as given in (5.6).

The iterative decoding thresholds of our protograph-based LDPC coded BICM schemes are given in Table 5.2 and 5.3. From Table 5.2, AR4JA protograph-based coded BICM can operate within about 0.7 dB to the BICM capacity limits. In the other hand, the family of protograph codes shown in Eq. (3.3) and (3.8) can operate within about $0.2 - 0.4$ dB to the BICM capacity limit, which shows an improvement of about 0.3 dB compared with that of the AR4JA family. This observation matches with the results reported in [30] which provided the coding thresholds in AWGN channel.

## 5.3 Numerical Results

In this section, we will compute the iterative decoding thresholds of AR4JA protographs given in (2.3) and our proposed nested protographs lengthened from the rate-1/2 protograph with an iterative decoding threshold of 0.395 dB given in (3.3) and (3.8) with three modulation schemes, i.e. QPSK, 8PSK and 16QAM.

Our protograph codes are built from the protographs in two lifting steps. First, the protograph is lifted by a small factor in order to accommodate all modulation schemes. In the case of three above modulation schemes, we need to lift the protograph by a factor of 12

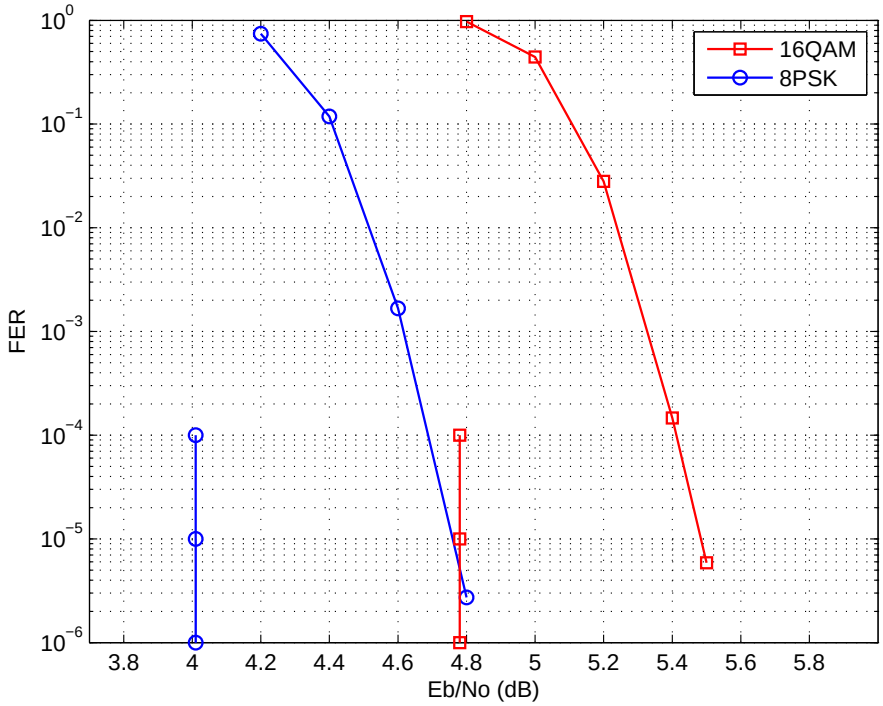


Figure 5.5. Rate-1/2 rate-1/2 AR4JA code using 8PSK/16QAM over Rayleigh fading.

which is a common divisor of $2, 3$ and $4$ bits. This first lifting procedure is implemented by using the progressive edge growth (PEG) algorithm [48] in order to remove all parallel edges. Secondly, the intermediate protograph is lifted using circulants to an expected codeword length depending on different applications [20].

In order to support multiple rates, we begin with the highest-rate $(4/5)$ protograph. Since the protographs have a nested structure, the parity-check matrix of lower rate can be obtained by removing certain columns from codes of higher rates. To decode the lower-rate codewords, the missing coded bits are replaced by erasure at the decoder. Thus the family of protograph codes can be implemented within a common encoder/decoder structure and at the same time being able to support multiple modulation schemes as well.

The performance of the rate-1/2 AR4JA protograph code transmitted with 8PSK/16QAM in the Rayleigh channel using the proposed mapping method is plotted in Figure 5.5 with

the information block-length of 16k. At FER = $10^{-5}$, the AR4JA-BICM scheme perform within a gap of $< 1$ dB to BICM capacity. Thus, the slope of the BICM coding scheme is steep.

## 5.4 Conclusion

This chapter presents a simple method for designing protograph-based BICM that is general and applies to any modulation. The iterative decoding thresholds of the protograph codes while mapped to higher order modulations in Rayleigh faded channels are calculated. This general coding framework can support not only multiple rates but also adaptive modulation. We report that certain families of protograph codes achieve a threshold within a gap of approximately $0.2 - 0.4$ dB of BICM capacity limit across a wide range of rates and modulations.

# CHAPTER 6
# RATE-COMPATIBLE PROTOGRAPH CODES FOR INTER-SYMBOL INTERFERENCE CHANNELS

Rate-compatiblity is a desirable feature that allows a common encoder/decoder. This gives a system flexibility to match channel variations. While the rate-compatible structure has been studies extensively for memoryless AWGN channels, to our knowledge its application for channels with memory, especially inter-symbol-interference (ISI) channels has been open until now.

The capacity of ISI channels with finite-alphabet inputs is still open, however, if the source is restricted to be independent and uniformly distributed (i.u.d.), the capacity of binary-input ISI channels, also called partial response channels is known as the i.u.d. capacity [75]. Irregular LDPC codes have been designed to approach the i.u.d. capacity of partial response channels [76, 77, 78, 79], but they generally lack a structure to enable easy encoding and fast decoding. In this letter, we address the problem of designing rate-compatible structured LDPC codes that are capacity-approaching for ISI channels.

In this chapter, we design a protograph-based LDPC code over binary-input partial response channels. A protograph code [17] is a structured LDPC code that can be constructed from a small protograph. Protograph codes over AWGN channels have demonstrated very good thresholds with low encoder complexity as well as fast decoding [26, 20]. In the context of coding for channels with memory, this chapter for the first time presents a method to compute an iterative decoding threshold of a protograph-based code. We then provide a simple optimization method for the threshold to find a protograph that is i.u.d. capacity approaching.

We briefly review some works related to this area. Franceschini et al. [79] proposed a design method based on the EXIT chart [70] to optimize an irregular LDPC code for coded modulation in ISI channels. However, the general EXIT chart cannot be used to compute an iterative decoding threshold of protograph-based LDPC codes [28]. A general LDPC code design for partial response channels was proposed by Kavcic et al. [75] via a modified density evolution [9]. Following the method of [75], several irregular LDPC codes optimized for certain specific partial response channels were reported including [76, 77, 78] and others. Recently, Fang et al. [80] designed a nested protograph family that is good for the dicode and EPR4 channels based on finite-length EXIT chart. However, none of the existing design methods produce rate-compatible codes for ISI channels.

The main contribution of this chapter is a *structured* coding scheme based on protographs that is i.u.d. capacity approaching. We find out that a protograph code that is good for AWGN channels may not perform well on partial response channels, in particular protograph codes containing punctured nodes seem to perform poorly with the Bahl-Cocke-Jelinek-Raviv (BCJR) equalizer. We design two families of protograph codes: the nested high-rate protographs where high-rate codes are built from a low-rate protograph by adding more variable nodes; and the rate-compatible protographs where information blocklength of all members are identical. Rate-compatible codes enjoy lower overall complexity since a group of codes can be implemented with a common encoder/decoder. All proposed codes have thresholds that are within a gap of 0.5 dB to i.u.d capacity. The implementation of the proposed codes with 16k data over the dicode and EPR4 channels are reported. The performance of our codes exhibits a frame error rate of $3 \times 10^{-6}$ at a gap of 1.1 dB from the i.u.d capacity limits.

## 6.1 System Model

A general system with a partial response channel is shown in Figure 6.1. An information sequence $\{b_i\}$ is encoded by a LDPC encoder to obtain a coded bit sequence $\{x_i\}$, $1 \leq i \leq n$, which is transmitted with BPSK modulation. A partial response channel, also known as

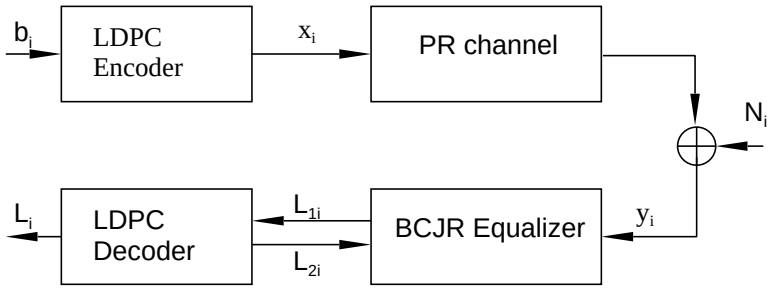


Figure 6.1. System diagram

an inter-symbol interference (ISI) channel, refers to an AWGN channel with memory, often modeled with a finite impulse response $h(D) = 1+h_1D+h_2D^2+\ldots+h_\nu D^\nu$. The discrete-time baseband channel model can be written as

$$y_i = \sum_{k=0}^{\nu} h_k x_{i-k} + N_i \tag{6.1}$$

where $\nu$ is the channel memory, $y_i$ is the received signal, $\{h_k\}_0^\nu$ is a set of channel coefficients, assumed to be known at the receiver, and $N_i$ is white Gaussian noise with zero mean and the variance of $N_0/2$. In this chapter, we mainly consider two popular channels for our code designs, the dicode channel ($h(D) = (1 - D)/\sqrt{2}$) and the EPR4 channel ($h(D) = (1 + D - D^2 - D^3)/2$).

The receiver includes a BCJR equalizer concatenated with a standard message passing LDPC decoder. After receiving $y_i$, $1 \leq i \leq n$, the BCJR equalizer outputs the log-likelihood ratios (LLRs)

$$L_{1i} = \log \frac{P(x_i = +1|y)}{P(x_i = -1|y)} - L_{2i} \tag{6.2}$$

where $L_{2i}$ is a priori information of $x_i$, obtained from the message passing decoding in the previous iteration. The message passing decoder takes the inputs $L_{1i}$ and computes an output soft information $L_i$. The extrinsic information that is passed to the BCJR equalizer is $L_{2i} = L_i - L_{1i}$.

