Efficient Low-Density Parity-Check Codes for Cooperative Communication

A dissertation submitted in partial satisfaction of the requirements for the degree
Doctor of Philosophy

in

Electrical Engineering (Communications and Signal Processing)

by

Osso Vahabzadeh

Committee in charge:

Professor Masoud Salehi, Chair
Professor Milica Stojanovic
Professor Edmund Yeh

2014
DEDICATION

To the loving memory of my mother,
and to my father.
# TABLE OF CONTENTS

Signature Page .............................................................. iii  
Dedication ................................................................. iv  
Table of Contents ......................................................... v  
List of Figures ............................................................ viii  
List of Tables ............................................................. x  
Acknowledgements ........................................................ xi  
Abstract of the Dissertation .......................................... xiii

Chapter 1  
1.1 Fundamentals of Error Control Coding ............................ 3 
  1.1.1 Linear Block Codes ............................................ 3 
  1.1.2 Techniques to Modify Linear Block Codes .................. 4 
  1.1.3 LDPC Codes ................................................... 5 
  1.1.4 The Sum-Product Decoding Algorithm ....................... 5 
  1.1.5 Density Evolution ............................................ 7 
  1.1.6 Extrinsic information transfer chart analysis .............. 9 
1.2 The Relay Channel .................................................. 11 
  1.2.1 Decode-and-Forward Relaying Strategy ...................... 13 
  1.2.2 A Two-User Cooperation Scheme ............................ 14 
1.3 Bilayer LDPC Codes ................................................ 15 
  1.3.1 Bilayer Lengthened LDPC Codes ............................ 15 
  1.3.2 Bilayer Expurgated LDPC Codes ............................ 17 
1.4 Protograph-Based LDPC Codes .................................... 18 
1.5 The Organization of this Dissertation ........................... 20

Chapter 2  
Complexity-Optimized Bilayer Lengthened LDPC Codes for Relay Channels ................................................ 22 
2.1 Introduction ......................................................... 22 
2.2 Preliminaries ......................................................... 24 
  2.2.1 The Relay Channel ............................................ 24 
  2.2.2 Bilayer Lengthened LDPC Codes ............................ 26 
  2.2.3 Relaying Strategy Based on Bilayer Lengthened LDPC Codes 27 
  2.2.4 Bilayer Density Evolution ................................... 28 
2.3 Extending EXIT Charts to Bilayer Lengthened LDPC Codes ... 29
<table>
<thead>
<tr>
<th>Chapter 6</th>
<th>Conclusion</th>
<th>100</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bibliography</td>
<td></td>
<td>103</td>
</tr>
</tbody>
</table>
LIST OF FIGURES

Figure 1.1: A bipartite graph. ................................................................. 6
Figure 1.2: EXIT Chart. ....................................................................... 10
Figure 1.3: The relay channel model. ................................................. 12
Figure 1.4: A two-user cooperation scheme [1]. ................................. 15
Figure 1.5: A bilayer lengthened LDPC code. ................................. 17
Figure 1.6: A bilayer expurgated LDPC code. ................................. 19
Figure 1.7: Lifting a protograph. ............................................................... 19

Figure 2.1: The relay channel model. .................................................. 24
Figure 2.2: A bilayer lengthened LDPC code. ...................................... 27
Figure 2.3: The EXIT chart of a bilayer lengthened LDPC code. ............. 31
Figure 2.4: Decoding complexity of the designed bilayer lengthened LDPC codes. .................................................. 40
Figure 2.5: Comparison of performance of codes (C_4), (C_5), and (C_6), evaluated at their target number of iterations with the performance of code (F) at the same number of iterations. .................................................. 41

Figure 3.1: The relay channel model. .................................................. 43

Figure 4.1: Two mobile users cooperate to transmit their messages to the base station. .................................................. 56
Figure 4.2: Protograph of the AR4JA class of LDPC codes. ....................... 57
Figure 4.3: Time division scheduling for transmission of each user’s message. .................................................. 58
Figure 4.4: Frame error-rate (FER) performance as a function of $E_b/N_0$ for different interuser channel qualities ($R = \frac{3}{4}$). .................................................. 61
Figure 4.5: Frame error-rate (FER) performance as a function of $E_b/N_0$ for different interuser channel qualities ($R = \frac{5}{4}$). .................................................. 62

Figure 5.1: Two mobile users cooperate to transmit their messages to the base station. .................................................. 67
Figure 5.2: Time division scheduling for transmission of each user’s message. .................................................. 68
Figure 5.3: The Parity-Check Matrix of a Root-LDPC Code of Rate 0.5. .................................................. 69
Figure 5.4: The Parity-Check Matrix of an RCR-LDPC Code of Rate 0.33. .................................................. 70
Figure 5.5: A PB Root-LDPC Code of Rate 0.5. .................................................. 73
Figure 5.6: A PB-RCR LDPC Code of Rate 0.33. .................................................. 74
Figure 5.7: Normalized logarithmic asymptotic weight distribution $r(\delta)$ for protograph codes of Figures 5.5 and 5.6. .................................................. 77
Figure 5.8: Applying the check node splitting technique to a degree-5 check node. .................................................. 78
Figure 5.9: RCPB-R-LDPC code construction (Method I). .................................................. 80
Figure 5.10: RCPB-R-LDPC code construction (Method II). .................................................. 83
Figure 5.11: An example of a family of RCPB-R-LDPC codes constructed using Method I. .................................................. 85
Figure 5.12: An example of a family of RCPB-R-LDPC codes constructed using Method II. .................................................. 87
Figure 5.13: An example of a family of RCPB-R-LDPC codes constructed using Method II. .......................................................... 88
Figure 5.14: The protograph of a rate 0.33 code with no root structure. .............. 90
Figure 5.15: The parity-check matrix of the regular (3, 9, 3, 6) RCR-LDPC code of rate 0.33 reported in [2] where the dots represent 1’s. ......................... 91
Figure 5.16: Comparison of the performance of three rate-0.33 codes: The regular (3, 9, 3, 6) RCR-LDPC code reported in [2]; the code $C_{1,c2}$ of Figure 5.11(c); and the protograph-based code of Figure 5.14 with no root structure. ... 92
Figure 5.17: Performance results for finite length $C_1$ codes of rates 0.50 and 0.40 under Scenario 1 where the interuser channel is error free. ......................... 95
Figure 5.18: Performance results for finite length $C_1$ codes of rates 0.33 and 0.28 under Scenario 1 where the interuser channel is error free. ......................... 96
Figure 5.19: Performance results for finite length $C_1$ code of rate 0.25 under Scenario 1 and comparison of all rates of $C_1$ code family. ......................... 97
Figure 5.20: Performance results for finite length $C_2$ codes of rates 0.50 and 0.44 under Scenario 2 where the SNR of the interuser channel is 5 dB higher than the SNR of the user-BS channel. ................................. 98
Figure 5.21: Performance results for finite length $C_2$ code of rate 0.36 under Scenario 2 and comparison of all rates of $C_2$ code family. ......................... 99
LIST OF TABLES

Table 2.1: Number of Iterations Required for the Convergence of Code (F) and Complexity-optimized Codes at Different Values of SNR (dB) . . . . . . . . . . . . . 39

Table 3.1: Optimized Degree Distributions for SNR_{SR} = 1.5SNR_{SD} . . . . . . . . . . 51
Table 3.2: Optimized Degree Distributions for SNR_{SR}(dB) = SNR_{SD}(dB) + 5dB . . . 52
ACKNOWLEDGEMENTS

I would like to express my sincerest gratitude to my advisor, Professor Masoud Salehi, for his wonderful guidance and mentorship throughout this work and my education at Northeastern University. Professor Salehi’s character has been truly inspirational for me. His deep and broad knowledge in the fields of communication, coding, and information theory has been exemplary. Furthermore, his critical thinking and deep insights on different problems of interest have always helped me to view problems from different perspectives. I would also like to thank Professor Salehi for his tireless editorial effort which has improved the quality of my work throughout.

I also wish to thank my dissertation committee members, Professor Milica Stojanovic and Professor Edmund Yeh. Their valuable feedback and support during my research have greatly improved the quality of this dissertation.

I am grateful to Dr. Thomas Richardson, Dr. Dariush Divsalar, Dr. Peyman Razaghi, Dr. Marwan Azmi, Dr. Pirouz Zarrinkhat, and Dr. Raman Yazdani for the insightful discussions that we had regarding different aspects of my research and for always answering my endless questions.

I am also thankful to Professors Bahram Shafai, Hanoch Lev-Ari, and Amir Farhat for their help and support throughout my education and Ms Faith Crisley and Ms Joan Pratt for their wonderful service to the department of Electrical and Computer Engineering and the Communications and Digital Signal Processing (CDSP) laboratory.

I would like also to thank my friends and colleagues in the CDSP lab, Yashar Aval, Yueqian Li, Parastoo Qarabaqi, Koorosh Firouzbakht, Fatemeh Fazel, Kuo Lun Huang, Rasoul Ghadami, Anshuman Mishra, Francesco Zorzi, and Rameez Ahmed for the productive discussions and memorable moments we had and for our wonderful friendships.

I am particularly indebted to my parents, my sisters, my brother, and auntie Zahra for their support, encouragement, and love throughout my life. Their support has strengthened me during those many years I was away from them. I am truly proud of them.

Last but not least, my special thanks goes to Leila, for enriching my life and for her support, encouragement and love in the process of completion of this work.
ABSTRACT OF THE DISSERTATION

Efficient Low-Density Parity-Check Codes for Cooperative Communication

by

Osso Vahabzadeh

Doctor of Philosophy in Electrical Engineering (Communications and Signal Processing)

Northeastern University, Boston, MA, 2014

Professor Masoud Salehi, Chair

In this dissertation, we address code design problem for cooperative communication over different channel models with emphasis on low complexity designs and structured codes that are attractive for practical implementation.

We start with the problem of designing efficient codes for the relay node in Gaussian relay channels. For a class of capacity approaching codes for this channel model, called bilayer lengthened LDPC (BL-LDPC) codes, we calculate a measure of decoding complexity as a function of the number of decoding iterations and propose a technique to design complexity-optimized BL-LDPC codes by minimizing the complexity measure of these codes. This is made possible by generalizing the EXIT charts to the case of BL-LDPC codes. Motivated by the fact that there are usually stricter hardware restrictions at the relay node, our tech-
nique targets minimizing the decoding complexity of the relay code. Furthermore, excessive delay due to decoding high rate codes at the relay results in additional delay at the destination. Using our technique, we design bilayer codes with noticeable reduction in decoding complexity and delay compared to the rate-optimized codes reported in the literature.

Next, we study the achievable rates for the decode-and-forward (DF) relaying strategy for the Rayleigh fading relay channel where the links have independent normalized Rayleigh fading coefficients and the channel side information is perfectly known at the corresponding receivers but not at the transmitters. We design BL-LDPC codes for this scenario for the case when the source-relay link is much stronger than the source-destination link as well as for the case when these two links have comparable SNRs.

We also propose a novel two-user cooperation scheme for the block fading channel model that employs protograph-based LDPC codes. The proposed scenario is based on time division where each user transmits its message to the base station (BS) in two successive frames. Cooperation is performed by employing the Alamouti scheme Whenever it is possible. Additionally, the users encode their information over protograph-based LDPC codes that allow flexible selection of rates and code lengths.

Finally, we introduce rate-compatible protograph-based root LDPC (RCPB-R-LDPC) codes for cooperative communication over block fading channels and propose two methods to construct these codes. The proposed techniques are based on the extension technique and offer broad design rates resulting in high flexibility. Furthermore, they are based on protograph constructions with minimum distance growing linearly with the block length, a property that improves the error floor performance of the designed codes. The outage probability limit under BPSK modulation is obtained for the cooperative scheme employed in this work and was used to evaluate the WER performance of the designed codes.
Chapter 1

Introduction

Cooperative communication has attracted significant attention in recent years as a class of strategies for achieving higher data rates by providing cooperative diversity. Relaying is one of the most widely used techniques for cooperation. Cooperative diversity allows different relay nodes to construct a virtual antenna array and thus benefit from spatial diversity. The classical relay channel model consists of three nodes; a source node (S), a relay node (R), and a destination node (D) [3], where the relay assists the destination by forwarding to it a function of the information which it receives from the source. Different relaying strategies for the relay channel are categorized based on whether the relay decodes the received signal from the source or merely forwards a function of the received signal to the destination [4,5]. The relaying strategy where the relay decodes the received signal from the source and then transmits an encoded version of it to the destination is called decode-and-forward (DF) which usually employs random binning [4]. In this strategy, the relay decodes the source message and sends its bin index to the destination. The destination receives the bin index of the source message and using the entire information received in the previous block decodes the source message. The code design problem for the decode-and-forward strategy for relaying requires designing a code $X$ at the source which is decodable at the relay, as well as designing a code $X_1$ at the relay which is decodable at the destination. The destination first decodes $X_1$ and with the help of the extra information gained by decoding $X_1$ (i.e. the bin index of $X$), it decodes $X$ [4].

Most of the coding schemes that have been proposed for relay channels do not design
codes specifically to achieve the capacity of the DF scheme [6–10]. However, methods for designing two classes of unstructured bilayer low-density parity-check (LDPC) codes called \emph{bilayer lengthened} and \emph{bilayer expurgated} LDPC codes were proposed in [11, 12] that efficiently implement the binning technique and target the capacity of the DF strategy for Gaussian relay channels.

Despite the progress that has been made to apply or design high rate unstructured LDPC codes for relay channels, the complexity problem for these codes has not been addressed so far. This is an important issue at all nodes; and in particular, at the relay node since there are generally stricter hardware and power constraints at the relay. Additionally, the decoding delay at the relay decreases the throughput at the destination.

Several protocols to obtain cooperative diversity in wireless networks were proposed in [13–17]. Cooperation between user mobiles was proposed in [1,18] as a new form of spatial diversity. User cooperation strategy is studied based on the assumption that the mobile users wish to transmit their own information to the base station (BS) and whenever possible, they assist their partner user by relaying its information to the BS.