## 6.2 Iterative Decoding Thresholds

An iterative decoding threshold of a protograph is the minimum channel quality that supports reliable iterative decoding of asymptotically large LDPC codes built from the protograph. The receiver in Figure 6.1 includes a BCJR equalizer concatenated with a LDPC decoder. The graph representation of a joint protograph with trellis nodes which represent the state structure of a channel with memory is shown in Figure 6.2. In order to compute the iterative threshold of the joint graph, we proposed a method that combines the EXIT analysis for the BCJR equalizer [70, 81, 82] with the protograph-based EXIT analysis [28].

Assuming that a protograph has $N$ variable nodes $V_1, \ldots, V_N$ and $M$ check nodes $C_1, \ldots, C_M$. We denote with $e_{i,j}$ the number of edges that connects $V_i$ to $C_j$, where $i = 1, \ldots, N$ and $j = 1, \ldots, M$. For convenience, we follow the notations of [28]. In each case, we use the word "knowledge" to indicate the mutual information between the LLRs and the coded bits of the codeword.

- $I_{Ad}, I_{Ed}$: a priori and a posteriori knowledge of the BCJR equalizer
- $I_{ch}^{i}$: a priori knowledge of a variable node $V_i$ in the LDPC decoder. If $V_i$ is punctured, $I_{ch}^{i} = 0$.
- $I_{Ev}^{i,j}, I_{Ec}^{i,j}$: outgoing extrinsic knowledge from a variable node $V_i$ to a check node $C_j$, and from a check node $C_i$ to a variable node $C_j$, respectively.
- $I_{Av}^{i,j}, I_{Ac}^{i,j}$: incoming a priori knowledge from $C^i$ to $V^j$, and from $V^i$ to $C^j$, respectively.
- $I_{APP}^{i}$: a posteriori knowledge evaluated by $V^i$.

The algorithm for the calculation of the iterative decoding threshold is as follows. We begin with an initial a priori mutual information and an initial $E_b/N_0$ as input parameters for the BCJR equalizer, which in turn provides an initial mutual information for the LDPC decoder.

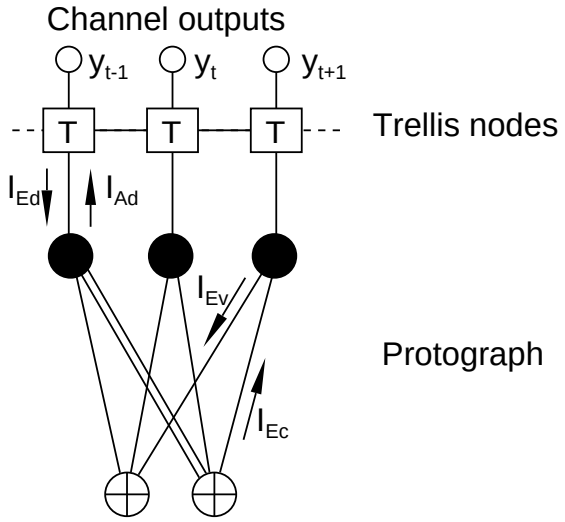


Figure 6.2. Joint protograph with BCJR equalizer

1. Initialization

   Set $E_b/N_0 = SNR_0$, and $I_{Ad} = 0$.

2. Computing a posteriori knowledge of the BCJR equalizer

   The mutual information $I_{Ed}$ can be calculated from the $LLR$ outputs of the BCJR equalizer [75, 82, 81], which in turn are obtained from the Monte Carlo simulation for the given $E_b/N_0$ and an a priori mutual information $I_{Ad}$.

3. Mutual information transfers from the BCJR equalizer to the variable nodes

   Set $I_{ch}^i = I_{Ed}$, for $i = 1, \ldots, N$. If $V^i$ is punctured, $I_{ch}^i = 0$.

4. Variable to check update

   For $j = 1, \ldots, N$ and $i = 1, \ldots, M$, if $e^{i,j} \neq 0$,

$$I_{Ev}^{i,j} = J\left(\left(\sum_{s \neq i} e^{s,j}[J^{-1}(I_{Av}^{s,j})]^2 + +(e^{i,j} - 1)[J^{-1}(I_{Av}^{i,j})]^2) + [J^{-1}(I_{ch}^i)]^2\right)^{1/2}\right)$$

   If $e^{i,j} = 0$, $I_{Ev}^{i,j} = 0$. For $j = 1, \ldots, N$ and $i = 1, \ldots, M$, set $I_{Ac}^{i,j} = I_{Ev}^{i,j}$

5. Check to variable update

   For $j = 1, \ldots, N$ and $i = 1, \ldots, M$, if $e^{i,j} \neq 0$,

   $$I_{Ec}^{i,j} = 1 - J\left(\left(\sum_{s \neq j} e^{s,j}[J^{-1}(1 - I_{Ac}^{i,s})]^2 + +(e^{i,j} - 1)[J^{-1}(1 - I_{Ac}^{i,j})]^2\right)^{1/2}\right)$$

   If $e^{i,j} = 0$, $I_{Ec}^{i,j} = 0$. For $j = 1, \ldots, N$ and $i = 1, \ldots, M$, set $I_{Av}^{i,j} = I_{Ec}^{i,j}$

6. Mutual information transfers from the LDPC decoder to the BCJR equalizer as a priori information $I_{Ad}$

   $$I_{Ad} = \frac{1}{N}\sum_{j=1}^{N} J\left(\left(\sum_{s} e^{s,j}[J^{-1}(I_{Av}^{s,j})]^2\right)^{1/2}\right) \tag{6.3}$$

7. APP-LLR mutual information evaluation

   For $j = 1, \ldots, N$

   $$I_{APP}^{j} = J\left(\left(\sum_{s} e^{s,j}[J^{-1}(I_{Av}^{s,j})]^2 + [J^{-1}(I_{ch}^{i})]^2\right)^{1/2}\right) \tag{6.4}$$

8. Repeat steps 2-7 until $I_{APP} = 1$, $\forall j$ or until maximum number of iteration.

This algorithm determines whether $I_{APP} = 1$ at a given SNR. Via bisection, one may then find the lowest SNR for which this property is true, i.e., the lowest SNR at which the decoder converges, which is the decoding threshold.

## 6.3 Protograph Design

In this section, we propose a simple approach to search for a good protograph with a low iterative decoding threshold in a partial response channel. We concentrate on graphs that possess good protograph properties: a low decoding threshold and linear minimum distance growth property [20] which reduces the likelihood of a shallow error floor. Protographs that lead to linear minimum distance growth must have degree-1 variable nodes [18, 19], as well as a fraction of high-degree and degree-2 nodes [9].

We begin by calculating the threshold of protographs that are designed for the AWGN channel but are used in ISI channels. This will shed light on whether existing AWGN codes are good enough for ISI channels. We revisit the proposed rate-1/2 code in Figure 3.2 whose iterative decoding threshold in the AWGN channel is 0.395 dB that has a gap of only 0.208 dB to capacity.

We calculate the thresholds of the AWGN code over the dicode and the EPR4 channels. The thresholds are given in Table 6.1. These thresholds are about 1 dB and 1.424 dB away from the i.u.d. capacity limits of the dicode and EPR4 channels respectively. Such big gaps show that the AWGN code may not perform well over these partial response channels. We conjecture that the poor performance may in part be due to punctured nodes (which are often used in AWGN protograph codes [20]), which may not be beneficial to the extrinsic information exchange between the LDPC decoder and the BCJR equalizer. That observation motivates us to seek protographs that do not contain punctured nodes.

To demonstrate with an example, let us search for a rate-1/2 protograph that contains 4 check nodes and 8 variable nodes without any punctured node. As mentioned in the first paragraph of this section, a good protograph should contain several nodes with degrees 1 and 2. Thus, to greatly reduce the search space, we just start by a search structure with two degree-1 and one degree-2 variable nodes in the form of

$$H^{1/2}_{4\times 8,search} = \begin{pmatrix} x_1 & x_5 & x_9 & x_{13} & 0 & 0 & 0 & 1 \\ x_2 & x_6 & x_{10} & x_{14} & 0 & 0 & 1 & 0 \\ x_3 & x_7 & x_{11} & x_{15} & 1 & 1 & 0 & 0 \\ x_4 & x_8 & x_{12} & x_{16} & 2 & 1 & 0 & 0 \end{pmatrix} \tag{6.5}$$

where $x_i$, $i = 1, \ldots, 16$, are the number of edges connecting their associated column (variable node) and row (check node). In order for the code to have the linear minimum distance growth property, the edge summation over the last two rows within first 4 columns should be 3 or higher [29], i.e. $x_3 + x_4 \geq 3$, $x_7 + x_8 \geq 3$, $x_{11} + x_{12} \geq 3$ and $x_{15} + x_{16} \geq 3$. We can further simplify the problem by limiting $x_i \in \{0, 1, 2\}$. Our objective in this specific example is to find a protograph that has the lowest iterative decoding threshold over the

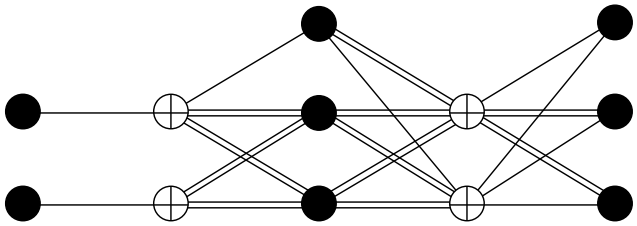

Figure 6.3. The rate-1/2 protograph with the $4\times8$ structure optimized for the dicode channel

Table 6.1. Thresholds (dB) of rate-1/2 AWGN and ISI codes in the dicode and EPR4 channels

| | Dicode $h(D) = (1-D)/\sqrt{2}$ | | EPR4 $h(D) = (1+D-D^2-D^3)/2$ | |
|---|---|---|---|---|
| | Threshold | Gap | Threshold | Gap |
| AWGN code | 1.821 | 0.998 | 2.584 | 1.424 |
| ISI-code | 1.217 | 0.394 | 1.531 | 0.371 |

dicode channel. The resulting protograph is in the form of

$$H^{1/2}_{4\times8,ISI} = \begin{pmatrix} 2 & 2 & 0 & 0 & 0 & 0 & 0 & 1 \\ 2 & 2 & 1 & 0 & 0 & 0 & 1 & 0 \\ 2 & 2 & 1 & 1 & 1 & 1 & 0 & 0 \\ 2 & 2 & 2 & 2 & 2 & 1 & 0 & 0 \end{pmatrix} \tag{6.6}$$

The designed ISI code is shown in Figure 6.3, and the corresponding thresholds in two channels are given in Table 6.1. This code has decoding thresholds within gaps of 0.394 dB and 0.371 away from the i.u.d. capacity limits of the dicode and EPR4 channels, respectively. The ISI code is also capacity approaching for the EPR4 channel, although it is optimized for the dicode channel.

It is worth noting that we can also apply the same approach to design a smaller protograph. For example, let us design a rate-1/2 protograph with 6 variable nodes and 3 check nodes, then using a search structure with two degree-1 and one degree-2 variable nodes in

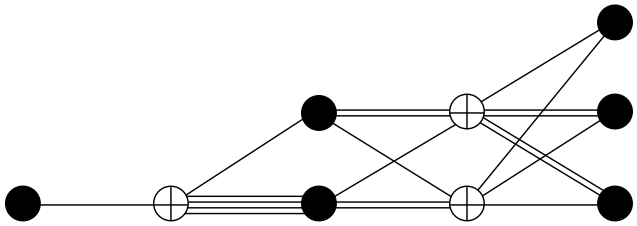

Figure 6.4. The rate-1/2 protograph with the $3\times6$ structure optimized for the dicode channel

the form of

$$H^{1/2}_{3\times6,search} = \begin{pmatrix} 1 & 0 & x_1 & x_4 & x_7 & y_1 \\ 0 & 1 & x_2 & x_5 & x_8 & y_2 \\ 0 & 1 & x_3 & x_6 & x_9 & y_3 \end{pmatrix} \tag{6.7}$$

where $x_i$, $i = 1,\ldots,9$ and $y_j$, $j = 1,\ldots,3$ , are the number of edges connecting their associated row (check node) and column (variable node) in which $y_j$ correspond to the highest degree variable node. In order for the code to have the linear minimum distance growth property, the edge summation over the last two rows within last 4 columns should be 3 or higher [29], i.e. $x_2 + x_3 \geq 3$, $x_5 + x_6 \geq 3$, $x_8 + x_9 \geq 3$ and $y_2 + y_3 \geq 3$. We can further simplify the problem by limiting $x_i \in \{0, 1, 2\}$ and $y_j \in \{1, 2, 3, 4\}$. The resulting protograph is in the form of

$$H^{1/2}_{3\times6,ISI} = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 4 \\ 0 & 1 & 2 & 1 & 2 & 2 \\ 0 & 1 & 1 & 2 & 1 & 1 \end{pmatrix} \tag{6.8}$$

The designed ISI code is shown in Figure 6.4, and the corresponding thresholds in two channels are given in Table 6.2. This code has decoding thresholds within gaps of 0.5 dB away from the i.u.d. capacity limits of the dicode and EPR4 channels. Since the $3 \times 6$ protograph family has a smaller search space compared with that of the $4 \times 8$ structure, it produces the code with a slightly higher iterative threshold compared with the larger protograph.

## 6.4 Nested and Rate-Compatible Codes

In the previous section, we proposed a method to design of a rate-1/2 protograph in ISI channels. However, practical systems, e.g. magnetic recoding or optical communication systems, require LDPC codes with rates up to 0.9. Applying this direct method to design high-rate codes is more time consuming since high-rate codes would have larger code search space.

Although puncturing is a common method for generating rate-compatible codes, in general it is not fruitful for channels with memory, as pointed out by [83] except when only degree-1 variable nodes are punctured [80]. In this section, we use extension to build rate-compatible codes. As we show in the sequel, nested high-rate and rate-compatible codes that can be built by extending from a rate-1/2 code are still i.u.d. capacity approaching.

### 6.4.1 Nested High-rate Codes

In order to use the extension method to design rate-compatible codes, we first need to design a high-rate code. We use a nested lengthening structure as follows

$$H_h = [H_l H_e] \tag{6.9}$$

where $H_h$, $H_l$ are protomatrices of a high-rate and low-rate protographs respectively, $H_e$ is an extension matrix whose columns are new added variable nodes and its elements are number of edges connecting a new variable node to an old check node.

In this example, we start from the rate-1/2 shown in Figure 6.4 as our daughter code. We then design a family of nested codes with rates in the form of $R = \frac{n}{n+1}$, $n = 1, 2, \ldots$. Within this nested family, each high rate code is built upon adding 3 new variable nodes[1] into the lower-rate code. We then apply the same search method used in Section 6.3 to find the codes with the lowest threshold. We then are able to design the code with rate up to 9/10. Due to space limitation, we only present the result of the rate 0.9 code in which other eight codes are deduced. The rate-0.9 code protomatrix is given as follows:

[1] If we use the rate-1/2 code in [83], 4 new variable nodes are needed instead.

$$H_{0.9} = \left(\begin{array}{cccccc|ccc|ccc|ccc|ccc|ccc|ccc} 1&0&0&1&0&4&2&0&0&0&0&2&2&0&0&0&0&1&0&0&0&2&0&0&0&0&0&0&0&1 \\ 0&1&2&1&2&2&1&2&2&2&1&2&2&2&2&2&2&1&2&2&2&1&2&1&1&1&1&2&2&1 \\ 0&1&1&2&1&1&2&1&1&1&2&1&1&1&1&1&1&2&2&1&1&2&1&2&2&2&2&2&1&2 \end{array}\right) \tag{6.10}$$

In the above protomatrix, we separate each code rate level by a line as shown in Equation (6.10). The thresholds of these nine nested codes are shown in Table 6.2 where the high-rate codes have their iterative thresholds within 0.4 dB to i.u.d capacity of dicode and EPR4 channels. Considering this small gap to i.u.d capacity of ISI channels, we can say that nested high-rate codes are good enough and there no need to design a high-rate code directly as described in Section 6.3.

### 6.4.2 Rate-compatible Codes

Using the extension method, the low rate protomatrix is in the following form

$$H_l = \left(\begin{array}{cc} H_1 & 0 \\ A & B \end{array}\right) \tag{6.11}$$

where $H_l$, $H_1$ is the protomatrix of the low-rate and high-rate codes respectively, 0 is the zero matrix, $A$ is the matrix whose elements are number of edges connecting new check nodes to the old variable nodes, and $B$ is the matrix whose elements are the number of edges connecting between new check nodes to new variable nodes. To simplify the problem, we assume that $B$ is identity.

Starting with the rate-9/10 code, we design a family of rate-compatible codes as follows. Each time, we add one new variable and one check nodes to the protomatrix of the high-rate code. The new codes have the rates in the form of $R = \frac{27}{30+m}$, where $m$ is the number of variable and check nodes added into the rate-9/10 code. Due to space limitation, we only design nine rate-compatible codes with rates from 27/41 to 9/10, where the biggest graph/code has the lowest rate of $R = 27/41$. Equation (6.12) shows its protomatrix from which other eight rate-compatible code protomatrices are derived. Iterative thresholds of these rate-compatible codes over dicode and EPR4 channels are shown in Table 6.2. Again,

Table 6.2. Threshold $E_b/N_0$ (dB) of new codes

| | dicode | | EPR4 | |
|---|---|---|---|---|
| Code Rate | Code thres. | Gap to cap. | Code thres. | Gap to cap. |
| The nested family | | | | |
| 1/2 | 1.3 | 0.5 | 1.7 | 0.5 |
| 2/3 | 2.2 | 0.4 | 2.6 | 0.4 |
| 3/4 | 2.7 | 0.3 | 3.2 | 0.4 |
| 4/5 | 3.1 | 0.3 | 3.6 | 0.4 |
| 5/6 | 3.3 | 0.2 | 3.9 | 0.3 |
| 6/7 | 3.6 | 0.2 | 4.1 | 0.3 |
| 7/8 | 3.8 | 0.2 | 4.3 | 0.3 |
| 8/9 | 4.0 | 0.3 | 4.2 | 0.3 |
| 9/10 | 4.2 | 0.3 | 4.7 | 0.4 |
| The rate-compatible family | | | | |
| 9/10 | 4.2 | 0.3 | 4.7 | 0.4 |
| 27/31 | 4.0 | 0.4 | 4.3 | 0.3 |
| 27/32 | 3.6 | 0.3 | 4.0 | 0.3 |
| 27/33 | 3.3 | 0.3 | 3.7 | 0.3 |
| 27/34 | 2.9 | 0.2 | 3.5 | 0.3 |
| 27/35 | 2.9 | 0.3 | 3.3 | 0.3 |
| 27/37 | 2.6 | 0.3 | 3.0 | 0.3 |
| 27/39 | 2.3 | 0.3 | 2.7 | 0.3 |
| 27/41 | 2.1 | 0.3 | 2.5 | 0.3 |

all these codes can operate closely to capacity with thresholds gaps of 0.4 dB to i.u.d capacity limits.