Designing efficient error correcting codes to utilize user-cooperation has been an interesting area of research since user cooperation was introduced as a technique to achieve spatial diversity, see [2,19–22] for a number of examples. A channel model that has been in the center of attention in many of these works is the block fading (BF) [23] frequency nonselective channel. The majority of the techniques that are proposed for \emph{coded cooperation} employ rate-compatible (RC) convolutional or LDPC codes. RC-LDPC codes are excellent error-correcting codes. However, the proposed RC-LDPC codes for the BF channel employ unstructured designs which make their practical implementation impossible.

In this dissertation, we address several problems: We design bilayer lengthened LDPC codes for Rayleigh fading relay channels. Furthermore, we formulate a technique to design complexity-optimized unstructured bilayer lengthened LDPC codes for relay channels. We also present a cooperation technique that employs space-time codes [24] and structured LDPC codes [25]. We finally address the code design problem for cooperative communication in BF channels by proposing two techniques for designing high performance structured rate-compatible LDPC codes.

In the rest of this chapter, we review the main concepts and techniques implemented
in this work. We begin with reviewing the main concepts of linear block codes and an important class of these codes, i.e., the LDPC codes. We also overview the important tools used to design and analyze these codes. Then we describe the relay channel model and the fundamental relaying techniques that are proposed for this channel. We further discuss important channel models for user cooperation. Then an efficient implementation of LDPC codes for relay channels, i.e., bilayer LDPC codes, is described. In addition, we review the fundamentals concepts of protograph-based LDPC codes, a class of structured codes that we employ in our designs. We conclude this chapter by reviewing the contributions of this dissertation.

1.1 Fundamentals of Error Control Coding

1.1.1 Linear Block Codes

A linear block code $C$ is a $k$-dimensional subspace of an $n$-dimensional space and can be generated using a $k \times n$ generator matrix which is denoted by $G$. The matrix $G$ maps any data block $u$ of length $k$ into a codeword $c$ of length $n$, i.e., $c = uG$. The extra $m = n - k$ added bits are called the parity check bits which add the redundancy to protect the information data against errors. The resulting code is called an $(n, k)$ linear block code. Whenever the information bits are embedded in the encoded bits, it is called a systematic code, and otherwise, it is called a non-systematic code. The rate $R$ of the code defined here is given by

$$R = \frac{k}{n} = 1 - \frac{m}{n}. \quad (1.1)$$

At a given signal-to-noise ratio (SNR), a well designed lower rate code is usually more successful in correcting errors because it has more redundant parity-check bits compared to a higher rate code.

The Hamming distance between two codewords $c_1, c_2 \in C$ is the number of components at which $c_1$ and $c_2$ differ. Furthermore, the minimum distance of the code $d_{\text{min}}$ is the minimum Hamming distance between any two distinct codewords of the code. The $d_{\text{min}}$ of a code determines error detecting and error correcting capabilities of the code.
The dual code of $\mathcal{C}$ is the $m$-dimensional orthogonal complement of $\mathcal{C}$ and can be generated using the $m \times n$ generator matrix of the dual code which is denoted by $H$. As a result, every row of $H$ is orthogonal to any codeword $c \in \mathcal{C}$. In other words, $cH^t = 0$. Matrix $H$ is called the parity-check matrix of $\mathcal{C}$.

### 1.1.2 Techniques to Modify Linear Block Codes

One of the methods to design codes with desired properties is to start from a good code and modify its parameters. There are several methods to modify a linear block codes; however, we describe four techniques that are more important in our work.

**Extending**

In extending it is usually assumed that $k$ is fixed and $n$ is increased, so $m$ is increased by the same amount as $n$ increases. As a result, rate always decreases and the minimum distance usually increases resulting in a better performance.

**Lengthening**

If we increase the code length, $n$, without increasing the number of parity-checks $m$, we call the new code a lengthened code. As a result of lengthening, the number of information bits $k$ is increased by the same amount and the rate $R$ is increased. The minimum distance of the lengthened code cannot exceed that of the original code.

**Expurgation**

Expurgation is another variant of the extending technique. If we increase the number of parity-checks $m$ without increasing the code length $n$, the number of information bits $k$ decreases. The result is called an expurgated code which has lower rate and a minimum distance at least equal to the minimum distance of the original code.

**Puncturing**

In puncturing, the number of information bits $k$ stays unchanged, while some bits of the codeword are deleted (reduced $n$). As a result, we get a higher rate code with minimum
distance at most equal to that of the original code.

1.1.3 LDPC Codes

LDPC codes are a class of linear block codes with very *sparse* parity-check matrices. More precisely, an \( m \times n \)-parity check matrix \( H \) is called \( c \)-sparse if \( mn \) tends to infinity and the number of nonzero elements in \( H \) is always less than \( c \max(m, n) \) [26]. An LDPC code can be represented by a bipartite graph that is also known as a Tanner graph [27]. Figure 1.1 represents the Tanner graph of an \((n, n - m)\) generic linear block code where the lower nodes represent the \( n \) encoded bits that are called bit nodes or variable codes and the upper nodes represent the \( m \) parity-check constraints and are called constraint nodes or check nodes. These two sets of nodes are connected through a set of *edges*. The degree of a node is the number of edges connected to it. An LDPC code is called *regular* when all its variable nodes have same degree \( d_v \) and all its check nodes have same degree \( d_c \). Different variable and check node degrees are allowed in the structure of an *irregular* LDPC code, which is characterized by a pair of degree distributions \((\lambda(x), \rho(x))\) defined as

\[
\lambda(x) = \sum_i \lambda_i x^{i-1} \quad \text{and} \quad \rho(x) = \sum_j \rho_j x^{j-1}
\]

where \( \lambda_i \) and \( \rho_j \) denote the fraction of edges of degrees \( i \) and \( j \) connected to variable and check nodes, respectively. For simplicity, hereafter we use \( \lambda \) for \( \lambda(x) \) and \( \rho \) for \( \rho(x) \). Given the degree distributions of an irregular LDPC code, the nominal rate \( R \) of the code is given by [28]

\[
R = 1 - \frac{\sum_j \rho_j / j}{\sum_i \lambda_i / i}
\]

(1.2)

1.1.4 The Sum-Product Decoding Algorithm

A well-known algorithm to decode LDPC codes is the message-passing based sum-product algorithm (SPA) [29]. This algorithm begins with message initialization at the variable nodes (as estimations of the transmitted bits based on the channel outputs) and continues with iterative exchange of the *extrinsic* messages between variable and check nodes. Each iteration of the SPA starts with updating the messages at the check nodes and ends
with updating the messages at the variable nodes using their corresponding update rules. At the end of each iteration, a bit by bit decision is made to detect a binary vector. The algorithm stops if the detected binary vector is a valid codeword or if the number of iterations achieves a preset maximum, where a detection failure is declared. The steps of the SPA can be summarized as follows [30]:

1. **Initialization.** The initial messages at the variable nodes are assumed to be the log-likelihood ratios (LLRs) of the messages received from the channel. If we denote the transmitted codeword by \( c = [c_1, c_2, \cdots, c_n] \), the LLR values for the binary-input (BPSK) additive white Gaussian noise (BI-AWGN) channel are given by

\[
L(c_i) = \frac{2y_i}{\sigma^2}, \quad i = 1, 2, \cdots, n \quad (1.3)
\]

where \( y_i \) is the \( i \)-th received bit from the channel and \( \sigma^2 \) is the variance of the channel noise.

The LLR of the \( i \)-th received bit in the case of Rayleigh fading channels with channel state known at the receiver is given by

\[
L(c_i) = \frac{2y_i}{\sigma^2} a_i, \quad i = 1, 2, \cdots, n \quad (1.4)
\]

where \( a_i \) is the Rayleigh distributed channel coefficient with \( \text{E}[a_i^2] = 1 \).

2. **Check node updates.** The messages from each check node to its neighboring variable nodes are updated. The message \( m_{c \rightarrow v} \) from check node \( c \) to variable node \( v \) is updated
according to the following update rule

\[ m_{c \rightarrow v} = 2 \tanh^{-1} \left( \prod_{v' \in \{n(c) \setminus v\}} \tanh \left( \frac{m_{v' \rightarrow c}}{2} \right) \right) \]  

(1.5)

where \( m_{v' \rightarrow c} \) denotes the message from variable node \( v' \) to check node \( c \) and \( \{n(c) \setminus v\} \) is the set of variable nodes that are connected to check node \( c \) excluding variable node \( v \).

3. **Variable node updates.** The messages are updated at the variable nodes according to the following update rule

\[ m_{v \rightarrow c} = m_0 + \sum_{c' \in \{n(v) \setminus c\}} m_{c' \rightarrow v} \]  

(1.6)

where \( m_0 \) is the initial message at variable node \( v \) and \( \{n(v) \setminus c\} \) is the set of check nodes connected to variable node \( v \) excluding check node \( c \).

4. **Detection.** At each variable node \( v \), if \( m_0 + \sum_{c' \in n(v) \setminus c} m_{c' \rightarrow v} < 0 \), it is declared that a “1” is transmitted; otherwise, it is declared that a “0” is transmitted.

5. **Syndrome check.** A check on the validity of the detected codeword \( c \) (through verifying whether \( cH^t = 0 \)) or the number of iterations is performed here. Stop if either one passes. Otherwise go back to step 2.

### 1.1.5 Density Evolution

For many interesting channels including the AWGN channel, it is possible to implement an algorithm called *density evolution* (DE) and calculate a *noise threshold* value for an ensemble of randomly constructed LDPC codes of a certain rate under which reliable communication is possible by deploying sufficiently long codewords and performing an iterative message passing decoding algorithm such as the sum-product algorithm. The threshold value is equivalently associated with the lowest value of the signal-to-noise ratio required for reliable communication which is called the *convergence threshold* and is denoted by \( \text{SNR}_{\text{threshold}} \). In the former case, one is interested to find the “worst” channel for a fixed rate, whereas in the latter case, the channel SNR is fixed and we look for the highest rate code.
DE studies the evolution of the message densities in the iterative message passing decoding of LDPC codes where without loss of generality, it is assumed that the all-one codeword was transmitted and that the log-likelihood ratios (LLRs) are used to represent the messages. Under the iterative message passing algorithm, DE is performed in two-step iterations and is initialized by equating the message density at the variable nodes with the density of the message LLRs, $p_c$, received from the channel. Each iteration begins with updating the message densities at the check nodes and ends with updating them at the variable nodes. If we denote the message density at the input of a degree-$j$ check node by $p$, under the sum-product message-passing algorithm, the message density at its output, denoted by $p'_j$, can be calculated by

$$p'_j = R^{j-1}p,$$  \hspace{1cm} (1.7)

where $R$ is the check node density update rule defined in [31]. In this notation, it is assumed that $R(p,p) = R^2p$ and as a result, we can write $R(p, R(p, \ldots, R(p, p), \ldots)) = R^{j-1}p$.

Additionally, if we denote the message density at the input of a degree-$i$ variable node by $p'$, the message density at its output, denoted by $p_i$, can be calculated by

$$p_i = \otimes^{i-1}p' \otimes p_c,$$  \hspace{1cm} (1.8)

where $\otimes^i$ is $i$-th order convolution. (See [31–33] for more details.)

The update rules given in (1.5) and (1.6) constitute the basis for a two-step DE technique. As a result, it is possible to study the evolution of the density of the message LLRs received from the channel at a specific SNR and the pair of degree distributions of the code and to determine whether the probability of error approaches zero (or a target value) after a specific number of iterations.

The method described above requires calculating the probability of error at each iteration which is normally performed after the variable node updates. Because the channels and the decoding algorithm are symmetric [32], the negative tail of the density represents the probability of error. If we denote by $e(\cdot)$ the function that calculates the probability of error for a given distribution and by $p'_i$ the message density at the outputs of degree-$i$ variable nodes at the $l$-th iteration, the overall message error probability after variable node
updates can be calculated by

\[ e(p') = \sum_{i} \lambda_i e(p'_i), \]  

(1.9)

where \( p' \) is the message density at the output of the variable nodes at iteration \( l \) [12].

Eq. (1.9) suggests that \( e(p') \) is a linear function of \( \lambda_i \)'s. However, due to the dependence of \( e(p'_i) \)'s on \( \lambda \), this relation is not linear. Nevertheless, if changes in \( \lambda \) are small, we can assume that \( e(p') \) is linearly dependent on \( \lambda \). As a result, a linear programming problem to design the highest rate LDPC code for a given SNR threshold can be formulated as

maximize \( \sum_{i} \lambda_i / i \)

subject to \( \sum_{i} \lambda_i e(p'_i) < \mu^{(h)} e(p'), \quad l = 1, 2, \ldots, L \)

\( \sum_{i} \lambda_i = 1, \)  

(1.10)

where \( L \) is the maximum number of iterations.

From Eq. (1.2), it is clear that maximizing the objective function above maximizes the rate of the designed LDPC code. In this formulation, \( h \) is the optimization iteration number, \( \mu \) is a positive coefficient which slowly increases towards one and determines the maximum value of changes in \( \lambda \) and \( L \) is the maximum allowed number of iterations in the iterative decoding algorithm. The inequality constraint ensures that using the updated \( \lambda \), the symbol error probability decreases at each decoding iteration.

### 1.1.6 Extrinsic information transfer chart analysis

Extrinsic information transfer (EXIT) chart is another tool to analyze iterative decoders where instead of tracking the evolution of the messages densities, the evolution of a single parameter corresponding to the messages is studied.

Initially, mutual information between the messages and the decoded bits was used in EXIT charts [34, 35]. However, later works employed other parameters in their analysis, including the SNR of the extrinsic information [36, 37] or the error probability of the message densities [38].
In this work, we are interested in tracking the evolution of the error probability of the message densities. This analysis is performed by defining a function $f$ which can compute the error probability at the output of the variable nodes at each iteration. The error probability is a function of the pair of degree distributions $(\lambda, \rho)$, as well as the error probability of the message densities at the output of the variable nodes at the previous iteration and that of the messages received from the channel. Generally, we are interested to determine the EXIT chart of an irregular LDPC code assuming a fixed SNR (i.e., a fixed probability of error) and a fixed pair of degree distributions. If we define $e^{(l)} \triangleq e(p^l)$, the EXIT chart $f$ is expressed as a function of $e^{(l)}$ only, i.e.,

$$e^{(l+1)} = f(e^{(l)}), \quad l = 1, \ldots, L.$$  \hfill (1.11)

resulting in the $L$ pairs $(e^{(1)}, f(e^{(1)})), \ldots, (e^{(L)}, f(e^{(L)}))$. Thus, we can obtain the function $f$ by interpolating a curve passing through all these pairs as well as the origin. Then, we can associate a pair $(e, f(e))$ to any point on the interpolated curve $f$.