## 6.5 Numerical Results

Our protograph codes are derived from proto-matrices (protographs) designed in Section 6.4 in two lifting steps. First, the protograph is lifted by a factor of 4 using the progressive edge growth (PEG) algorithm [48] in order to remove all parallel edges. Then, a second lifting using the PEG algorithm was performed to determine a circulant permutation of each edge class that would yield the desired code block length. In this section, protograph codes are

$$H_{14\times 41} = \begin{pmatrix}
1&0&0&1&0&4&2&0&0&0&0&2&2&0&0&0&0&1&0&0&0&2&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0\\
0&1&2&1&2&2&1&2&2&2&1&2&2&2&2&2&2&1&2&2&2&1&2&1&1&1&1&2&2&1&0&0&0&0&0&0&0&0&0&0&0\\
0&1&1&2&1&1&2&1&1&1&2&1&1&1&1&1&1&2&2&1&1&2&1&2&2&2&2&2&1&2&0&0&0&0&0&0&0&0&0&0&0\\
0&0&1&1&1&1&1&0&1&0&0&1&0&0&0&0&0&0&0&0&0&0&0&1&1&0&0&1&0&0&1&0&0&0&0&0&0&0&0&0&0\\
0&0&1&1&1&2&0&0&1&0&1&0&1&1&1&0&1&0&0&0&0&0&0&0&0&0&0&0&0&1&0&1&0&0&0&0&0&0&0&0&0\\
0&0&1&1&1&1&1&0&1&1&0&1&1&0&1&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0\\
0&1&0&0&0&2&0&0&0&0&1&1&1&0&1&0&1&0&0&0&0&1&0&1&0&0&0&1&0&1&0&0&0&1&0&0&0&0&0&0&0\\
0&0&1&0&1&1&0&0&0&0&0&0&1&1&0&0&0&0&0&0&0&0&1&0&0&1&0&0&0&1&0&0&0&0&1&0&0&0&0&0&0\\
0&0&0&0&0&2&0&0&1&0&0&0&1&0&0&0&1&0&0&0&0&1&1&0&1&0&0&0&0&1&0&0&0&0&0&1&0&0&0&0&0\\
0&0&0&0&0&2&0&0&0&0&0&1&0&1&0&0&0&0&0&0&1&0&0&0&0&0&0&1&0&1&0&0&0&0&0&0&1&0&0&0&0\\
0&0&0&0&0&1&0&0&0&0&0&1&1&1&0&0&0&0&0&0&0&1&0&0&0&0&0&1&1&1&0&0&0&0&0&0&0&1&0&0&0\\
0&0&0&0&0&2&0&0&1&0&0&1&0&0&0&0&1&0&0&0&1&1&0&0&0&0&1&0&0&1&0&0&0&0&0&0&0&0&1&0&0\\
0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&1&0&0&0&1&0&0&0&1&0&0&1&1&1&0&0&0&0&0&0&0&0&0&1&0\\
0&0&0&0&0&1&1&0&0&1&0&0&0&0&0&1&0&0&0&0&1&0&0&0&1&0&1&1&0&0&0&0&0&0&0&0&0&0&0&0&1
\end{pmatrix} \tag{6.12}$$

simulated with the information blocklength of 16k. For nested codes, lifting factors of codes with rates 1/2, 2/3, 3/4, 4/5, 5/6, 6/7, 7/8, 9/10 are $4 \times 1364$, $4 \times 683$, $4 \times 455$, $4 \times 342$, $4 \times 273$, $4 \times 227$, $4 \times 195$, and $4 \times 153$, respectively. For rate-compatible codes, only the lowest rate code whose protomatrix is shown in (6.12) is constructed with the lifting factor of $4 \times 153$. Other codes are obtained by removing redundant coded bits and check equations.

The WER and BER performances of the rate-1/2 AWGN and ISI codes over the dicode channel are shown in Figure 6.5. No error floors are observed up to $WER = 10^{-6}$, where the ISI code has a gain of 0.7 dB over the AWGN code and has a gap of 0.9 dB from the i.u.d. capacity limit.

The WER performance of the nested and rate-compatible codes over the dicode and EPR4 channels are shown in Figures 6.6, 6.7, 6.8 and 6.9 respectively. No error floors are observed up to $WER = 3 \times 10^{-6}$, with a gap of 1.1 dB from the i.u.d. capacity limit.

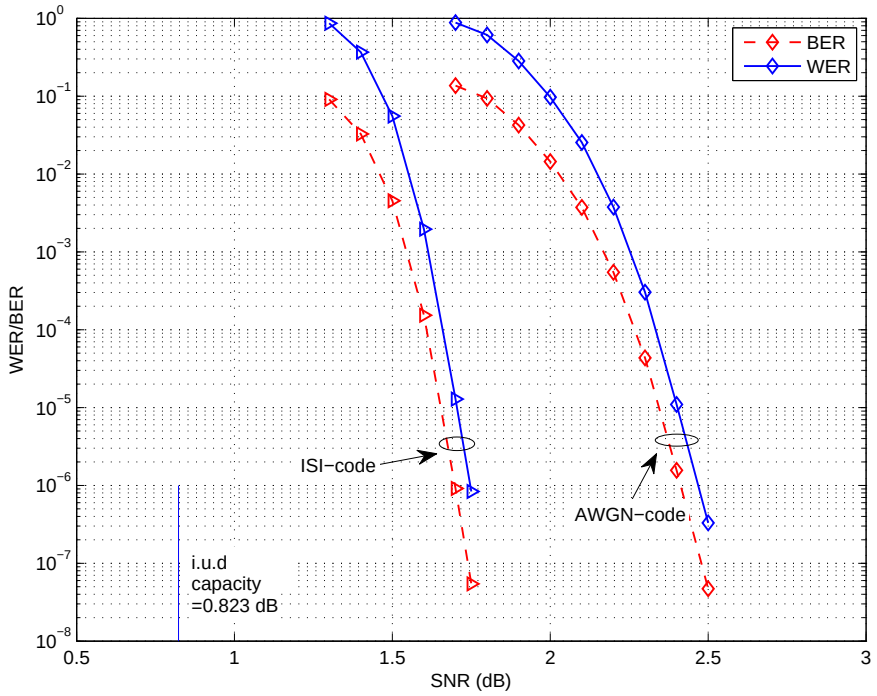


Figure 6.5. Performance of the rate-1/2 AWGN protograph plotted in Figure 3.2 and the rate-1/2 ISI protograph plotted in Figure 6.3 in the dicode channel. The information block-length of 16k.

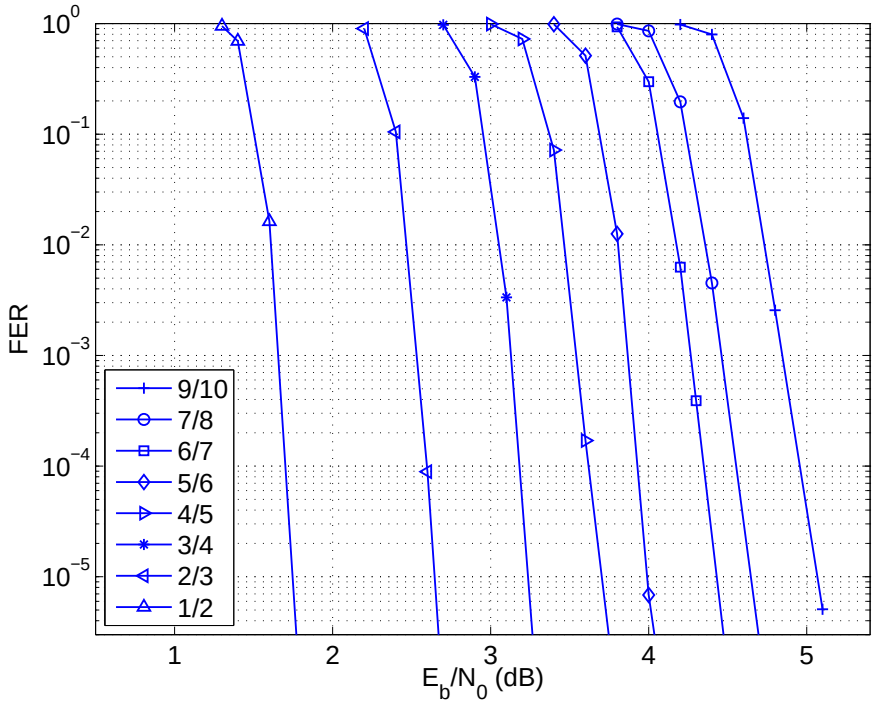


Figure 6.6. Performance of nested protograph family over the dicode channel.

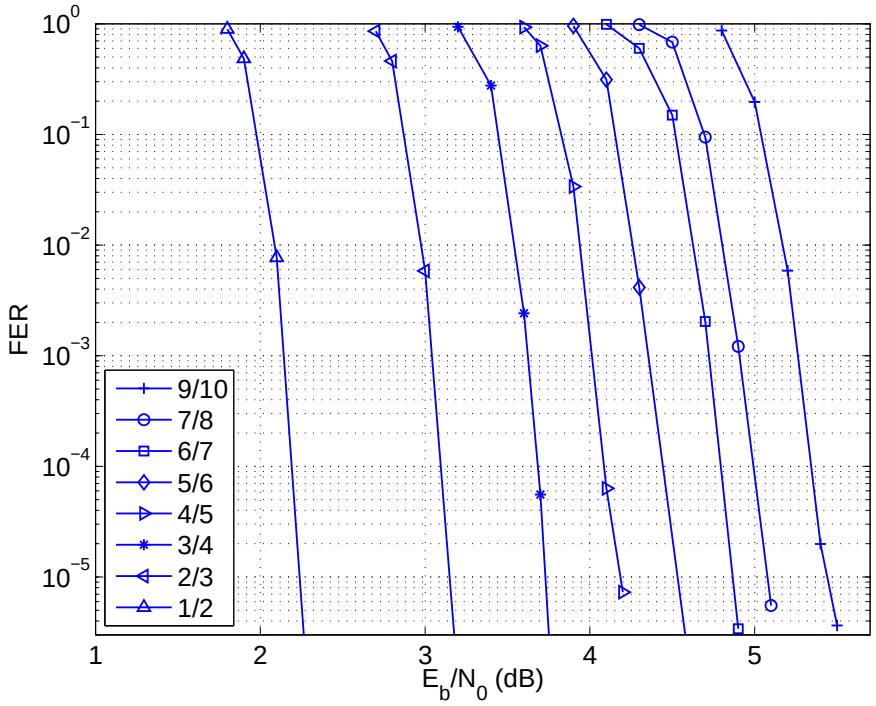


Figure 6.7. Performance of nested protograph family over the EPR4 channel.