One efficient method to obtain $f$ is by finding the so-called elementary EXIT chart

![EXIT Chart](image.png)

**Figure 1.2: EXIT Chart.**
associated with degree-\(i\) variable nodes for \(i = 2, ..., d_v,\text{max}\) and to find the overall EXIT chart as a linear combination of elementary EXIT charts. This can be done by setting

\[
f(e) = \sum_{i} \lambda_i f_i(e), \quad l = 1, .., L.
\]

(1.12)

where \(f_i(e)\) is the elementary EXIT chart associated with degree-\(i\) variable nodes.

In order to visualize the decoding behavior of an LDPC code using its EXIT chart, both \(f\) and its inverse \(f^{-1}\) are plotted on the same figure as shown in Figure 1.2. This representation is particularly convenient since the output of \(f\) at each iteration serves as its input in the next iteration. The initial probability of error —at which the process starts— belongs to the error probability of the messages received from the channel, which we denote by \(e_0\) in this work. Moreover, the process stops when the error probability of the message densities achieves a target error probability \(e_t\) within the maximum number of iterations \(L\) or when we have reached \(L\) without a success in achieving \(e_t\) in which case it is announced that the code does not converge.

As it can be implied from Figure 2.1, the shape of the EXIT chart determines the convergence behavior of the code, i.e., whether the iterative decoding converges and the required number of iterations for convergence, denoted by \(N\). We can observe a tunnel between function \(f\) and the line \(e^{(l)} = e^{(l+1)}\). If the tunnel is closed, —i.e., if \(f\) crosses the line \(e^{(l)} = e^{(l+1)}\)— the decoder does not converge. In fact, the convergence threshold of the code is the worst channel condition for which the tunnel is open. For more details on EXIT charts, please refer to [34, 38, 39].

1.2 The Relay Channel

The classical relay channel model consists of a source node (S), a relay node (R), and a destination node (D) as shown in Figure 2.1. The existence of the relay node makes it possible to introduce a cooperative scheme in order to achieve a higher data rate for transmission of information from source to destination.

The initial coding schemes for the relay channel were proposed by Cover and El Gamal where they introduced two strategies for relaying [4]. The first technique, called decode-and-forward (DF), is a block-transmission strategy. In the DF technique, the relay decodes the
received signal in one block and transmits an encoded version of the decoded message to the destination in the following block. The second technique, often called compress-and-forward (CF), is also a block-transmission strategy. In this technique, the relay does quantizes the received signal (instead of decoding it) and forwards a compressed or scaled version of the quantized signal to the destination.

Another well-known relaying strategy is called Amplify-and-Forward (AF) where the relay amplifies the received signal, subject to its power constraint, and forwards the amplified signal to the destination. Although the AF technique results in less delay compared to the DF and CF techniques, the DF technique has the advantage that it does not forward the received noise to the destination and therefore can achieve higher rates compared with the other two techniques (see [40]). The disadvantage of the DF technique is that it requires decoding at the relay resulting in increased complexity and delay. Reducing the decoding complexity and delay at the relay when the DF technique is employed is one of our goals in this work.

Since we focus on DF in the development of our coding strategies, this technique is described in more detail in the following section.

Figure 1.3: The relay channel model.
1.2.1 Decode-and-Forward Relaying Strategy

The decode-and-forward (DF) strategy was originally suggested in [4] using binning and block Markov superposition coding techniques where the transmissions are performed in successive blocks and implemented as follows: The source node encodes a new message and transmits the encoded message to both relay and destination. The rate of the encoded data is higher than the capacity of the source-destination (SD) link but lower than the capacity of the source-relay (SR) link. The destination stores the received message while the relay node decodes the message and thus knows an index called the *bin index* by which the source message becomes restricted to a smaller set. In the next block, the relay encodes the bin index of the source message and transmits it to the destination, while the source transmits a signal which is the *superposition* of the bin index of its previous message and the new message. At the receiver, the destination first decodes the bin index received from the both source and relay. Then it decodes the source’s previous message using its bin index, while storing its new message.

In [4], Cover and El Gamal also showed that the block Markov superposition DF technique achieves the capacity for the class of physically degraded relay channels. By a physically degraded relay channel we mean the case where the received signal at the destination is a degraded (noisy) version of the received signal at the relay. An example of this case is the Gaussian degraded relay channel where the additive white Gaussian noises \( Z_1 \sim N(0, N_1) \) and \( Z_2 \sim N(0, N_1 + N_2) \) corrupt the transmitted signal at the relay and destination, respectively. If we denote the transmitted signal at the source by \( X \) and the transmitted signal at the relay by \( X_1 \) and the channel outputs at the relay and destination by \( Y_1 \) and \( Y \), respectively, we have

\[
Y_1 = X + Z_1 \quad \text{(1.13)}
\]

\[
Y = X + X_1 + Z_2 \quad \text{(1.14)}
\]

Let us assume that the transmitted powers at the source and relay are denoted by \( P \) and \( P_1 \), respectively. We assume the source divides its power \( P \) into two parts, a fraction \( \alpha P \) is used to transmit the new message and a fraction \( (1 - \alpha)P \) for cooperatively transmitting the bin index of the previous message. It is shown in [4] that for the Gaussian degraded
relay channel model described above, the overall DF rate $R$ is given by

$$R = \max_{\alpha} \min \left\{ \frac{1}{2} \log \left( 1 + \frac{\alpha P}{N_1} \right), \frac{1}{2} \log \left( 1 + \frac{P + P_1 + 2\sqrt{\alpha P P_1}}{N_1 + N_2} \right) \right\}$$

(1.15)

where $\bar{\alpha} = 1 - \alpha$. Moreover, the effective SNR of the SD and SR links are given by

$$\text{SNR}_{SR} = \frac{\alpha P}{N_1}$$

(1.16)

$$\text{SNR}_{SD} = \frac{\alpha P}{N_1 + N_2}$$

(1.17)

and the effective rates corresponding to these SNRs can be calculated by

$$R_{SR} = \frac{1}{2} \log (1 + \text{SNR}_{SR})$$

(1.18)

$$R_{SD} = \frac{1}{2} \log (1 + \text{SNR}_{SD})$$

(1.19)

The practical implementation of binning implies designing two different codes; a high rate relay code with rate $R_{SR}$ and a low rate destination code with rate $R_{SD}$ which is decodable at the destination only after its bin index is transmitted by the relay. Thus the overall DF rate can be written as

$$R = \min \{R_{SR}, R_1 + R_{SD}\}$$

(1.20)

where $R_1 = I(X_1; Y)$.

### 1.2.2 A Two-User Cooperation Scheme

In [1], Sendonaris, et al. proposed a more general scheme for user cooperation (Figure 1.4), where each transmitter has its own message to transmit and the users cooperate to transmit their corresponding messages. As shown in this figure, the transmitted signal by user 1 is denoted by $X_1$ and is the encoded version of the summation of user 1’s message $W_1$ and the signal received from user 2 in the previous block, denoted by $Y_1$. Similar principles and notations apply to user 2. The destination receives the sum of the signals transmitted by both users. The input-output relations can be summarized as

$$Y_0(t) = K_{10}X_1(t) + K_{20}X_2(t) + Z_0(t)$$

(1.21)

$$Y_1(t) = K_{21}X_2(t) + Z_1(t)$$

(1.22)

$$Y_2(t) = K_{12}X_1(t) + Z_2(t)$$

(1.23)
where \( K_{i0}, i = 1, 2, \) denotes the fading coefficient from user \( i \) to the destination for and \( K_{ij}, j = 1, 2, j \neq i \) denotes the fading coefficient from user \( i \) to user \( j \). Additive channel noises \( Z_0, Z_1, \) and \( Z_2 \) are assumed to be independent.

### 1.3 Bilayer LDPC Codes

A practical method to design capacity approaching codes for the relay and destination which employs binning was proposed by Razaghi and Yu in [11], where they proposed two classes of the so-called bilayer LDPC codes; the bilayer lengthened and the bilayer expurgated LDPC codes. Each of these codes consists of a high-rate relay code (decodable at SNR\(_{\text{SR}}\)) and a low-rate destination code (decodable at SNR\(_{\text{SD}}\)). We study their structure and the corresponding encoding/decoding schemes for relay channels below.

#### 1.3.1 Bilayer Lengthened LDPC Codes

The general form of the Tanner graph for a bilayer lengthened LDPC code is shown in Figure 2.2 which consists of two sets of variable nodes belonging to the lower and upper layer graphs and a set of check nodes that is in common between both layers. The set of
edges that connect the lower variable nodes to the check nodes are characterized by the degree distribution pair \((\lambda^{(1)}, \rho^{(1)})\) and the set of edges connecting the upper variable nodes to the check nodes are characterized by the degree distribution pair \((\lambda^{(2)}, \rho^{(2)})\). Throughout this work, we assume that all check nodes of the lower and upper graphs are of degrees \(d_c\) and \(d'_c\), respectively. Hence, the ratio of the number of edges in the lower graph to the total number of edges in the bilayer graph, denoted by \(\eta\), is \(\eta = d_c/(d_c + d'_c)\).

The Tanner graph of a bilayer lengthened LDPC code represents two different codes (see Figure 2.2.) The first code which is represented by the lower graph has \(n_1\) variable nodes and \(m\) parity check nodes and has rate \(R_1 = 1 - m/n_1\). The second code is the result of lengthening the first code by adding \(n_2\) extra variable nodes to the first code —while keeping \(m\) constant— and has rate \(R_{BL} = 1 - m/(n_1 + n_2)\); hence, \(R_{BL} > R_1\). Let all check nodes have degree \(d_c\) in the lower graph and degree \(d'_c\) in the upper graph, the rate \(R_{BL}\) of BL-LDPC codes is given by

\[
R_{BL} = 1 - \frac{m}{n_1 + n_2} \tag{1.24}
= 1 - \frac{1}{d_c \sum_i \lambda_i^{(1)}/i + d'_c \sum_i \lambda_i^{(2)}/i} \tag{1.25}
\]

If we assume that the source-destination (SD) link has a lower effective SNR in comparison with the source-relay (SR) link, then the low rate and the high rate codes can be optimized considering the effective SNRs of the SD and SR links, respectively.

In order to employ BL-LDPC codes in the relay channel under the DF strategy, the source encodes its information using the high rate code and transmits the encoded message. Although both the relay and the destination receive the transmitted message, only the relay can decode it since the high rate code is optimized for the relay node. In the next step, the relay encodes the values of the upper variable nodes and transmits the encoded message at a rate which is decodable at the destination. Only after decoding the values of the upper variable nodes transmitted by the relay, the destination can decode the source’s message. We will elaborate more on DF relaying using BL-LDPC codes in Chapter 3.
1.3.2 Bilayer Expurgated LDPC Codes

In order to understand the structure of bilayer expurgated LDPC codes, we first describe how the relay cooperates with the source in this scheme by forwarding the extra parity bits to the destination. Assume that the source encodes its message using the high rate code optimized for SNR$_{SR}$. Similar to the case described before, the destination stores the received message. The relay, however, decodes the received signal and based on the decoded source message produces a number of extra presumably nonzero parity bits and transmits them to the destination. The destination decodes the relay’s extra parity bits first. It then decodes the received signal from the source using the extra information it has gained about the source message.

The relaying strategy described above forms the basis for the structure of bilayer expurgated LDPC codes. In contrast to bilayer lengthened LDPC codes, bilayer expurgated LDPC codes (BE-LDPC) consist of two sets of check nodes and one set of variable nodes which is shared between the lower and the upper graphs (Figure 1.6). The lower and upper check nodes are characterized by degree distributions $\rho^{(1)}$ and $\rho^{(2)}$, respectively, and the variable nodes are characterized by $\lambda_{i,j}$ which defines the fraction of the edges that are connected to degree-$i$ lower and degree-$j$ upper variable nodes. If we assume that all check...
nodes have degree $d_c$ in the lower graph and degree $d'_c$ in the upper graph, we obtain

$$\eta = \frac{d_c k_1}{d_c k_1 + d'_c k_2}$$

Moreover, the marginal lower and upper variable node degree distributions $\lambda_i$ and $\lambda_j$ are given by

$$\lambda_i = \frac{1}{\eta} \sum_j \frac{i}{i+j} \lambda_{i,j} \quad \text{and} \quad \lambda_j = \frac{1}{\eta} \sum_i \frac{j}{i+j} \lambda_{i,j}. \tag{1.27}$$

If we assume that there are $m_1$ lower check nodes, $m_2$ upper check nodes, and $n$ variable nodes in the graph, the rate of the BE-LDPC code is given by

$$R_{BL} = 1 - \frac{m_1 + m_2}{n} \tag{1.28}$$

$$= 1 - \frac{1}{d_c \sum_i \frac{\lambda_i^{(1)}}{i} + d'_c \sum_i \frac{\lambda_i^{(2)}}{i}} \tag{1.29}$$

The structure of bilayer LDPC codes described above results in designing very high rate codes for both relay and destination nodes that are optimized for the effective thresholds of the corresponding links and hence, the decoding complexity at both nodes is high. In practice, however, there are usually strict hardware and power constraints, particularly at the relay. Furthermore, decoding delay at the relay results in a higher decoding delay at the destination. Due to the importance of these practical issues, one of our goals in this dissertation is to reduce the decoding complexity of bilayer lengthened codes where we propose efficient methods to design complexity-optimized codes.

### 1.4 Protograph-Based LDPC Codes

A *protograph* is a Tanner graph with relatively small number of nodes. A $G = (V, C, E)$ protograph consists of a set of variable nodes $V$, a set of check nodes $C$, and a set of edges $E$. Each edge $e_{ij} \in E$ connects variable node $v_i \in V$ to check node $c_j \in C$. An example of such a graph is shown in Figure 1.7(a). A protograph may alternatively be called a *base graph* or a *projected* graph and is equivalent to a linear block code and since its degree distributions are known, we can analyze its performance in the asymptotic case, (i.e., when $n \to \infty$) using both density evolution and EXIT chart techniques.
A class of structured LDPC codes is introduced in [25] where a small protograph is used to construct a larger graph with desired properties. The process to generate the larger graph is termed \textit{lifting}. The result is often called a \textit{lifted} or \textit{derived} graph which represents a protograph-based LDPC (PB-LDPC) code. The structure imposed on these codes can reduce implementation cost, enable parallel architecture of the decoder and simplify encoding.
In order to construct a protograph-based code, the base protograph is lifted in two steps. In the beginning, \( L \) copies of the base graph are made (see Figure 1.7(b)), where \( L \) is called the lifting factor and is determined by the desired codeword length \( n \) since for a given protograph of length \( N \), the codeword length is given by \( n = N \times L \). Protograph-based LDPC codes are a special class of multi-edge type LDPC codes [41] where each edge in the protograph represents an edge type. Therefore, all copies of one edge type form a cluster of edges. The next step is to permute edges inside each cluster, i.e., all edges of the same type, and thus, to interconnect the copies that are made (see Figure 1.7(c)).

There are several methods to permute edges in a protograph. For example, random permutations may be used to permute different edge clusters. The disadvantage of using random permutations is that its hardware implementation is difficult. However, techniques to permute edges exist that are very suitable for hardware implementation. For instance, one may choose permutations that are supported by an \( \Omega \)-network. This technique is outside the scope of our work. For more information on \( \Omega \)-networks, see [42–44].

Another technique that is very suitable for hardware implementation is to choose permutations from circulants [45]. Protograph codes constructed from circulants are categorized as quasi-cyclic LDPC codes and several efficient techniques have been proposed to encode these codes (see, for instance, [46, 47]).

### 1.5 The Organization of this Dissertation

In Chapter 2, we propose a technique to design complexity-optimized bilayer lengthened LDPC codes for Gaussian relay channels. Our technique fixes the rate-optimized destination code and targets minimizing the decoding complexity of the relay code by minimizing a function of the upper variable node degree distribution, which approximates \( N \), the number of iterations at the relay node, while retaining the rate of the bilayer code. This is done by generalizing EXIT charts to the case of bilayer lengthened LDPC codes and estimating the number of decoding iterations using the EXIT chart of an irregular LDPC code. Using this technique, we have designed bilayer codes with noticeable reduction in decoding complexity and delay compared to the rate-optimized codes reported in the literature.

Moreover, we design an extended class of complexity-optimized bilayer lengthened
low-density parity-check (LDPC) codes by considering upper check nodes with different degrees. We show that by implementing a low complexity relay decoder, our codes outperform the rate-optimized codes in a wide range of signal to noise ratios.

In Chapter 3, we address the code design problem for Rayleigh fading relay channels. We first calculate the capacity of decode-and-forward relaying in these channels. Then, we design bilayer lengthened LDPC codes for this channel model in different SNR regimes, namely, when the link between the source and the relay is much stronger than the link between the source and the destination and when the SNRs of these links are comparable.

In Chapter 4, we propose a user cooperation strategy based on protograph-based LDPC codes and the Alamouti space-time coding scheme. The proposed scheme is flexible with employing protograph codes of various lengths and rates.

We address the code design problem for coded cooperation under slow fading channels in Chapter 5 where we propose two methods to construct channel codes for this channel model. Our techniques are based on protograph-based designs and offer extensive rate-compatibility. Furthermore, we derive the BPSK outage probability limit for our cooperative scheme and show that the code we design achieve full-diversity and approach the outage probability limit.

We conclude this dissertation in Chapter 6 with suggestions for future work.
Chapter 2

Complexity-Optimized Bilayer Lengthened LDPC Codes for Relay Channels

2.1 Introduction

Relaying is one of the most widely used techniques in cooperative communication. A three node relay channel consists of a source node (S), a relay node (R), and a destination node (D) [1], where the relay assists the destination by forwarding to it a function of the information that it receives from the source.

As we noted in Chapter 1, one of the coding techniques for the relay channel is called decode-and-forward (DF) which is a block-transmission strategy. In the DF technique, the relay decodes the received signal in one block and transmits an encoded version of the decoded message to the destination in the following block.

The DF technique has the advantage that it does not forward the received noise to the destination and therefore can achieve higher rates compared with the other relaying techniques (see [40]). However, the disadvantage of the DF technique is that it requires decoding at the relay resulting in increased complexity and delay. Reducing the decoding complexity and delay at the relay when the DF technique is employed is the focus of this chapter. We propose a technique to minimize the decoding complexity at the relay subject
to a number of design constraints.

One of the techniques used to deploy the DF strategy is to employ bilayer lengthened LDPC (BL-LDPC) codes [12]. A BL-LDPC code consists of one high rate code optimized for the source-relay link and a low rate code optimized for the source-destination link. The high rate code is constructed to be decodable at the relay while the low rate code is designed to be decodable at the destination with the help of the additional information provided by the relay.

In bilayer lengthened LDPC codes, the low rate code is designed first and then the high rate code is obtained from it by keeping the total number of parity check nodes \(m\) constant and increasing the length of the code \(n\) while keeping the low rate code unchanged.

Application of these techniques results in high rate codes for the relay channel. However, decoder complexity and delay constraints, which play an important role in practice, are not addressed in the design process. Decreasing decoding complexity at the relay is particularly important since there are usually stricter hardware and power constraints at the relay and also because decoding delay at the relay increases the decoding delay at the destination.

Inspired by [52, 53], we propose an approach to design complexity-optimized bilayer lengthened LDPC codes by utilizing extrinsic-information transfer (EXIT) charts of BL-LDPC codes. For this purpose, we extend the notion of EXIT charts for BL-LDPC codes — calculated using the bilayer density evolution method — and then shape the overall EXIT chart of the bilayer code such that the number of the decoding iterations is minimized while retaining the rate of the rate-optimized bilayer code. We also formulate a measure of decoding complexity for these codes and discuss how the overall EXIT chart of the bilayer code plays a role in approximating the decoding complexity. Our technique results in designing bilayer lengthened LDPC codes with dramatic reduction in decoding complexity compared to the rate-optimized codes.

The rest of this chapter is organized as follows. In Section 2.2, we briefly review the coding problem for the relay channel, as well as the structure of BL-LDPC codes and the role of bilayer density evolution in determining the error probability of these codes. In Section 2.3, we examine and extend the notion of overall EXIT charts to the BL-LDPC codes. We formulate a measure of complexity for these codes in Section 2.4 and then show how the
Figure 2.1: The relay channel model.

overall EXIT chart of the bilayer code can play a role in estimating the decoding complexity. In Section 2.5, we apply our method to design complexity-optimized BL-LDPC codes and present our results in Sections 2.6 and 2.7. We conclude this chapter in Section 2.8.

2.2 Preliminaries

2.2.1 The Relay Channel

The classic relay channel model consists of a source node (S), a relay node (R), and a destination node (D) as shown in Figure 2.1. The role of the relay node is to make it possible to achieve a higher data rate for transmission of information from the source to the destination.

The DF strategy, originally suggested in [4], uses binning and block Markov superposition coding techniques to transmit information in successive blocks. The source node encodes the message and transmits the encoded message to both relay and destination. It also randomly assigns codewords to a certain number of bins. The rate of the encoded data is higher than the capacity of the source-destination (SD) link but lower than the capacity of the source-relay (SR) link. The destination, therefore, cannot decode the message and can only come up with a list of candidate messages. The relay, however, decodes the message and thus knows both the message and the bin index. In the next block, the transmitter and the relay, both knowing the bin index of the previous block, cooperate to transmit this in-
formation to the destination while the transmitter transmits fresh information as well using superposition coding. At the end of this block, the receiver first decodes the bin number of the previous block that was cooperatively sent to him by the transmitter and the relay. Then it uses the bin number and the list that it had decoded at the end of the previous block to finalize the decoding of the message in the previous block.

In [4], Cover and El Gamal showed that the block Markov superposition technique described above achieves the capacity for the class of physically degraded relay channels which includes, for instance, the Gaussian degraded relay channel where the additive white Gaussian noises $Z_1 \sim N(0, N_1)$ and $Z_2 \sim N(0, N_1 + N_2)$ corrupt the transmitted signal at the relay and destination, respectively. If we denote the transmitted signals at the source and relay by $X$ and $X_1$ and the channel outputs at the relay and destination by $Y_1$ and $Y$, respectively, we have

$$Y_1 = X + Z_1$$  \hspace{1cm} (2.1)

$$Y = X + X_1 + Z_2$$  \hspace{1cm} (2.2)

Let us assume that the transmitted powers at the source and destination are denoted by $P$ and $P_1$, respectively. We assume the source divides its power $P$ into two parts, a fraction $\alpha P$ is used to transmit the new message and, the remaining power $(1 - \alpha)P$ is devoted to cooperative transmission of the bin index of the previous message. It is shown in [4] that for the Gaussian degraded relay channel model described above, the overall DF rate $R$ is given by

$$R = \max_\alpha \min \left\{ \frac{1}{2} \log \left(1 + \frac{\alpha P}{N_1}\right), \frac{1}{2} \log \left(1 + \frac{P + P_1 + 2\sqrt{\alpha P P_1}}{N_1 + N_2}\right) \right\}$$  \hspace{1cm} (2.3)

where $\bar{\alpha} = 1 - \alpha$. Moreover, the effective SNR of the SD and SR links are given by

$$\text{SNR}_{SR} = \frac{\alpha P}{N_1}$$  \hspace{1cm} (2.4)

$$\text{SNR}_{SD} = \frac{\alpha P}{N_1 + N_2}$$  \hspace{1cm} (2.5)

and the effective rates corresponding to these SNRs are given by

$$R_{SR} = \frac{1}{2} \log (1 + \text{SNR}_{SR})$$  \hspace{1cm} (2.6)

$$R_{SD} = \frac{1}{2} \log (1 + \text{SNR}_{SD})$$  \hspace{1cm} (2.7)
The practical implementation of binning implies designing two different codes; a relay code with rate $R_{SR}$ and a destination code with rate $R_{SD}$ which is decodable at the destination only after its bin index is transmitted by the relay. Thus, for a given $\alpha$, the overall DF rate can be written as

$$R = \min \{ R_{SR}, R_1 + R_{SD} \}$$

where $R_1 = I(X_1; Y) = \frac{1}{2} \log \left( 1 + \left( \frac{\sqrt{\alpha P} + \sqrt{P_1}}{\alpha P + N_1 + N_2} \right)^2 \right)$.

### 2.2.2 Bilayer Lengthened LDPC Codes

We study the structure of the class of bilayer lengthened LDPC codes in this section. A conventional LDPC code consists of a set of variable nodes, a set of check nodes and a set of edges that connect variable nodes and check nodes. These edges can be characterized by a pair of degree distributions $(\lambda(x), \rho(x))$ defined as

$$\lambda(x) = \sum_i \lambda_i x^{i-1} \quad \text{and} \quad \rho(x) = \sum_j \rho_j x^{j-1}$$

where $\lambda_i$ and $\rho_j$ describe the fraction of edges connected to variable and check nodes of degrees $i$ and $j$, respectively. For simplicity, hereafter, we use the notation $\lambda$ for $\lambda(x)$ and $\rho$ for $\rho(x)$.

As stated in [11], a bilayer lengthened LDPC code consists of two sets of variable nodes and a set of check nodes (see Figure 2.2.) Note that the set of check nodes is shared between the lower and upper layer graphs. The set of edges that connect the lower variable nodes to the check nodes are characterized by the degree distribution pair $(\lambda^{(1)}, \rho^{(1)})$ and the set of edges connecting the upper variable nodes to the check nodes are characterized by the degree distribution pair $(\lambda^{(2)}, \rho^{(2)})$. Let the terms “lower degree” and “upper degree” be used to denote the degrees of the check nodes in the lower and upper graphs, respectively. Further, throughout this work, we assume that all check nodes have lower degree $d_c$ and upper degree $d'_c$. Hence, the ratio of the number of edges in the lower graph to the total number of edges in the bilayer graph, denoted by $\eta$, is $\eta = d_c/(d_c + d'_c)$.

The Tanner graph of a bilayer lengthened LDPC code represents two different codes. The first code, represented by the lower graph, has $n_1$ variable nodes and $m$ parity check
nodes and represents the low rate code used for communicating with the destination. The second code is constructed by lengthening the first code through adding $n_2$ extra variable nodes—while keeping $m$ constant—and hence, creating a high rate code used for transmission to the relay node and represented by the overall graph.\footnote{We assume that the source-destination link has an SNR lower than the SNR of the source-relay link and as a result, the low and high rate codes are optimized for $\text{SNR}_{\text{SD}}$ and $\text{SNR}_{\text{SR}}$, respectively.} The rate of the bilayer lengthened LDPC code characterized above is

$$R_{BL} = 1 - \frac{m}{n_1 + n_2} = 1 - \frac{1}{d_c \sum_i \lambda_i^{(1)}/i + d'_c \sum_i \lambda_i^{(2)}/i}$$

(2.9)

The bilayer structure of these codes provides an elegant framework which is highly adaptable to the block Markov coding technique, thus, making these codes very promising for relaying based on the DF strategy.

### 2.2.3 Relaying Strategy Based on Bilayer Lengthened LDPC Codes

To deploy the relaying technique based on bilayer lengthened LDPC codes, the source encodes its data using a high-rate LDPC code (characterized by the overall graph) and then transmits the encoded information. The destination receives and stores the channel output.
Since the rate of the code is higher than the capacity of the source-destination link, the receiver at the destination cannot yet decode the received signal. However, the relay decodes the received sequence. The relay then uses another LDPC code to encode the upper variable nodes and sends the result to the destination. This LDPC code is designed such that the destination is capable of decoding the upper variable nodes. After decoding the relay’s message, the destination uses the values of the upper variable nodes, together with the channel output that it had saved in the previous interval, to decode the entire message.

### 2.2.4 Bilayer Density Evolution

Density Evolution (DE) has been traditionally used to design LDPC codes and analyze their asymptotic performance under iterative message passing decoding algorithm. An extension to this technique for bilayer LDPC codes, called bilayer DE, is developed in [6].