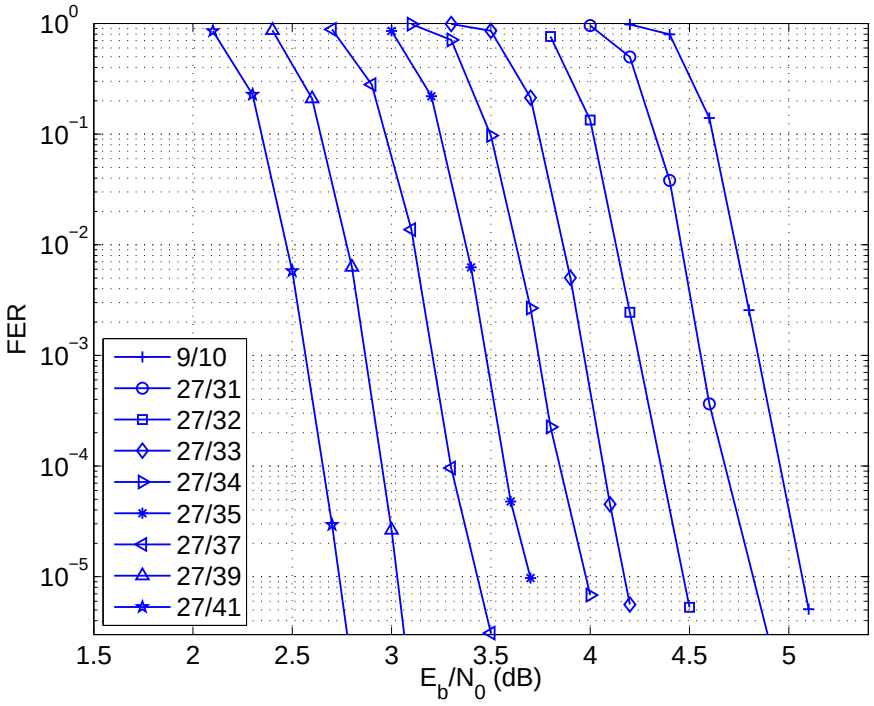


Figure 6.8. Performance of rate-compatible protograph family in the dicode channel.

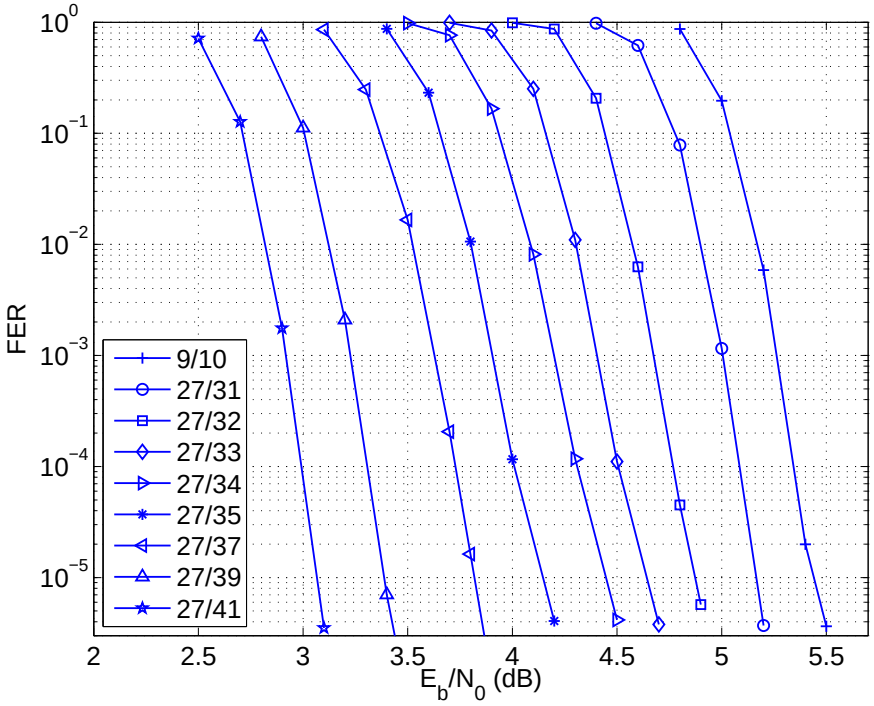


Figure 6.9. Performance of rate-compatible protograph family in the EPR4 channel.

## 6.6 Conclusion

This chapter presents a design for nested and rate-compatible protograph-based LDPC codes for ISI channels. Iterative thresholds and finite-length performance of the codes are reported. Analysis and simulation results show that our codes, which allow easy encoding and fast decoding, can perform closely to i.u.d capacity.

# CHAPTER 7
# CONCLUSIONS AND FUTURE WORK

This chapter summarizes the contributions of this dissertation and provides some possible avenues for future research in the area of protograph-based LDPC codes. The findings of this dissertation also appear in [47, 84, 85, 30, 86, 83].

This dissertation presents a simple yet effective method for designing nested families of LDPC codes. Rate compatible codes are essential for many communication applications, e.g. hybrid automatic repeat request (HARQ) systems, and their design is nontrivial due to the difficulty of simultaneously guaranteeing the quality of several related codes. Puncturing can be used to generate rate-compatible LDPC codes, but it produces a gap to capacity that, in practice, often significantly exceeds the gap of the mother code. We propose an alternative method based on successively extending a high-rate protograph. The resulting codes not only inherit the advantages of protograph codes, namely low encoding complexity and efficient decoding algorithms, but also cover a wide range of rates and have very good performance with iterative decoding thresholds that are within 0.2 dB of their capacity limits.

This dissertation studies the problem of designing protograph codes over relay channels. Despite encouraging advances in the design of relay codes, several important challenges remain. Many of the existing LDPC relay codes are tightly optimized for fixed channel conditions and not easily adapted without extensive re-optimization of the code. Some have high encoding complexity and some need long block lengths to approach capacity. A high-performance protograph-based LDPC coding scheme is devised for the half-duplex relay channel that addresses simultaneously several important issues: structured coding that permits easy design, low encoding complexity, embedded structure for convenient adaptation to various channel conditions, and performance close to capacity with a reasonable block length. The application of the coding structure to multi-relay networks is demonstrated.

Finally, a simple new methodology for evaluating the end-to-end error performance of relay coding systems is developed and used to highlight the performance of the proposed codes.

This dissertation studies the application of protograph codes in coded modulation schemes. Protograph-based bit-interleaved coded modulation (BICM) provides an elegant way of designing coded modulation over Rayleigh faded channels, however, to date the available designs have been limited to specific modulations and the corresponding decoding thresholds have not been known for Rayleigh faded channels. A simple method for designing protograph-based BICM that is devised is general and applies to any modulation, and furthermore we calculate the iterative decoding thresholds of the protograph codes while mapped to higher order modulations. This general coding framework can support not only multiple rates but also adaptive modulation. We report that certain families of protograph codes achieve a threshold within a gap of approximately $0.2 - 0.4$ dB of BICM capacity limit across a wide range of rates and modulations.

This dissertation presents the design of a family of rate-compatible protograph-based LDPC codes which can approach the independent and uniformly distributed (i.u.d.) capacity of inter-symbol interference (ISI) channels. This problem is non-trivial due to the joint design of structured (protograph-based) LDPC codes and the state structure of ISI channels. We describe a method to design nested high-rate protograph codes by adding variable nodes into the protograph of a lower rate code. We then design a family of rate-compatible protograph codes using extension method, i.e. adding the same number of variable and check nodes into the graph of a high-rate code. Resulting protograph codes have iterative thresholds close to i.u.d capacity. Numerical results are provided to support our analysis.

We foresee some extensions to the research topics proposed in this dissertation. Designing practical coding schemes for multiple node wireless networks has been a long standing open problem. Our coding scheme for a relay channel could be used as a starting point for designing general multi-node networks. An interesting direction is to extend our design framework for channels such as multiple access channels, broadcast channels and interference channels. Interactions with network coding is also an interesting problem to be considered.

Another important research topic is to design protograph codes for MIMO and OFDM fading channels. Even though there are several random LDPC code designs for MIMO and OFDM configurations, there is a need to design protograph codes to approach the fundamental limits of practical MIMO-OFDM systems.

Practical coding for non-coherent communication systems is also another interesting direction. A good start research is to extend the proposed framework to design good protograph codes for differential modulation schemes for the point-to-point AWGN channel.

One promising area where findings from this dissertation can be applied is the problem of designing practical nested lattice codes. We expect that solving this problem will produce many new results and facilitate the practical application of network information theory.

# APPENDIX A

## A.1 The Progressive-Edge-Growth Algorithm

The PEG algorithm [48] generates a LDPC Tanner graph from a degree distribution such that the Tanner graph has a large local girth. We start with a protograph that has $v$ variable nodes and $c$ check nodes, and $N$ is the lifting factor. Variable nodes of the protograph are ordered according to increasing degree. The resulting graph has $n = N \times v$ variable nodes and $m = N \times c$ check nodes. Unlike a general irregular LDPC codes, in a protograph code each variable node is only connected to particular set of check nodes, called a type, as imposed by the protograph structure.

Our protograph codes are built from a protograph via a two-step lifting. In the first step, the protograph is lifted by a small factor, which is 4 for all codes in this dissertation, to remove all parallel edges in the protograph. The first lifting step is not necessarily circulant (in fact it usually is not). Then in the second step this intermediate protograph is lifted again to the full code. This second lifting step is via a circulant lifting. Both the lifting steps are performed using the PEG algorithm. In the first lifting step, the PEG algorithm operates on all edges. In the second step, the PEG algorithm operates only on the first edge of each type, since the other edges of the same type follow the same behavior due to the circulant lifting.

We use the same notations and definitions as in [48] with the exception that $\bar{\mathcal{N}}_{s_j}^l$ is defined as $V_c \setminus \mathcal{N}_s^l$ where $V_c$ is the set of checks allowed to connect to the variable node $s$ as imposed by the protograph structure. The standard PEG algorithm for constructing a protograph code is described in the following.