Iterative decoding algorithm works by successively sending messages from variable nodes to check nodes and vice versa. The messages at both variable nodes and check nodes are updated according to certain rules. This algorithm is initialized by log-likelihood ratio (LLR) values of the messages received from the channel and continues by iteratively updating these messages at both sets of nodes. A decoding iteration of an iterative decoding scheme begins with updating the messages in the check nodes followed by updates at the lower and upper variable nodes [30]. By using the DE technique, it is possible to study the evolution of the probability density function of the messages in the iterative message passing decoding of LDPC codes. Considering the fact that the channels and the decoding algorithm are symmetric [54], the negative tail of the density represents the probability of error. If we denote the probability density functions of the messages at the outputs of degree-\(i\) lower and upper variable nodes at the beginning of the \(l\)-th iteration by \(q^l_i\) and \(p^l_i\), respectively, and their corresponding error probabilities by \(e_1(p^l_i, q^l_i)\) and \(e_2(p^l_i, q^l_i)\), respectively, the overall message error probability at the beginning of the \(l\)-th iteration can be calculated by (see [12] for further details)

\[
e(p^l, q^l) = \sum_i \left\{ \eta \lambda^{(1)}_i e_1(p^l_i, q^l_i) + (1 - \eta) \lambda^{(2)}_i e_2(p^l_i, q^l_i) \right\},
\]

where \(q^l\) and \(p^l\) \((l = 1, \cdots, L)\) are the message densities at the inputs of the lower and upper check nodes at the beginning of the \(l\)-th iteration, respectively. In designing bilayer
lengthened LDPC codes, the lower graph code (i.e., the destination code) is designed first. Then this code is fixed and the overall graph code (i.e., the relay code) is designed accordingly. As a result, assuming a fixed $\lambda^{(1)}$, Eq. (2.10) suggests a linear dependence of $e(p^l, q^l)$ on $\lambda^{(2)}$. However, due to the dependence of $e_1(p^l_i, q^l_i)$ and $e_2(p^l_i, q^l_i)$ on $\lambda^{(2)}$, this relation is not linear. Nevertheless, we can assume that $e(p^l, q^l)$ is linearly dependent on $\lambda^{(2)}$ by assuming very small changes in $\lambda^{(2)}$. Therefore, a linear programming problem to design the highest rate bilayer lengthened LDPC code for a given $\sigma_{SR}$ can be formulated as

$$\begin{align*}
\text{maximize} & \quad \sum_i \lambda_i^{(2)}/i \\
\text{subject to} & \quad \sum_i \left\{ \eta \lambda_i^{(1)} e_1(p_i^{l+1}, q_i^{l+1}) \\
& \quad + (1 - \eta) \lambda_i^{(2)} e_2(p_i^{l+1}, q_i^{l+1}) \right\} \\
& < \mu^{(h)} e(p^l, q^l), \quad l = 1, \ldots, L \\
\sum_i \lambda_i^{(2)} &= 1. \quad (2.11)
\end{align*}$$

From Eq. (2.9), it is clear that maximizing the objective function maximizes the rate of the bilayer code. In this formulation, $h$ is the optimization iteration number, $\mu$ is a positive coefficient which slowly increases towards one and determines the maximum value of changes in $\lambda^{(2)}$ and $L$ is the maximum allowed number of iterations in the iterative decoding algorithm. Furthermore, $\lambda^{(1)}$ is assumed fixed as discussed earlier in this section. The inequality constraint ensures that by using the updated $\lambda^{(2)}$, the symbol error probability decreases at each decoding iteration.

### 2.3 Extending EXIT Charts to Bilayer Lengthened LDPC Codes

EXIT charts are effective tools to analyze iterative decoders where instead of tracking the evolution of the messages densities, the evolution of a single parameter corresponding to the messages is studied.

Initially, the mutual information between the messages and the decoded bits was used in EXIT charts [34,35]. Subsequently, other parameters were employed in analysis of EXIT
charts, including the SNR of the extrinsic information [36,37] or the error probability of the message densities [38].

In this work, we are interested in tracking the evolution of the error probability of the message densities and wish to extend the notion of EXIT charts to BL-LDPC codes. The analysis is performed by defining a function $f$ which can compute the error probability at the output of the variable nodes at each iteration (see [39].) Recall that the overall message error probability at the beginning of the $l$-th iteration (i.e., at the output of the variable nodes) is denoted by $e(p', q')$. For a fixed $d_c'$, the variable node degree distributions for the upper graph, $\lambda^{(2)}$, should be designed such that the symbol error probability is decreased at each iteration. If we define $e^{(l)} = e(p', q')$, the above constraint on $\lambda^{(2)}$ results in $e^{(l+1)} < e^{(l)}, l = 1, \cdots , L$ and by plotting $e^{(l+1)}$ vs. $e^{(l)}$ for $l = 1, \cdots , L$, we obtain the overall EXIT chart of the BL-LDPC code which can be considered as a mapping

$$e^{(l+1)} = f(e^{(l)}), \quad l = 1, \cdots , L$$

(2.12)

where $f$, denoting the EXIT chart, can be obtained by interpolating $L$ points $(e^{(1)}, f(e^{(1)})), \cdots , (e^{(L)}, f(e^{(L)}))$, as well as the origin. Similar to the case of conventional LDPC codes, we can associate elementary EXIT charts with the message error probabilities at the outputs of degree-$i$ variable nodes and determine the overall EXIT chart of BL-LDPC codes as the linear combination of such elementary EXIT charts, i.e.,

$$f(e) = \sum \left\{ \eta \lambda^{(1)}_i f^{(1)}_i(e, \lambda^{(1)}) + (1 - \eta) \lambda^{(2)}_i f^{(2)}_i(e, \lambda^{(2)}) \right\},$$

(2.13)

where $f^{(1)}_i(e, \lambda^{(1)})$ and $f^{(2)}_i(e, \lambda^{(2)})$ are the elementary EXIT charts associated with degree-$i$ lower and upper variable nodes, respectively. This relation indicates that the overall EXIT chart depends on the lower and upper degree distributions as well as the message error probabilities at the previous iteration. However, since in designing a complexity-optimized BL-LDPC the lower graph code is fixed, the overall EXIT chart is only dependent on the upper graph degree distributions. An EXIT chart of a bilayer LDPC code is plotted in Figure 2.3. The dotted curves represent the elementary EXIT charts ($f^{(2)}_i$’s) and the dashed curve demonstrates the overall EXIT chart. It is obvious from this figure that the openness of the EXIT chart with respect to the line $e^{(l)} = e^{(l+1)}$ determines how fast the message error
probability can reach the target error probability. Consequently, $\lambda^{(2)}$ should be designed such that the overall EXIT chart is shaped for fast convergence.

2.3.1 Estimating the Number of Decoding Iterations Using EXIT Charts

In [53], the following relation was proposed to estimate the required number of iterations for the convergence of the iterative decoding of an irregular LDPC code using its EXIT chart

$$N \approx \int_{e_t}^{e_0} \frac{de}{\ln \sum \lambda_i f(e)}, \quad (2.14)$$

where $e_0$ and $e_t$ are the initial and target error probabilities, respectively. The proof is outlined in [53] as follows: In the beginning, it is assumed that the EXIT chart $f(\cdot)$ is a straight line passing through the origin and the point $(e^{(k)}, f(e^{(k)}))$ for $0 \leq k \leq N$, and as
such, a relation is developed which precisely calculates the number of iterations $N$ to go from $e_0$ to $e_t$. Then the analysis is extended to the case of nonlinear $f(\cdot)$ by computing the incremental increase in $N$ as a function of the incremental decrease in $e$ and finally, integrating over all $e \in [e_t, e_0]$.

Using this relation, an optimization problem is formulated to design irregular LDPC codes with reduced decoding complexity. Here we will use a similar approach to design complexity-optimized codes for relay channels.

### 2.4 Decoding Complexity for Bilayer LDPC Codes

Our goal here is to minimize decoding complexity at the relay only. The reason for this approach is twofold. First, power and computational capabilities at the relay are usually more constrained than at the destination. Second, any decoding delay at the relay would increase the overall decoding delay at the destination. Since iterative decoding is used to decode bilayer LDPC codes, we need to develop a metric by which the decoding complexity at these codes is measured.

The decoding complexity of iterative decoding is proportional to $NE$ where $N$ is the number of iterations and $E$ is the total number of edges. Each codeword encodes $nR_{BL}$ information bits where $R_{BL}$ is the rate of the bilayer code. Therefore, a measure of the decoding complexity per information bit, denoted by $K$, can be expressed as

$$K = \frac{NE}{nR_{BL}}.$$  \hspace{1cm} (2.15)

For the general BL-LDPC codes, where the check nodes have irregular lower and upper degrees, the total number of edges, $E$, can be calculated as

$$E = E_1 + E_2 = m \left( \frac{1}{\sum_j \rho_j^{(1)}/j} + \frac{1}{\sum_j \rho_j^{(2)}/j} \right).$$  \hspace{1cm} (2.16)

By replacing Eq. (2.16) in Eq. (2.15) and after some manipulation, we obtain

$$K = N \left( \frac{1 - R_{BL}}{R_{BL}} \right) \left( \frac{1}{\sum_j \rho_j^{(1)}/j} + \frac{1}{\sum_j \rho_j^{(2)}/j} \right).$$  \hspace{1cm} (2.17)
Eq. (2.17) indicates that the decoding complexity is a function of the number of
iterations required for the convergence of the bilayer code, as well as the sum of the averages
of the lower and upper check node degrees, i.e.,

$$\bar{d}_c = \left( \frac{1}{\sum_j \rho_j^{(1)}/j} \right) \text{ and } \bar{d}'_c = \left( \frac{1}{\sum_j \rho_j^{(2)}/j} \right)$$

In Sec ??, we indicated a method to estimate the number of iterations using the EXIT
chart of an LDPC code [53]. Based on the extended notion of EXIT charts for BL-LDPC
codes elaborated earlier, we can use a similar technique to estimate the number of iterations
required for the convergence of a BL-LDPC code as

$$N \approx \int_{e_t}^{e_0} \frac{de}{e \ln \frac{e}{f(e)}}, \quad (2.18)$$

where \(f(e)\) can be calculated using Eq. (2.13). This relation allows us to use a continuous
optimization approach to determine \(\lambda^{(2)}\) such that the required number of iterations for
convergence is minimized.

2.5 The Design Methodology

As previously explained, in designing complexity-optimized bilayer lengthened LDPC
codes, no restriction is applied on the complexity of the destination code (the lower graph
code). However, we wish to keep the rates of both the relay and the destination codes
unchanged, since for a given signal-to-noise ratio, these rates are obtained using a rate-
optimization algorithm (see [11].) Thus, a complexity-optimized bilayer code can be designed
by fixing the lower graph (a rate-optimized code) and optimizing the upper graph such that
the overall bilayer code is complexity-optimized at the relay. The optimization problem to
select the parameters of the upper graph, i.e., $\rho^{(2)}$ and $\lambda^{(2)}$, can be formulated as

$$\text{minimize } (1 - R_{BL}) \left( \frac{1}{\sum_j \rho_j^{(1)} / j} + \frac{1}{\sum_j \rho_j^{(2)} / j} \right) \times \int_{e_0}^{e_T} e \ln \left( \frac{e}{\sum_i \eta \lambda_i^{(1)} f_i^{(1)}(e, \lambda^{(1)}) + (1 - \eta) \lambda_i^{(2)} f_i^{(2)}(e, \lambda^{(2)})} \right) \, de$$

subject to

$$\sum_i \lambda_i^{(2)} \geq \sum_j \rho_j^{(2)} \left( \frac{1}{1 - R_{BL}} - \frac{\sum_i \lambda_i^{(1)} / i}{\sum_j \rho_j^{(1)} / j} \right),$$

$$\sum_i \left\{ \eta \lambda_i^{(1)} e_1(p_i^{l+1}, q_i^{l+1}) + (1 - \eta) \lambda_i^{(2)} e_2(p_i^{l+1}, q_i^{l+1}) \right\} < \mu^{(h)} e(p_l, q_l), \quad l = 1, \cdots, L,$$

$$\sum_i \lambda_i^{(2)} = 1 \quad (2.19)$$

In order to initialize the above optimization problem, we use the upper variable node degree distribution of a rate-optimized BL-LDPC code and denote it by $\lambda_{\text{init}}^{(2)}$. At the first optimization round ($h = 1$), a very small value for $\mu^{(h)}$ is selected. This ensures that the updated $\lambda^{(2)}$ is very close to $\lambda_{\text{init}}^{(2)}$. If the updated $\lambda^{(2)}$ does not improve in comparison with $\lambda_{\text{init}}^{(2)}$, we increase $\mu^{(h)}$ in the next round of the optimization until the updated $\lambda^{(2)}$ improves compared to $\lambda_{\text{init}}^{(2)}$; we then use this value as the new starting point. The optimization is then repeated until $\mu^{(h)} = 1$.

From Eq. (2.17), the complexity measure, $K$, also depends on the upper check node degree distribution $\rho^{(2)}$. In this work, we have studied only regular check node degrees. The degree of the check nodes has been set to the check node degree of the rate-optimized codes which we have considered for initializing our optimization problem. First, we select a fixed $d'_c$ in our codes to ensure that a fair comparison between our results and the existing rate-optimized codes is possible. Then we investigate the impact of choosing different check node degrees in order to find the optimum $d'_c$. 
2.6 Code Design Results

We design complexity-optimized BL-LDPC codes and compare their convergence behavior as well as performance with the rate-optimized BL-LDPC code (F) reported in [11]. Code (F) has rates (0.4877, 0.6906) and is designed for target rates (0.5, 0.7). We select our target rates to be the same as those used in designing code (F). The lower rate corresponds to the destination code and the higher rate to the BL-LDPC code intended for the relay. The lower graph for code (F) is characterized by \( \rho^{(1)}(x) = x^7 \) and

\[
\lambda^{(1)}(x) = 0.2421x + 0.2039x^2 + 0.1677x^5 + 0.0829x^6 + 0.3034x^{19}
\]

and the upper graph for this code is characterized by \( \rho^{(2)}(x) = x^5 \) and

\[
\lambda^{(2)}(x) = 0.1468x + 0.2331x^2 + 0.3039x^6 + 0.0298x^7 + 0.2864x^{18}.
\]

The corresponding convergence thresholds for code (F) are (0.3443 dB, 1.3716 dB). Thus, for any SNR > 1.3716 dB, we can solve the optimization problem (2.19).