**for** $j = 0 \rightarrow n-1$ **do**

**for** $k = 0 \rightarrow d_{s_j} - 1$ **do**

**if** $k = 0$ **then** $E^0_{s_j} \leftarrow$ edge $(c_i, s_j)$, where $E^0_{s_j}$ is the first edge incident to symbol node $s_j$, and $c_i$ is one check node such that it belongs to one of $N$ check nodes of the same type as imposed by the protograph structure.

**else**

expand a subgraph from symbol node $s_j$ up to depth $l$ under the current graph setting such that the cardinality of $\mathcal{N}^l_{s_j}$ stops increasing but is less than $m$, or $\bar{\mathcal{N}}^l_{s_j} \neq \emptyset$ but $\bar{\mathcal{N}}^{l+1}_{s_j} = \emptyset$, then $E^k_{s_j} \leftarrow$ edge $(c_i, s_j)$, where $E^k_{s_j}$ is the $k^{th}$ edge incident to symbol node $s_j$, and $c_i$ is a check picked from the set $\bar{\mathcal{N}}^l_{s_j}$.

**end if**

**end for**

**end for**

## A.2 Protograph-Based EXIT Analysis for AWGN Channels

The Protograph-based EXIT (PEXIT) analysis was originally proposed by Liva and Chiani [28] by extending the standard extrinsic information transfer (EXIT) analysis [14] to calculate iterative decoding thresholds of protograph codes.

Assuming that a protograph has $N$ variable nodes $V_1, \ldots, V_N$ and $M$ check nodes $C_1, \ldots, C_M$. We denote $e_{i,j}$ the number of edges that connects $V_i$ to $C_j$, where $i = 1, \ldots, N$ and $j = 1, \ldots, M$. For convenience, we follow the notations of [28].

- $I^i_{ch}$: a priori channel mutual information of a variable node $V_i$. If $V_i$ is punctured, $I^i_{ch} = 0$.
- $I^{i,j}_{Ev}$: the mutual information between the message sent by $V_j$ to $C_i$ and associated codeword bit, on one of the $b_{i,j}$ edges connecting $V_j$ to $C_i$. ($I^{i,j}_{Ev} = 0$ if $b_{i,j} = 0$)
- $I^{i,j}_{Ec}$: the mutual information between the message sent by $C_i$ to $V_j$ and associated codeword bit, on one of the $b_{i,j}$ edges connecting $C_i$ to $V_j$. ($I^{i,j}_{Ec} = 0$ if $b_{i,j} = 0$)

- $I^i_{APP}$: the mutual information between the a posteriori probability log-likelihood ratio (modeled as a random variable)

$$L^j = \ln \frac{Pr(x_j = +1|r)}{Pr(x_j = -1|r)} \tag{A.8.1}$$

evaluated by $V_j$ and the associated codeword bit $x_j$. The received vector, given by the transmitted codeword corrupted by the channel, is denoted by $r$.

We denote $J(\sigma)$ the mutual information between a binary random variable $X$ with $Pr(X = +\mu) = 1/2$ and $Pr(X = -\mu) = 1/2$, and a continuous Gaussian distributed random variable $Y$ with mean $\mu$ and variance $\sigma^2 = 2\mu$. $J(\sigma)$ represents the capacity of the binary-input additive Gaussian noise, given in [14].

$$J(\sigma) = 1 - \int_{-\infty}^{+\infty} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(y-\sigma^2/2)^2}{2\sigma^2}} \log_2(1 + e^{-y}) dy \tag{A.8.2}$$

$J(\sigma)$ function can be computed easily by simple approximations [70] as follows. For $\sigma^* = 1.6363$,

$$J(\sigma) \simeq \begin{cases} a_{J,1}\sigma^3 + b_{J,1}\sigma^2 + c_{J,1}\sigma, & 0 \le \sigma \le \sigma^* \\ 1 - e^{a_{J,2}\sigma^3 + b_{J,2}\sigma^2 + c_{J,2}\sigma + d_{J,2}}, & \sigma^* \le \sigma \le 10 \\ 1, & \sigma \ge 10 \end{cases} \tag{A.8.3}$$

where

$$\begin{aligned} &a_{J,1} = -0.0421061 \quad b_{J,1} = 0.209252 \quad c_{J,1} = -0.00640081 \\ &a_{J,2} = 0.00181491 \quad b_{J,2} = -0.142675 \quad c_{J,2} = -0.0822054 \quad d_{J,2} = 0.0549608. \end{aligned}$$

The inverse function $J^{-1}(\cdot)$ is approximated as follows. For $I^* = 0.3646$,

$$J^{-1}(I) \simeq \begin{cases} a_{\sigma,1}I^2 + b_{\sigma,1}I + c_{\sigma,1}\sqrt{(I)}, & 0 \le I \le I^* \\ -a_{\sigma,2}\ln[b_{\sigma,2}(1-I)] - c_{\sigma,2}I, & I^* \le I \le 1 \end{cases} \tag{A.8.4}$$

where

$$\begin{aligned} &a_{\sigma,1} = 1.09542 \quad b_{\sigma,1} = 0.214217 \quad c_{\sigma,1} = 2.33727 \\ &a_{\sigma,2} = 0.706692 \quad b_{\sigma,2} = 0.386013 \quad c_{\sigma,2} = -1.75017. \end{aligned}$$

The algorithm presents an initialization step, which provides the channel mutual information for a given channel parameter $\frac{E_b}{N_0}$. The limit value of this channel parameter, which allows the decoder to converge successfully to an error probability $P_e = 0$ in the limit of infinite blocklengths, is called the iterative decoding threshold $(\frac{E_b}{N_0})^*$.

1. Initialization

   Set $I_{ch}^j = J(\sigma_{ch,j})$, for $j = 0, \ldots, N-1$, with

   $$\sigma_{ch,j}^2 = 8R \left( \frac{E_b}{N_0} \right)_{V^j} \tag{A.8.5}$$

   where $R$ is the code rate of the protograph and $\left(\frac{E_b}{N_0}\right)_{V^j}$ represents the signal-to-noise ratio associated to the channel input to the $j$-th variable node. Note that if $V^j$ is punctured, $\left(\frac{E_b}{N_0}\right)_{V^j} = 0$.

2. Variable to check update

   For $j = 0, \ldots, N-1$ and $i = 0, \ldots, M-1$, if $b^{i,j} \neq 0$,

   $$I_{Ev}^{i,j} = J\left( \Big( \sum_{s \neq i} b^{s,j} [J^{-1}(I_{Av}^{s,j})]^2 + (b^{i,j} - 1)[J^{-1}(I_{Av}^{i,j})]^2) + [J^{-1}(I_{ch}^i)]^2 \Big)^{1/2} \right)$$

   If $b^{i,j} = 0$, $I_{Ev}^{i,j} = 0$. For $j = 0, \ldots, N-1$ and $i = 0, \ldots, M-1$, set $I_{Ac}^{i,j} = I_{Ev}^{i,j}$

3. Check to variable update

   For $j = 0, \ldots, N-1$ and $i = 0, \ldots, M-1$, if $b^{i,j} \neq 0$,

   $$I_{Ec}^{i,j} = 1 - J\left( \Big( \sum_{s \neq j} b^{s,j} [J^{-1}(1 - I_{Ac}^{i,s})]^2 + (b^{i,j} - 1)[J^{-1}(1 - I_{Ac}^{i,j})]^2 \Big)^{1/2} \right)$$

   If $b^{i,j} = 0$, $I_{Ec}^{i,j} = 0$. For $j = 0, \ldots, N-1$ and $i = 0, \ldots, M-1$, set $I_{Av}^{i,j} = I_{Ec}^{i,j}$

4. APP-LLR mutual information evaluation

   For $j = 0, \ldots, N-1$

   $$I_{APP}^j = J\left( \Big( \sum_{s} b^{s,j} [J^{-1}(I_{Av}^{s,j})]^2 + [J^{-1}(I_{ch}^i)]^2 \Big)^{1/2} \right) \tag{A.8.6}$$

5. Repeat steps 2-5 until $I_{APP} = 1, \forall j$ or until maximum number of iteration.

The threshold $(E_b/N_0)^*$ is the lowest value for which the mutual information between the APP-LLR messages and the associate codeword symbol converges to 1.

## A.3 The Sum-Product Algorithm

This section presents the iterative message passing algorithm to decode the protograph codes that is used throughout this dissertation. In our simulations we use the fixed-point iterative decoder with 8-bit quantization as described in [65]. Using the same notions and definitions as in [65], a summary of the decoding algorithm is given in the following.

### A.3.1 Quantization

In a practical decoder, log-likelihood ratios (LLRs) are represented by digital quantities. This quantization limits both the dynamic range and the resolution of the LLRs. A quantization is in the form of

$$Q(x) = \begin{cases} 127 & Cx \geq 127 \\ -127 & Cx \leq -127 \\ [Cx] & \text{otherwise} \end{cases} \tag{A.8.7}$$

where $C$ is a scalar constant, and $[Cx]$ stands for rounding $CX$ to the nearest integer. In our code simulations, we use $C = 8$, which corresponds to a step-size of $1/8$ and an LLR dynamic range of $(-15\frac{7}{8}, +15\frac{7}{8})$.

### A.3.2 Variable Node Processing

A given variable node receives LLR messages $u_1, u_2, \ldots, u_d$ from $d$ check nodes, where $d$ is the degree of the variable node, along with an $LLR$ from the demodulator. The message the variable node sends back to the $j$th of the $d$ check nodes connected to it is given by

$$v_j = \lambda + \sum_{i=1, i\neq j}^{d} u_i \tag{A.8.8}$$

Given the quantized inputs $Q(\lambda)$ and $Q(u_i)$, the outgoing quantized message may be computed as

$$Q(v_j) = c\left(Q(\lambda) + \sum_{i=1, i\neq j}^{d} Q(u_i)\right) \tag{A.8.9}$$

where

$$c(x) = \begin{cases} 127 & x \geq 127 \\ -127 & x \leq -127 \\ x & \text{otherwise} \end{cases} \tag{A.8.10}$$

This can also be written as

$$Q(v_j) = c(U - u_j) \tag{A.8.11}$$

where $U \triangleq Q(\lambda) + \sum_{i=1}^{d} Q(u_i)$. $U$ can be clipped prior to the subtraction

$$Q(v_j) = c(c(U) - u_j) \tag{A.8.12}$$

This further clipping improves the performance of protograph codes and is called "Jones clipping" [65].