We initialize our optimization problem by starting from the rate-optimized code (F). Since no restrictions are applied on the complexity of the destination code, the lower graph degree distribution pair \((\rho^{(1)}, \lambda^{(1)})\) is fixed and the optimization is performed only over the upper graph degree distribution pair. We set the maximum allowed number of iterations to 600 and the target error probability to 10^{-6}. We employ the discretized density evolution technique of [33] with 13 quantization bits. The highest permissible variable node degree is set to 19, equal to the upper variable nodes in code (F).

We consider two different cases corresponding to the upper degrees of the check nodes \(d'_c\). First, we set \(d'_c\) equal to the upper degrees of the check nodes of code (F), i.e., \(d'_c = 6\). We design three codes of rates (0.4877, 0.6906) for different values of signal-to-noise and compare the convergence behavior of our codes and of code (F). Then we consider the impact of choosing different values of the upper degrees of the check nodes on the complexity measure \(K\). For this case, we assume \(d'_c = 5, 6, \) and 7.
2.6.1 Case 1: $d'_c = 6$

We set $\rho^{(2)}(x) = x^5$ and SNR = 1.6140 dB. Solving the optimization problem formulated in Eq. (19) results in the following BL-LDPC code with $R_{BL} = 0.6906$:

$$\lambda^{(2)}(x) = 0.0019x + 0.4537x^2 + 0.2105x^6 + 0.1005x^7 + 0.0936x^{10} + 0.0856x^{11} + 0.0542x^{18}.$$  

We denote the designed code by $(C_1)$. As expected, code $(C_1)$ has a faster convergence speed (i.e., lower decoding complexity) compared to code $(F)$. For example, while code $(F)$ converges at 133 iterations at SNR = 1.6140 dB, code $(C_1)$ converges at only 52 iterations, demonstrating an improvement of 60.9% compared to the rate-optimized code $(F)$. Furthermore, at SNR = 1.4919 dB, code $(F)$ converges at 214 iterations, while code $(C_1)$ converges at only 99 iterations indicating an improvement of 53.7% in decoding complexity.

In the next step, we set $\rho^{(2)}(x) = x^5$ and SNR = 1.8633 dB, and design the following bilayer LDPC code with $R_{BL} = 0.6906$:

$$\lambda^{(2)}(x) = 0.0039x + 0.4433x^2 + 0.2045x^6 + 0.1639x^7 + 0.1094x^{10} + 0.0750x^{18}.$$  

We denote this code by $(C_2)$ which converges at 40 iterations (SNR = 1.7377 dB) and 33 iterations (SNR = 1.8633 dB) compared to 96 and 74 iterations for code $(F)$. This indicates that in comparison with code $(F)$, the decoding complexity of code $(C_2)$ is reduced by 58.3% and 55.4%, respectively.

Finally, we set $\rho^{(2)}(x) = x^5$ and SNR = 2.12 dB, and design the following BL-LDPC code with $R_{BL} = 0.6906$:

$$\lambda^{(2)}(x) = 0.0050x + 0.4473x^2 + 0.1902x^6 + 0.1555x^7 + 0.1164x^{10} + 0.0856x^{18}.$$  

We denote this code by $(C_3)$ which converges at 29 iterations (SNR = 1.9907 dB) and 26 iterations (SNR = 2.12 dB). While code $(F)$ converges at 60 iterations (SNR = 1.9907 dB) and 50 iterations (SNR = 2.12 dB). As a result, in comparison with code $(F)$, the decoding complexity of code $(C_3)$ is reduced by 51.6% and 48%, respectively.
2.6.2 Case 2: $d'_c = 5, 6, \text{ or } 7$

In this part, we continue to assume regular check nodes. However, we do not restrict the upper degrees of the check nodes to 6, rather we design codes with check nodes of upper degrees 5, 6, or 7.

As discussed before, for any SNR $> 1.3716 \text{ dB}$, we can design a complexity-optimized bilayer code of rate 0.6906. For example, code $(C_4)$ is optimized for SNR = 1.6956 dB and has an upper graph degree distribution pair $\rho^{(2)}(x) = x^6$ and

$$
\lambda^{(2)}(x) = 0.0036x + 0.3517x^2 + 0.2426x^7 + 0.2103x^8 \\
+ 0.1918x^{18}.
$$

Code $(C_5)$ is optimized for SNR = 2.0758 dB and has an upper graph degree distribution pair $\rho^{(2)}(x) = x^5$ and

$$
\lambda^{(2)}(x) = 0.0074x + 0.4544x^2 + 0.1572x^6 + 0.1289x^7 \\
+ 0.0938x^{10} + 0.0881x^{11} + 0.0702x^{18}.
$$

Code $(C_6)$ is optimized for SNR = 2.4738 dB and has an upper graph degree distribution pair $\rho^{(2)}(x) = x^4$ and

$$
\lambda^{(2)}(x) = 0.0205x + 0.6244x^2 + 0.0999x^6 + 0.0808x^7 \\
+ 0.0636x^{10} + 0.0554x^{11} + 0.0555x^{18}.
$$

We emphasize that all these codes have similar lower graph degree distributions as code $(F)$. For different values of SNR, we compare the required number of iterations for convergence of codes $(C_i, i = 4, 5, 6)$ and code $(F)$ in Table I, where we have also calculated the maximum improvement in decoding complexity compared to code $(F)$. This table indicates that although codes $(C_4)$, $(C_5)$, and $(C_6)$ are optimized for a particular channel threshold, they converge faster than code $(F)$ in a wide range of SNRs. However, it should be noted that for lower SNRs, these codes are likely to perform worse than code $(F)$. This is due to the fact that the target SNRs for which these codes are optimized are normally higher than the corresponding value for code $(F)$. For example, code $(C_4)$ does not converge at SNR = 1.4507 dB.
The complexity measure $K$ for codes $(C_i, i = 4, 5, 6)$ and code $(F)$ has been calculated using Eq. (2.17) and the results for different values of channel noise threshold $\sigma$ are illustrated in Figure 2.4. We observe that the complexity measure of codes $(C_i, i = 4, 5, 6)$ is significantly less than the complexity measure of code $(F)$. Additionally, using these plots, it is possible to choose the upper check node degree which yields the lowest complexity at the relay decoder by knowing the noise threshold of the source-relay channel. We can also determine a particular level of complexity at which the relay needs to operate and then determine the upper check node degree and the corresponding bilayer code which can operate at the highest level of noise threshold at the target complexity measure.

Since the complexity-optimized codes are designed to converge in the smallest number of iterations, we expect that when the number of decoding iterations at the relay decoder is restricted to the number at which these codes are optimized, they perform better than the rate-optimized code $(F)$. Our simulation results illustrated in the next section confirms this.

### 2.7 Performance Results

We have simulated our designed codes over a classical relay channel by constructing BL-LDPC codes with block length $10^5$, transmitted using BPSK signaling. The bit-error-rate (BER) vs. $E_b/N_0$ plots for three pair of codes, $(C_4, F)$, $(C_5, F)$, and $(C_6, F)$ are shown in Figure 2.5. The number of iterations for each pair is the number at which the complexity-optimized code $(C_i, i = 4, 5, 6)$ is optimum, i.e., we have set the maximum number of iterations for the decoding of $(C_4, F)$ to 43, for $(C_5, F)$ to 27, and for $(C_6, F)$ to 21.

Simulation results show that the waterfall region occurs earlier for the rate-optimized code $(F)$ compared to the complexity-optimized codes and that in low values of $E_b/N_0$, code $(F)$ performs better than the complexity-optimized codes. This is due to the fact that these codes are designed at a higher target $E_b/N_0$ in comparison with code $(F)$. However, as $E_b/N_0$ increases (even at bit error probabilities higher than $10^{-3}$), the complexity-optimized codes outperform the rate-optimized code $(F)$. For example, at BER $= 10^{-4}$, codes $(C_4)$, $(C_5)$, and $(C_6)$ outperform code $(F)$ by 0.05 dB, 0.16 dB, and 0.18 dB, respectively. As $E_b/N_0$ increases, the difference between the coding gains of the complexity-optimized codes and
Table 2.1: Number of Iterations Required for the Convergence of Code (F) and Complexity-optimized Codes at Different Values of SNR (dB).

<table>
<thead>
<tr>
<th>Code</th>
<th>(F)</th>
<th>(C₄)</th>
<th>(C₅)</th>
<th>(C₆)</th>
<th>Max. Improve.</th>
</tr>
</thead>
<tbody>
<tr>
<td>SNR = 1.4507</td>
<td>269</td>
<td>N.C.</td>
<td>186</td>
<td>514</td>
<td>31%</td>
</tr>
<tr>
<td>SNR = 1.5721</td>
<td>153</td>
<td>67</td>
<td>58</td>
<td>67</td>
<td>62%</td>
</tr>
<tr>
<td>SNR = 1.6956</td>
<td>106</td>
<td>43</td>
<td>43</td>
<td>48</td>
<td>59%</td>
</tr>
<tr>
<td>SNR = 1.8204</td>
<td>81</td>
<td>35</td>
<td>36</td>
<td>39</td>
<td>57%</td>
</tr>
<tr>
<td>SNR = 1.9474</td>
<td>65</td>
<td>30</td>
<td>31</td>
<td>33</td>
<td>54%</td>
</tr>
<tr>
<td>SNR = 2.0758</td>
<td>53</td>
<td>27</td>
<td>27</td>
<td>29</td>
<td>49%</td>
</tr>
<tr>
<td>SNR = 2.2066</td>
<td>45</td>
<td>24</td>
<td>24</td>
<td>26</td>
<td>47%</td>
</tr>
<tr>
<td>SNR = 2.3391</td>
<td>39</td>
<td>22</td>
<td>22</td>
<td>23</td>
<td>44%</td>
</tr>
<tr>
<td>SNR = 2.4738</td>
<td>34</td>
<td>20</td>
<td>20</td>
<td>21</td>
<td>41%</td>
</tr>
<tr>
<td>SNR = 2.6107</td>
<td>30</td>
<td>18</td>
<td>18</td>
<td>19</td>
<td>40%</td>
</tr>
<tr>
<td>SNR = 2.7497</td>
<td>27</td>
<td>17</td>
<td>17</td>
<td>18</td>
<td>37%</td>
</tr>
</tbody>
</table>

code (F) increases as the BER curves of these codes experience steeper waterfall regions compared with the BER curves of code (F). Furthermore, it is noticed that unlike code (F), the complexity-optimized codes do not exhibit error floors in the SNRs of interest. We note that no algorithm is implemented to remove cycles of length four and above from the parity-check matrices of the codes. However, the error floor phenomenon can be attributed to the fraction of degree-2 variable nodes in these codes. It is known that two parameters that have an impact on the error floor performance of LDPC codes, i.e., minimum distance of the code and size of the smallest stopping set, depend on the fraction of degree-2 variable nodes (see [55] and [56].) It is noticed that the complexity-optimized codes have substantially smaller fractions of degree-2 variable nodes in comparison with the rate-optimized code (F), a fact that also results in a more open EXIT chart near the origin for complexity-optimized
Figure 2.4: Decoding complexity of the designed bilayer lengthened LDPC codes.

codes which explains their fast convergence in comparison with code (F).

2.8 Conclusion

We propose a technique to design complexity-optimized bilayer lengthened LDPC codes. Our technique retains the rate of the relay code and minimizes its decoding complexity. This is possible by generalizing EXIT charts to BL-LDPC codes and estimating the number of decoding iterations using the generalized EXIT chart. Our technique keeps the rate-optimized code of the destination node unchanged. Using this approach, we design BL-LDPC codes with noticeable reduction in decoding complexity and delay compared to the rate-optimize codes reported in the literature.

We also design a wide class of complexity-optimized BL-LDPC codes with different upper check node degrees and study their decoding complexity over a wide range of conver-
Figure 2.5: Comparison of performance of codes \((C_4), (C_5),\) and \((C_6),\) evaluated at their target number of iterations with the performance of code \((F)\) at the same number of iterations.

gence thresholds. Our results show that these codes require considerably fewer iterations to converge compared to the corresponding rate-optimized code. When simulated over codes with long block lengths, these codes demonstrate lower bit error probability especially at mid-to-high signal to noise ratios. For example, at BER = \(10^{-4}\), our codes achieve a performance of up to 0.18 dB better than the rate-optimized codes.
Chapter 3

Bilayer Lengthened LDPC Codes for Rayleigh Fading Relay Channels

In this chapter, we first study the decode-and-forward achievable rates for a Rayleigh fading relay channel. Then, we design bilayer lengthened LDPC codes for this channel model under different SNR assumptions, namely, when the link between the source and the relay has a much higher SNR than the link between the source and the destination, and when the two SNRs are comparable.

3.1 System Model

The system model is the classic relay channel model as shown in Figure 3.1. We recall that the system consists of a source node (S), a relay node (R), and a destination node (D).

The links are all assumed to have normalized Rayleigh fading coefficients, i.e., S-R link with coefficient $a_1$, S-D link with coefficient $a_2$, and R-D link with coefficient $a_3$, and all the coefficients are independent from each other. Hence, we have $E[a_i^2] = 1$ and $p(a_i) = 2a_i \exp(-a_i^2), i = 1, 2, 3$. We also assume that the corresponding receivers have perfect channel side information (CSI), but the channel is unknown to the transmitters.