As discussed in Chapter 3, a good protograph often includes degree-1 variable nodes. The performance of protograph codes can be improved further in the error floor region by clipping the degree-1 variable messages to 116 as described in [65]. Note that the best clipping may depend on the protograph structure. In our simulations, the degree-1 clipping is set to 90.

### A.3.3 Check Node Processing

A given check node receives messages $v_1, v_2, \ldots, v_d$ from $d$ variable nodes, where $d$ is the degree of the check node. The message the check node send back to the $j$th of the $d$ variable nodes connected to it is given by

$$u_j = 2\tanh^{-1}\left(\prod_{i=1, i\neq j}^{d} \tanh\frac{v_i}{2}\right) \tag{A.8.13}$$

This can be computed by repetitively applying the function

$$\min{}^*(x,y) = 2\tanh^{-1}\left(\tanh\frac{x}{2}\tanh\frac{y}{2}\right) \tag{A.8.14}$$

$$= \mathrm{sgn}(xy)\left[\min(|x|,|y|) - \ln(1+e^{-(||x|-|y||)}) + \ln(1+e^{-(|x|+|y|)})\right] \tag{A.8.15}$$

In the quantization representation, the above relationship is given by

$$Q(\min{}^*(Q(x),Q(y))) = Q\left\{2\tanh^{-1}\left(\tanh\frac{Q(x)}{2C}\tanh\frac{Q(y)}{2C}\right)\right\} \tag{A.8.16}$$

This function can be computed once, ahead of time, using a look-up table.

One additional decoder variation also makes a big different in the error floor performance: partial hard limiting check node messages. Messages from each check node were partially hard-limited, so that every message from a check node which has $|Q(u_j)| \geq 100$, was set to the maximum magnitude $|Q(u_j)| = 127$.

## REFERENCES


[1] C. E. Shannon, "A mathematical theory of communication," *The Bell system technical journal*, vol. 27, pp. 379–423, July 1948.

[2] G. Forney and D. Costello, "Channel coding: The road to channel capacity," *Proceedings of the IEEE*, vol. 95, no. 6, pp. 1150 –1177, June 2007.

[3] R. G. Gallager, "Low-density parity-check codes," Ph.D. dissertation, MIT, Cambrigde, Massachusetts, 1963.

[4] D. J. C. MacKay and R. M. Neal, "Good codes on very sparse matrices," in *Proc. Cryptography Coding, 5th IMA Conf.*, C. Boyd, Ed. Berlin, Germany: Springer, 1995, pp. 100 –111.

[5] M. Sipser and D. Spielman, "Expander codes," *IEEE Trans. Inform. Theory*, vol. 42, no. 6, pp. 1710 –1722, Nov. 1996.

[6] V. V. Zyablov and M. S. Pinsker, "Estimation of the error-correction complexity of Gallager low-density codes," *Problems of Information Transmission*, vol. 11, pp. 23 –26, Jan. 1975.

[7] R. Tanner, "A recursive approach to low complexity codes," *IEEE Trans. Inform. Theory*, vol. 27, no. 5, pp. 533 – 547, Sept. 1981.

[8] G. A. Margulis, "Explicit constructions of graphs without short cycles and low density codes," *Combinatorica*, vol. 2, no. 1, pp. 71 –78, 1982.

[9] T. J. Richardson, M. A. Shokrollahi, and R. L. Urbanke, "Design of capacity-approaching irregular low-density parity-check codes," *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 619–637, Feb. 2001.

[10] S. Y. Chung, "On the contruction of some capacity-approaching coding schemes," Ph.D. dissertation, MIT, Cambrigde, Massachusetts, Sept. 2000.

[11] J. Hou, P. Siegel, and L. Milstein, "Performance analysis and code optimization of low density parity-check codes on Rayleigh fading channels," *IEEE J. Select. Areas Commun.*, vol. 19, no. 5, pp. 924 –934, May 2001.

[12] M. Luby, M. Mitzenmacher, A. Shokrollahi, D. Spielman, and V. Stemann, "Practical loss-resilient codes," in *Proc. 29th Annu. ACM Symp. Theory of Computing*, 1997, pp. 150–159.

[13] T. J. Richardson and R. L. Urbanke, "The capacity of low-density parity-check codes under message-passing decoding," *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 599–618, Feb. 2001.

[14] S. ten Brink, "Convergence behavior of iteratively decoded parallel concatenated codes," *IEEE Trans. Commun.*, vol. 49, no. 10, pp. 1727–1737, Oct. 2001.

[15] D. Divsalar, H. Jin, and R. J. McEliece, "Coding theorems for "Turbo-Like" codes," in *Proc. Allerton Conf. Commun., Control, and Computing*, Sept. 1998, pp. 201 –210.

[16] Y. Kou, S. Lin, and M. Fossorier, "Low-density parity-check codes based on finite geometries: a rediscovery and new results," *IEEE Trans. Inform. Theory*, vol. 47, no. 7, pp. 2711 –2736, Nov. 2001.

[17] J. Thorpe, "Low-density parity-check (LDPC) codes constructed from protogrtaphs," IPN Progress Report 42-154, Aug. 2003.

[18] T. Richardson and R. Urbanke, "Multi-edge type LDPC codes," LTHC, Tech. Rep., 2004, submitted IEEE IT.

[19] A. Abbasfar, D. Divsalar, and K. Yao, "Accumulate-repeat-accumulate codes," *IEEE Trans. Commun.*, vol. 55, no. 4, pp. 692–702, Apr. 2007.

[20] D. Divsalar, S. Dolinar, C. R. Jones, and K. Andrews, "Capacity-approaching protograph codes," *IEEE J. Select. Areas Commun.*, vol. 27, no. 6, pp. 876–888, Aug. 2009.

[21] A. Jimenez-Feltstrom and K. S. Zigangirov, "Time-varying periodic convolutional codes with low-density parity-check matrix," *IEEE Trans. Inform. Theory*, vol. 45, pp. 2181 –2191, Sept. 1999.

[22] T. Richardson and R. Urbanke, *Modern Coding theory*. Cambridge University Press, 2008.

[23] W. E. Ryan and S. Lin, *Channel codes: Classical and Modern*. Cambridge University Press, 2009.

[24] S.-Y. Chung, J. Forney, G.D., T. Richardson, and R. Urbanke, "On the design of low-density parity-check codes within 0.0045 db of the shannon limit," *IEEE Commun. Lett.*, vol. 5, no. 2, pp. 58 –60, Feb. 2001.

[25] Z. Li, L. Chen, L. Zeng, S. Lin, and W. Fong, "Efficient encoding of quasi-cyclic low-density parity-check codes," *IEEE Trans. Commun.*, vol. 54, no. 1, pp. 71–81, Jan. 2006.

[26] J. Thorpe, K. Andrews, and S. Dolinar, "Methodologies for designing LDPC codes using protographs and circulants," in *Proc. IEEE ISIT*, July 2004, pp. 236–236.

[27] T. J. Richardson, "Error floors of LDPC codes," in *Proc. Allerton Conf. Commun., Control, and Computing*, 2003, pp. 1426–1435.

[28] G. Liva and M. Chiani, "Protograph LDPC codes design based on EXIT analysis," in *Proc. IEEE GLOBECOM*, Nov. 2007, pp. 3250–3254.

[29] S. Abu-Surra, D. Divsalar, and W. Ryan, "On the existence of typical minimum distance for protograph-based LDPC codes," in *Information Theory and Applications Workshop (ITA)*, Jan. 2010, pp. 1–7.

[30] T. V. Nguyen, A. Nosratinia, and D. Divsalar, "The design of rate-compatible protograph LDPC codes," *IEEE Trans. Commun.*, vol. 60, no. 10, pp. 2841–2850, Oct. 2012.

[31] J. Hagenauer, "Rate-compatible punctured convolutional codes (RCPC codes) and their applications," *IEEE Trans. Commun.*, vol. 36, no. 4, pp. 389 –400, Apr. 1988.

[32] J. Ha, J. Kim, and S. McLaughlin, "Rate-compatible puncturing of low-density parity-check codes," *IEEE Trans. Inform. Theory*, vol. 50, no. 11, pp. 2824 – 2836, Nov. 2004.

[33] H. Pishro-Nik and F. Fekri, "Results on punctured LDPC codes," in *Proc. IEEE Information Theory Workshop*, Oct. 2004, pp. 215 – 219.

[34] T. Tian and C. R. Jones, "Construction of rate-compatible LDPC codes utilizing information shortening and parity puncturing," *EURASIP J. Wirel. Commun. Netw.*, vol. 2005, no. 5, pp. 789–795, 2005.

[35] M. R. Yazdani and A. H. Banihashemi, "On construction of rate-compatible low-density parity-check codes," *IEEE Commun. Lett.*, vol. 8, no. 3, pp. 159 –161, Mar. 2004.

[36] J. Ha, J. Kim, D. Klinc, and S. McLaughlin, "Rate-compatible punctured low-density parity-check codes with short block lengths," *IEEE Trans. Inform. Theory*, vol. 52, no. 2, pp. 728 –738, Feb. 2006.

[37] H. Y. Park, J. W. Kang, K. S. Kim, and K. C. Whang, "Efficient puncturing method for rate-compatible low-density parity-check codes," *IEEE Trans. Wireless Commun.*, vol. 6, no. 11, pp. 3914 –3919, Nov. 2007.

[38] B. Vellambi and F. Fekri, "Finite-length rate-compatible LDPC codes: a novel puncturing scheme," *IEEE Trans. Commun.*, vol. 57, no. 2, pp. 297–301, Feb. 2009.

[39] N. Jacobsen and R. Soni, "Design of rate-compatible irregular LDPC codes based on edge growth and parity splitting," in *Proc. IEEE Vehicular Technology Conference (VTC)*, Oct. 2007, pp. 1052 –1056.

[40] C.-H. Hsu and A. Anastasopoulos, "Capacity achieving LDPC codes through puncturing," *IEEE Trans. Inform. Theory*, vol. 54, no. 10, pp. 4698 –4706, Oct. 2008.

[41] J. Li and K. R. Narayanan, "Rate-compatible low density parity check codes for capacity-approaching ARQ schemes in packet data communications," in *Proc. CIIT*, Nov. 2002.

[42] S. Dolinar, "A rate-compatible family of protograph-based LDPC codes built by expurgation and lengthening," in *Proc. IEEE ISIT*, Sept. 2005, pp. 1627 –1631.

[43] D. Divsalar, S. Dolinar, and C. Jones, "Low-rate ldpc codes with simple protograph structure," in *Proc. IEEE ISIT*, Sept. 2005, pp. 1622 –1626.

[44] M. El-Khamy, J. Hou, and N. Bhushan, "Design of rate-compatible structured LDPC codes for hybrid ARQ applications," *IEEE J. Select. Areas Commun.*, vol. 27, no. 6, pp. 965–973, Aug. 2009.

[45] J. Kim, A. Ramamoorthy, and S. Mclaughlin, "The design of efficiently-encodable rate-compatible LDPC codes," *IEEE Trans. Commun.*, vol. 57, no. 2, pp. 365 –375, Feb. 2009.

[46] C. Shi and A. Ramamoorthy, "Design and analysis of E2RC codes," *IEEE J. Select. Areas Commun.*, vol. 27, no. 6, pp. 889 –898, Aug. 2009.

[47] T. V. Nguyen, A. Nosratinia, and D. Divsalar, "Bilayer protograph codes for half-duplex relay channels," in *Proc. IEEE ISIT*, June 2010, pp. 948–952.

[48] X.-Y. Hu, E. Eleftheriou, and D.-M. Arnold, "Regular and irregular progressive edge-growth Tanner graphs," *IEEE Trans. Inform. Theory*, vol. 51, pp. 386–398, 2003.

[49] CCSDS, "Low density parity check codes for use in near-earth and deep space applications," CCSDS 131.1-O-2. Orange Book, Issue 2, Sept. 2007.

[50] ETSI TR 102 376 v1.1.1, "Digital video broadcasting (DVB) user guidelines for the second generation system for broadcasting, interactive services, news gathering and other broadband satellite applications (DVB-S2)," 2005.

[51] T. Cover and A. E. Gamal, "Capacity theorems for the relay channel," *IEEE Trans. Inform. Theory*, vol. 25, no. 5, pp. 572–584, Sept. 1979.

[52] G. Kramer, M. Gastpar, and P. Gupta, "Cooperative strategies and capacity theorems for relay networks," *IEEE Trans. Inform. Theory*, vol. 51, no. 9, pp. 3037 – 3063, Sept. 2005.

[53] M. Janani, A. Hedayat, T. Hunter, and A. Nosratinia, "Coded cooperation in wireless communications: space-time transmission and iterative decoding," *IEEE Trans. Signal Processing*, vol. 52, no. 2, pp. 362 – 371, Feb. 2004.

[54] T. Hunter and A. Nosratinia, "Cooperation diversity through coding," in *Proc. IEEE ISIT*, June 2002, p. 220.

[55] M. Valenti and B. Zhao, "Distributed turbo codes: towards the capacity of the relay channel," in *Proc. IEEE Vehicular Technology Conference (VTC)*, vol. 1, Oct. 2003, pp. 322 – 326.

[56] Z. Zhang and T. Duman, "Capacity approaching turbo coding for half duplex relaying," in *Proc. IEEE ISIT*, Sept. 2005, pp. 1888 –1892.

[57] ——, "Capacity-approaching turbo coding and iterative decoding for relay channels," *IEEE Trans. Commun.*, vol. 53, no. 11, pp. 1895 – 1905, Nov. 2005.

[58] M. Azmi, J. Yuan, J. Ning, and H. Q. Huynh, "Improved bilayer LDPC codes using irregular check node degree distribution," in *Proc. IEEE ISIT*, July 2008, pp. 141–145.

[59] M. Azmi and J. Yuan, "Design of multi-edge-type bilayer-expurgated LDPC codes," in *Proc. IEEE ISIT*, July 2009, pp. 1988–1992.

[60] J. Hu and T. M. Duman, "Low density parity check codes over half-duplex relay channels," in *Proc. IEEE ISIT*, July 2006, pp. 972–976.

[61] P. Razaghi and W. Yu, "Bilayer low-density parity-check codes for decode-and-forward in relay channels," *IEEE Trans. Inform. Theory*, vol. 53, no. 10, pp. 3723–3739, Oct. 2007.

[62] C. Li, G. Yue, X. Wang, and M. A. Khojastepour, "LDPC code design for half-duplex cooperative relay," *IEEE Trans. Wireless Commun.*, vol. 7, no. 11, pp. 4558–4567, Nov. 2008.

[63] A. Chakrabarti, A. D. Baynast, A. Sabharwal, and B. Aazhang, "Low density parity check codes for the relay channel," *IEEE J. Select. Areas Commun.*, vol. 25, no. 2, pp. 280–291, Feb. 2007.

[64] J. P. Cances and V. Meghdadi, "Optimized low density parity check codes designs for half duplex relay channels," *IEEE Trans. Wireless Commun.*, vol. 8, no. 7, pp. 3390–3395, July 2009.

[65] J. Hamkins, "Performance of low-density parity-check coded modulation," IPN Progress Report 42-184, Feb. 2011.

[66] L.-L. Xie and P. Kumar, "An achievable rate for the multiple-level relay channel," *IEEE Trans. Inform. Theory*, vol. 51, no. 4, pp. 1348 –1358, Apr. 2005.

[67] A. Host-Madsen, "On the capacity of wireless relaying," in *Proc. IEEE Vehicular Technology Conference (VTC)*, vol. 3, 2002, pp. 1333 – 1337.

[68] G. Caire, G. Taricco, and E. Biglieri, "Bit-interleaved coded modulation," *IEEE Trans. Inform. Theory*, vol. 44, no. 3, pp. 927 –946, May 1998.

[69] J. Hou, P. Siegel, L. Milstein, and H. Pfister, "Capacity-approaching bandwidth-efficient coded modulation schemes based on low-density parity-check codes," *IEEE Trans. Inform. Theory*, vol. 49, no. 9, pp. 2141 – 2155, Sept. 2003.

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

[71] H. Niu, M. Shen, and J. Ritcey, "Threshold of LDPC-coded BICM for Rayleigh fading," *IEEE Commun. Lett.*, vol. 8, no. 7, pp. 455 – 457, July 2004.

[72] Y. Li and W. E. Ryan, "Design of LDPC-coded modulation schemes," in *Turbo Codes and Related Topics, 5th International Symposium on*, Sept. 2003.

[73] D. Divsalar and C. Jones, "Protograph based low error floor LDPC coded modulation," in *Proc. IEEE MILCOM*, vol. 1, Oct. 2005, pp. 378 –385.

[74] Y. Jin, M. Jiang, and C. Zhao, "Optimized variable degree matched mapping for protograph LDPC coded modulation with 16QAM," in *Turbo Codes and Iterative Information Processing (ISTC), 6th International Symposium on*, Sept. 2010, pp. 161 –165.

[75] A. Kavcic, X. Ma, and M. Mitzenmacher, "Binary intersymbol interference channels: Gallager codes, density evolution, and code performance bounds," *IEEE Trans. Inform. Theory*, vol. 49, no. 7, pp. 1636 – 1652, July 2003.

[76] A. Thangaraj and S. McLaughlin, "Thresholds and scheduling for LDPC-coded partial response channels," *IEEE Trans. Magn.*, vol. 38, no. 5, pp. 2307 – 2309, Sept. 2002.

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

[78] J. Soriaga, H. Pfister, and P. Siegel, "Determining and approaching achievable rates of binary intersymbol interference channels using multistage decoding," *IEEE Trans. Inform. Theory*, vol. 53, no. 4, pp. 1416 –1429, Apr. 2007.

[79] M. Franceschini, G. Ferrari, and R. Raheli, "EXIT chart-based design of LDPC codes for inter-symbol interference channels," in *Proc. IST Mobile and Wireless Communications*, June 2005.

[80] Y. Fang, P. Chen, L. Wang, and F. Lau, "Design of protograph LDPC codes for partial response channels," *IEEE Trans. Commun.*, vol. 60, no. 10, pp. 2809–2819, Oct. 2012.

[81] J. Hagenauer, "The EXIT chart - Introduction to extrinsic information transfer in iterative processing," in *"Proceedings of 12th European Signal Processing Conference (EUSIPCO)"*, Sept. 2004.

[82] L. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. Inform. Theory*, vol. 20, no. 2, pp. 284 – 287, Mar. 1974.

[83] T. V. Nguyen, A. Nosratinia, and D. Divsalar, "Protograph-based LDPC codes for partial response channels," in *Proc. IEEE ICC*, June 2012, pp. 2194–2199.

[84] ——, "The design of rate-compatible protograph LDPC codes," in *Proc. Allerton Conf. Commun., Control, and Computing*, Oct. 2010, pp. 1–5.

[85] ——, "Threshold of protograph-based LDPC coded BICM for rayleigh fading," in *Proc. IEEE GLOBECOM*, Dec. 2011, pp. 1 –5.

[86] ——, "Bilayer protograph codes for half-duplex relay channels," *IEEE Trans. Wireless Commun.*, 2012, (under revision).

## VITA

Thuy Van Nguyen was born in Hanoi, Vietnam, in 1976. He received the Bachelor of Science degree in Electronics and Telecommunications from Hanoi University of Technology, Hanoi, Vietnam in 1999 and the Master of Science degree in Electrical Engineering from New Mexico State University, Las Cruces, USA in 2005. From September 2004 to May 2005, he was a Teaching Assistant at New Mexico State University. From 2000 to 2003 and from 2005 to 2008, he was a lecturer in Telecommunication-I department at Posts and Telecommunications Institute of Technology, Hanoi, Vietnam. In September 2008, he started his Ph.D. studies in Electrical Engineering at the University of Texas at Dallas.