We assume independent AWGN noises, $Z_1 \sim N(0, N_1)$ at the relay and $Z_2 \sim N(0, N_1 + N_2)$ at the destination. Hence, if we denote the transmitted signal by the source by $X$ and the transmitted signal by the relay by $X_1$, and at the relay, $Y_1$, and the output of the channel
Figure 3.1: The relay channel model.

at the destination, $Y$, can be written as

\begin{align}
Y_1 &= a_1 X + Z_1 \\
Y &= a_2 X + a_3 X_1 + Z_2.
\end{align}

The relaying strategy for our system is based on the classic decode-and-forward (DF) technique described in Chapter 1. Before we proceed to section 3.2, where we discuss how to calculate the achievable rate for the DF scheme, we show how to calculate the achievable rate for a direct link where the channel is Rayleigh fading with normalized fading coefficient $a$ and CSI is known at the receiver only. Consider a channel with the following input-output relation

\begin{equation}
Y = aX + Z
\end{equation}

where $X$ and $Y$ are the input signal and the channel output, respectively, and $Z \sim N(0, N_0)$. If the transmitter power is constraint to $P$, the channel capacity is given by

\begin{equation}
C(a) = \frac{1}{2} \log \left(1 + \frac{a^2 P}{N_0}\right)
\end{equation}
hence, the ergodic capacity of the Rayleigh fading link is given by (cf. [57])

\[
C = \int_0^\infty C(a)p(a)da \\
= \int_0^\infty \log \left(1 + \frac{a^2 P}{N_0}\right) a \exp(-a^2)da \\
= -\frac{1}{2\ln 2} e^{\frac{1}{\rho}} Ei \left(-\frac{1}{\rho}\right) \\
\triangleq C_R(\rho),
\]

where \( \rho = P/N_0 \) and \( Ei(\cdot) \) is the exponential integral function defined as

\[
Ei(x) \triangleq \int_{-\infty}^x \frac{e^t}{t} dt.
\]

### 3.2 Achievable Rate Calculation for the Decode-and-Forward Strategy

In the DF strategy deployed in this work, the transmitter divides its power, \( P \), into two parts. One part is allocated to transmit the new message \( w_i \) and the other part is allocated to transmit the index \( s_i \) of the previous message \( w_{i-1} \). As a result, the transmitter transmits

\[
X = \tilde{X}(w_i) + \sqrt{\frac{(1 - \alpha)P}{P_1}} X_1(s_i),
\]

where \( \alpha \) is the power allocation factor, \( \alpha P \) is the power of \( \tilde{X} \), and \( P_1 \) is the power of \( X_1 \). Therefore, the signal received at the relay is given by

\[
Y_1 = a_1 \tilde{X}(w_i) + a_1 \sqrt{\frac{(1 - \alpha)P}{P_1}} X_1(s_i) + Z_1.
\]

Since the relay has decoded \( s_i \) in the previous block, it subtracts the corresponding term in (3.9) from \( Y_1 \). Subsequently, for the relay node to decode \( w_i \), the decode-and-forward rate
$R$ should be upper bounded by

$$R \leq I(X;Y_1|X_1)$$

$$= -\frac{1}{2} \ln 2 e^{\left(\frac{1}{\rho_r}\right)} \text{Ei} \left( -\frac{1}{\rho_r} \right)$$

$$= C_R(\rho_r), \quad (3.10)$$

where $\rho_r = \alpha P/N_1$ denotes the relay SNR.

The destination receives $Y$ which can be calculated by

$$Y = a_2X + a_3X_1 + Z_2$$

$$= a_2\tilde{X}(w_i) + \left(a_2 \sqrt{\frac{(1 - \alpha)P}{P_1}} + a_3\right)X_1(s_i) + Z_2. \quad (3.11)$$

In order to decode $s_i$, the destination regards $a_2\tilde{X}(w_i)$ as noise. For a fixed value of $a_2$ and $a_3$, $R_1$ should satisfy the following inequality,

$$R_1(a_2, a_3) \leq I(X_1;Y)$$

$$= \frac{1}{2} \log \left( 1 + \frac{\left(a_2\sqrt{(1 - \alpha)P} + a_3\sqrt{P_1}\right)^2}{a_2\alpha P + N_1 + N_2} \right). \quad (3.12)$$

However, $a_2$ and $a_3$ are independent Rayleigh fading coefficients and, hence, we obtain

$$R_1 = \int_0^\infty \int_0^\infty \log \left( 1 + \frac{\left(a_2\sqrt{(1 - \alpha)P} + a_3\sqrt{P_1}\right)^2}{a_2\alpha P + N_1 + N_2} \right)$$

$$\times 2a_2a_3e^{-(a_2^2+a_3^2)}da_2da_3. \quad (3.13)$$

After decoding $X_1(s_i)$, the destination can subtract the corresponding term from (3.11) and decode $\tilde{X}(w_i)$ provided that

$$R - R_1 \leq I(\tilde{X};Y|X_1)$$

$$= -\frac{1}{2} \ln 2 e^{\left(\frac{1}{\rho_d}\right)} \text{Ei} \left( -\frac{1}{\rho_d} \right)$$

$$= C_R(\rho_d), \quad (3.14)$$
where $\rho_d = \alpha P/(N_1 + N_2)$ denotes the destination SNR.

The capacity of relay channels can be determined by applying the max-flow min-cut theorem of Ford and Fulkerson which states that in a flow network the maximum flow passing from the source to the sink is upper bounded by the capacity of a min-cut of such network [58]. By applying this theorem, it is shown in [4] that the overall DF rate of the degraded relay channel is given by

$$R = \max_{\alpha} \min \{C_R(\rho_r), C_R(\rho_d) + R_1\},$$

(3.15)

where $\alpha$ can be found by equating the two terms inside the braces in (3.15).

In summary, the source codeword $\tilde{X}(w_i)$ is decodable at the relay at

$$\text{SNR}_{SR} = \frac{\alpha P}{N_1} = \rho_r,$$

(3.16)

while it is decodable at the destination at

$$\text{SNR}_{SD} = \frac{\alpha P}{N_1 + N_2} = \rho_d,$$

(3.17)

and with the extra information made available by the relay. Consequently, the codes designed for this relaying strategy have rates equal to

$$R_{SR} = C_R(\text{SNR}_{SR})$$

(3.18)

$$R_{SD} = C_R(\text{SNR}_{SD}).$$

(3.19)

### 3.3 Code Design Based on Bilayer Density Evolution

In this section, we describe the technique to design bilayer lengthened low-density parity-check (BL-LDPC) codes for Rayleigh fading relay channels based on the achievable rates derived in the previous section. For this purpose, we deploy the bilayer density evolution technique as implemented in [11].
The design target rate is set to be the ergodic achievable rate for the decode and forward strategy over the independent Rayleigh fading relay channel with perfect side information available at the corresponding receivers.

The first step in designing a BL-LDPC code is to design a good conventional LDPC code for the source-destination link (the lower layer code). For this purpose, we only need to use the conventional density evolution technique, i.e. we need to analyze the evolution of the initial message density received from the channel passed along the lower edges of the graph and updated only at the lower variable nodes and the check nodes. From [59], the initial density of the log-likelihood-ratios of the messages received from the channel is

\[
P(0)(q_0) = \frac{\sigma}{\sqrt{2\pi}} \exp \left( -\frac{q_0(\sqrt{2\sigma^2 + 1} - 1)}{2} \right) \times \int_0^\infty \exp \left( -\frac{\left( \frac{\sigma^2}{2\pi} q_0 - a\sqrt{2\sigma^2 + 1} \right)^2}{2\sigma^2} \right) da,
\]

where \( q_0 \) is the LLR of the message received from the channel, \( \sigma^2 \) is the variance of the noise and \( a \) is the normalized Rayleigh fading coefficient. The channel side information is only known at the corresponding receivers, as discussed in section 4.2.

Using the update rules of conventional density evolution technique, we can track the evolution of \( P(0)(q_0) \) and formulate the design problem based on linear programming to find an optimized degree distribution pair \( \lambda^{(1)}(x) \) and \( \rho^{(1)}(x) \) for the lower layer. As discussed in Chapter 1, we assume that the check nodes have a fixed lower degree equal to \( d_c \) which is generally found by checking all possible values. Hence, we use the optimization problem of Eq. (1.9) to find \( \lambda^{(1)}(x) \) and design the highest rate lower layer code which is supposed to be decodable at the destination where its bin index has been provided by the relay.

The next step to design a BL-LDPC code is to fix the lower layer code and optimize the upper layer degree distribution pair \( \lambda^{(2)}(x) \) and \( \rho^{(2)}(x) \) based on the given \( \text{SNR}_{SR} \) and bilayer density evolution update rules. Similar to the lower layer code, we assume that the check nodes have fixed upper degrees equal to \( d'_c \) which is found in similar to \( d_c \). In the following, we explain how bilayer DE is formulated and can be used to design bilayer lengthened LDPC codes by formulating a linear programming problem.
3.3.1 Bilayer Density Evolution

Bilayer DE is an extension of DE for bilayer LDPC codes [12]. This technique is used to study the evolution of the message densities over the bilayer graph. The bilayer DE update rules in both variable and check nodes are similar to the update rules for conventional DE. However, since in the structure of BL-LDPC codes, the check nodes are common between both lower and upper edges, the update rules at check nodes are modified to account for the presence of both sets of edges.

In order to design BL-LDPC codes, we should calculate the error probability of the message densities at each iteration and formulate a design problem to find the upper variable node degree distribution $\lambda_i^{(2)}$ such that the error probability is decreased at each iteration.

If we denote the message densities at the outputs of degree-$i$ lower and upper variable nodes at the beginning of the $l$-th iteration by $q^l_i$ and $p^l_i$, and their corresponding error probabilities by $e^{(1)}(p^l_i, q^l_i)$ and $e^{(2)}(p^l_i, q^l_i)$, respectively, the overall message error probability at the beginning of the $l$-th iteration can be written as (for more details, refer to [12])

$$e(p^l_i, q^l_i) = \sum_i \left\{ \eta \lambda_i^{(1)} e^{(1)}(p^l_i, q^l_i) + (1 - \eta) \lambda_i^{(2)} e^{(2)}(p^l_i, q^l_i) \right\}$$

where $q^l_i$ ($p^l_i$) is the message density at the input of the check nodes in the lower (upper) subgraph.

Since in designing the upper graph degree distributions of BL-LDPC codes we fix the lower graph code (i.e., a fixed $\lambda^{(1)}$), Eq. (3.22) suggests a linear dependence of $e(p^l_i, q^l_i)$ on $\lambda^{(2)}$. However, due to the dependence of $e^{(1)}(p^l_i, q^l_i)$ and $e^{(2)}(p^l_i, q^l_i)$ on $\lambda^{(2)}$, this relation is not really linear. However, assuming very small changes in $\lambda^{(2)}$, we can use a linear approximation of $e(p^l_i, q^l_i)$ as a function of $\lambda^{(2)}$. Hence, in a linear programming setting the problem of designing the highest rate bilayer lengthened LDPC code for a given SNR can
be formulated as

\[
\text{maximize } \sum_i \frac{\lambda_i^{(2)}}{i} \\
\text{subject to } \sum_i \left\{ \eta \lambda_i^{(1)} e^{(1)}(p_i^{l+1}, q_i^{l+1}) + (1 - \eta) \lambda_i^{(2)} e^{(2)}(p_i^{l+1}, q_i^{l+1}) \right\} < \mu^{(h)} e(p^l, q^l), \quad l = 1, \cdots, L \\
\sum_i \lambda_i^{(2)} = 1. \tag{3.22}
\]

### 3.4 Simulation Results

We have used discretized density evolution approach of [33] to perform the density evolution technique. The maximum value of the log-likelihood ratio (LLR) was set to 25 and was discretized using 12 quantization bits. The maximum number of iterations in the density evolution method and the target error probability were set to 600 and $10^{-6}$, respectively. Consequently, each of the designed codes achieves an error probability equal to $10^{-6}$ in less than 600 iterations. Using the described technique, four bilayer lengthened LDPC codes have been designed and the rates for the corresponding asymptotic infinite-length converging codes are computed.

We have designed bilayer lengthened LDPC codes for two different scenarios. In the first scenario, the source-relay and the source-destination links have comparable SNRs, whereas in the second scenario, the source-relay link is much stronger than the source-destination link. In all designs, we have used regular check node degrees. We first designed two codes for the source-destination link with maximum variable node degrees equal to 10 and 20. These codes have been used as the destination codes in both scenarios. Next, we designed two codes for the source-relay link in each scenario, where the designed codes have maximum variable node degrees equal to 11 and 21.

Parameters of the codes we have designed for these scenarios are represented in Tables 3.1 and 3.2. The corresponding target SNRs are denoted by $\text{SNR}_{\text{SR}}$ and $\text{SNR}_{\text{SD}}$. Our results show that the destination codes have a smaller gap to the capacity in comparison with the relay codes. For example, the gap that each destination code has to the capacity is less than 0.44 dB. However, the gap generally increases for the relay code (with the exception of code
Furthermore, we notice that as the rate of the relay code increases (i.e., the codes in Table 3.2), the gap to the capacity increases too.

### 3.5 Concluding Remarks

In this chapter, we studied the achievable rates for the decode-and-forward (DF) relaying strategy for the Rayleigh fading relay channel where the links have independent normalized Rayleigh fading coefficients and the channel side information is perfectly known at the corresponding receivers but not at the transmitters. We also designed bilayer lengthened LDPC codes for this scenario for the case when the source-relay link is much stronger than the source-destination link as well as for the case when these two links are comparable. Our code design results reveal that the thresholds of bilayer codes have a smaller gap to the capacity when the source-relay and the source-destination links have comparable SNRs.
Table 3.1: Optimized Degree Distributions for $\text{SNR}_{SR} = 1.5\text{SNR}_{SD}$.

<table>
<thead>
<tr>
<th>Code</th>
<th>$C_1$</th>
<th>$C_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$\lambda_i$</td>
<td>$\lambda_l$</td>
</tr>
<tr>
<td>$\lambda_2$</td>
<td>0.3417</td>
<td>0.1659</td>
</tr>
<tr>
<td>$\lambda_3$</td>
<td>0.2796</td>
<td>0.2314</td>
</tr>
<tr>
<td>$\lambda_6$</td>
<td>0.1915</td>
<td></td>
</tr>
<tr>
<td>$\lambda_7$</td>
<td></td>
<td>0.5222</td>
</tr>
<tr>
<td>$\lambda_9$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\lambda_{10}$</td>
<td>0.1872</td>
<td></td>
</tr>
<tr>
<td>$\lambda_{20}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\lambda_{21}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$d_c$</td>
<td>6</td>
<td>11</td>
</tr>
<tr>
<td>$d'_c$</td>
<td>4</td>
<td>6</td>
</tr>
<tr>
<td>$R_{SD}$</td>
<td>0.4704</td>
<td>0.6373</td>
</tr>
<tr>
<td>$R_{SR}$</td>
<td>0.6507</td>
<td>0.7341</td>
</tr>
<tr>
<td>$\text{SNR}_{SD}$ (dB)</td>
<td>2.0001</td>
<td>3.698</td>
</tr>
<tr>
<td>$\text{SNR}_{SR}$ (dB)</td>
<td>3.7610</td>
<td>5.4588</td>
</tr>
<tr>
<td>$\text{Gap}_{SD}$ (dB)</td>
<td>0.4321</td>
<td>0.4080</td>
</tr>
<tr>
<td>$\text{Gap}_{SR}$ (dB)</td>
<td>0.3010</td>
<td>0.7788</td>
</tr>
</tbody>
</table>
Table 3.2: Optimized Degree Distributions for $\text{SNR}_{SR}(\text{dB}) = \text{SNR}_{SD}(\text{dB}) + 5\text{dB}$.

<table>
<thead>
<tr>
<th>Code</th>
<th>$C_3$</th>
<th>$C_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$\lambda_i$</td>
<td>$\lambda_i'$</td>
</tr>
<tr>
<td>$\lambda_2$</td>
<td>0.3417</td>
<td>0.2165</td>
</tr>
<tr>
<td>$\lambda_3$</td>
<td>0.2796</td>
<td>0.2757</td>
</tr>
<tr>
<td>$\lambda_4$</td>
<td>0.0004</td>
<td></td>
</tr>
<tr>
<td>$\lambda_6$</td>
<td>0.1915</td>
<td></td>
</tr>
<tr>
<td>$\lambda_7$</td>
<td>0.5222</td>
<td>0.2859</td>
</tr>
<tr>
<td>$\lambda_{10}$</td>
<td>0.1872</td>
<td></td>
</tr>
<tr>
<td>$\lambda_{11}$</td>
<td>0.6192</td>
<td></td>
</tr>
<tr>
<td>$\lambda_{20}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\lambda_{21}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$d_c$</td>
<td>6</td>
<td>11</td>
</tr>
<tr>
<td>$d'_c$</td>
<td>9</td>
<td>13</td>
</tr>
<tr>
<td>$R_{SD}$</td>
<td>0.4704</td>
<td>0.6373</td>
</tr>
<tr>
<td>$R_{SR}$</td>
<td>0.7713</td>
<td>0.8461</td>
</tr>
<tr>
<td>$\text{SNR}_{SD}$ (dB)</td>
<td>2.00</td>
<td>3.698</td>
</tr>
<tr>
<td>$\text{SNR}_{SR}$ (dB)</td>
<td>7.00</td>
<td>8.698</td>
</tr>
<tr>
<td>$\text{Gap}_{SD}$ (dB)</td>
<td>0.4321</td>
<td>0.4080</td>
</tr>
<tr>
<td>$\text{Gap}_{SR}$ (dB)</td>
<td>1.1162</td>
<td>1.4867</td>
</tr>
</tbody>
</table>
Chapter 4

Cooperative Communication Using Protograph-Based Low-Density Parity-Check Codes

We propose a novel two-user cooperation scheme that employs protograph-based low-density parity-check (LDPC) codes. The proposed scenario is based on time division where each user transmits its message to the base station (BS) in two successive intervals. In the first interval, the user sends its message to the BS as well as its partner user. If the partner user successfully decodes the received signal, the two users cooperatively send the main user’s message using the Alamouti scheme during the second interval. Otherwise, the main user simply retransmits its message to the BS. The users change roles during the next two intervals. Furthermore, the users encode their information over a class of protograph-based LDPC codes called the Accumulate-Repeat-4-Jagged-Accumulate (AR4JA) codes that allow flexible selections of rate and length. Using this technique, an order-3 diversity is achieved when there exists a strong interuser link, whereas when the interuser channel has a poor quality, at least an order-2 diversity is achieved.
4.1 Introduction

During the last decade, cooperative communication has attracted significant attention as a class of strategies for achieving higher data rates by providing cooperative diversity as a form of spatial diversity. Sendonaris, et. al. proposed a technique in [1] to achieve cooperative diversity via the cooperation of two mobile users that wish to communicate with a common base-station (BS). They evaluated the potential benefits of user cooperation using an information-theoretic point of view and also provided an example of how user cooperation can be implemented in a practical system.

Additionally, many techniques have been proposed to incorporate different channel coding techniques into the two-user cooperation strategy. For example, in [19], a rate-1/2 convolutional code is used in the interuser link and a rate-1/4 convolutional code in the user-BS channels. Other coding techniques have also been incorporated into cooperative strategies, such as using space-time codes, Turbo codes and low-density parity-check (LDPC) codes. For a number of examples, see [20–22, 60].

On the other hand, various algorithms have been proposed recently to design channel coding techniques that have excellent performance and benefit from simple encoder/decoder architectures. An interesting class of such codes was proposed by Thorpe in [25] where he introduced a class of LDPC codes that are constructed from protographs. A protograph is a bipartite graph of relatively short length that serves as a template for constructing an LDPC code of arbitrary size. The performance of the constructed LDPC code can be predicted by analyzing the performance of the template protograph which is often called the base-protograph.

In this chapter, we propose a two-user cooperation strategy which employs a class of protograph-based LDPC codes called Accumulate-Repeat-4-Jagged-Accumulate (AR4JA) codes. This class of codes benefits from a few key features which make it a strong candidate for many applications. In the proposed cooperative model, each user wishes to transmit its own message to a common base station (BS). Furthermore, when possible, the two users cooperate in order to transmit their messages to the BS by employing the Alamouti scheme in a distributed way.

The rest of this chapter is organized as follows. In Sec. 4.2, we describe the system
model. Sec. 4.3 provides an introduction to the AR4JA family of protograph-based LDPC codes. Then the proposed cooperative strategy is introduced in Sec. 4.4. We also discuss rate and length considerations in our system in Sec. 4.5. Simulation results are presented in Sec. 4.6 and we conclude the chapter in Sec. 4.7.

4.2 System Model

The proposed system comprises of two users and a base station (BS) where each user wishes to send its own message to the BS and also tries to assist the other user— which will be called the partner user— to transmit its message, cooperatively, to the BS. (See Figure 4.1.) Let us assume that user $i$ ($i = 1, 2$) transmits the modulated signal $s_i(n)$ at time $n$. The received signal at user $j$ ($j = 1, 2$ and $j \neq i$), denoted by $r_{ij}(n)$ is

$$r_{ij}(n) = h_{ij}(n)s_i(n) + n_j(n)$$  \hspace{1cm} (4.1)

where $h_{ij}(n)$ is the channel coefficient between user $i$ and user $j$ and $n_j(n)$ is the additive noise at user $j$. We assume flat Rayleigh fading links. Therefore, the channel coefficients are modeled as zero-mean complex Gaussian random variables with variance 0.5 per dimension. In this work, we consider quasi-static fading which means that the fading coefficients are constant during a time frame of size, say, $N_f$. Moreover, zeros-mean complex AWGN noise with variance 0.5 per dimension is assumed at the corresponding receivers. In the remainder of this chapter, we drop the coefficient $n$ for simplicity.

Using a similar notation, the received signal at the BS is given by

$$r_{i0} = h_{i0}s_i + n_0$$  \hspace{1cm} (4.2)

where subscript “0” indicates the BS. It is assumed that perfect channel state information is available to the corresponding receivers, and the transmitters only know the statistics of the channel, but not its current realization.

**Definition 1.** The diversity order [61], [62] that can be obtained by a code is defined as

$$d = - \lim_{\gamma \to +\infty} \frac{\log P_{ew}}{\log \gamma}$$  \hspace{1cm} (4.3)

where $\gamma$ is the SNR and $P_{ew}$ is the word error probability.
4.3 The AR4JA Family of Protograph-Based LDPC Codes

As we mentioned in Sec. 4.1, an LDPC code of arbitrary size can be constructed from a base protograph. The process that results in such construction consists of lifting the protograph and applying edge permutations. Lifting refers to creating multiple copies of the same protograph where the number of copies is determined by the desired length of the target LDPC code. Furthermore, various copies of the protograph are connected by applying permutations among the edges that connect similar pairs of variable-check nodes in different replicas of the protograph. Such edges are considered to be of the same type. Generally, circulant permutations are chosen in order to simplify the hardware implementation.

Figure 4.2 shows the protograph of the AR4JA class of LDPC codes introduced in [63]. The numbers on edges indicate the number of parallel edges. The AR4JA codes benefit from a few features that make them strong candidates for many applications. First, they represent a family of rate-compatible LDPC codes of rates $R = \frac{n}{n+1}, n = 1, 2, \ldots$, where the proper rate might be chosen depending on channel conditions. Second, as will be shown in this section, the parity-check matrix of any code in this family (except for the rate 1/2 code) is constructed from the $H$ matrix of the lower rate codes and thus it is possible to implement a simple decoder architecture for a variety of rates. Moreover,
the further structure imposed on the graph of protograph-based codes results in reduced implementation cost, parallel architecture of the decoder and simple encoding. Finally, it is shown in [63] that the minimum linear distance of these codes grows linearly with the block length which is a desirable feature for a linear block code. The AR4JA codes have been already incorporated into near-earth and deep space applications [64].

Let us show how parity check matrices of AR4JA codes of different rates can be constructed. If we denote the parity-check matrix of a rate 1/2 AR4JA code by $H_{\frac{1}{2}}$, from Figure 4.2 it is easy to verify that

$$H_{\frac{1}{2}} = \begin{bmatrix} 1 & 0 & 0 & 0 & 2 \\ 0 & 1 & 1 & 1 & 3 \\ 0 & 1 & 2 & 2 & 1 \end{bmatrix}$$

Now assume that the parity check matrix of an AR4JA code of rate $R = \frac{n}{n+1}$ is given and denoted by $H_{\frac{n}{n+1}}$. Then the parity-check matrix of the AR4JA code of rate $R' = \frac{n+1}{n+2}$

\[\text{Figure 4.2: Protograph of the AR4JA class of LDPC codes.}\]
can be constructed by

\[
H_{n+1} = \begin{bmatrix} 0 & 0 \\ 3 & 1 \\ 1 & 3 \\ H_{n+1} \end{bmatrix}
\] (4.5)

Therefore, the parity-check matrix of an AR4JA code of any rate can be easily constructed using Eqs. (3-4).

The protograph of AR4JA codes as shown in Figure 4.2, as well as the above discussion on constructing higher rate codes of this family, suggest the bilayer structure of these codes which may be deployed in scenarios incorporating relaying. In fact, the AR4JA class represents a family of structured bilayer lengthened LDPC codes, which is in contrast to unstructured bilayer lengthened LDPC codes (see [65].) In this work, however, we do not exploit the bilayer structure of AR4JA codes. Instead, we propose a strategy for user cooperation which employs these codes as well as the Alamouti scheme.

### 4.4 The Cooperative Communication Strategy

The proposed cooperative strategy is based on time division. The users encode their information using the same AR4JA code which has length \( N \) and each user transmits its message over two successive blocks of length \( N \). (See Figure 4.3.) Let us assume \( N = N_f \), i.e., the fading frame size. However, this is not the only possible assumption and we will see in the next section that we can relax this constraint.

Additionally, we assume that the links between the users and the BS are orthogonal and therefore when cooperation is not employed, direct transmission between the users and
the BS is possible. The first two blocks are dedicated to the transmission of user 1’s message and the second two blocks are dedicated to the transmission of user 2’s message. During blocks 2 and 4, the users choose to either cooperate or transmit their messages to the BS directly, depending on whether or not the partner successfully decoded the message. However, if a user chooses to cooperate, it forms a distributed two-antenna transmitter with its partner to employ the Alamouti scheme in a distributed fashion and transmits its message to the BS cooperatively.

Let us describe our scheme in more details. Following the above discussion, user 1 transmits its message during block 1. At the end of this block, the BS only stores the received signal, while user 2 decodes the received codeword. Here we assume that the users can detect the detection errors using a cyclic redundancy check (CRC) and that they can inform their partner user whether or not the decoding has been successful using an error-free link. Therefore, at the end of the first block, user 1 knows if user 2 has been successful in decoding its message and therefore, one of the following two scenarios can be employed in block 2:

**Scenario 1.** User 2 has successfully decoded user 1’s message and thus they can employ the Alamouti scheme to transmit user 1’s message entirely. Here the role of a two-antenna system is played by the two-user system in a distributed fashion. Therefore, at the end of block 2, the BS receives an additional sequence with an order-2 diversity. In order to decode user 1’s message, the sum-product decoding algorithm is performed at the BS where the algorithm is initialized by combining the channel outputs at the end of block 1 and the outputs of the maximum-likelihood detector of the Alamouti scheme at the end of block 2. The received message at the BS thus benefits from an order-3 diversity.

We note that the proper rate can be selected based on the channel conditions and the block length N. We will elaborate more on this in Sec. 4.5.

**Scenario 2.** User 2 has been unsuccessful in decoding user 1’s message. In this case, the users do not cooperate; rather user 1 retransmits its message entirely during block 2. Therefore, the BS receives two version’s of user 1’s message and the system achieves an order-2 diversity.

In blocks 3 and 4, the users exchange roles and the same approach is employed to transmit user 2’s message to the BS.
4.5 Rate and Length Considerations

Although our proposed scheme can be implemented using a wide range of channel codes, for reasons mentioned in Sec. 4.3, we employ AR4JA codes of different rates. From Eq. (4.5) we note that a high rate AR4JA code can be constructed using a lower rate code of this family. We also note that by increasing the rate, the codeword length increases as well. Thus, if one wishes to have flexibility in choosing different rates or codeword lengths, incorporating these codes into the proposed cooperative strategy allows flexibility, while maintaining a unified decoder architecture in the corresponding receivers.

In this work, we take advantage of the constructions introduced and advised in [64]. The shortest rate-1/2 AR4JA code that has been advised in [64] is of length 640 bits among which 128 bits are punctured and only 512 bits are transmitted. The rate-2/3 code constructed based on this code has 896 bits with 768 transmitted bits. The next extension is the rate-3/4 code of length 1152 with 1024 transmitted bits. The rate-1/2, 2/3, and 3/4 codes described here have 256, 512, and 768 information bits, respectively.

We note that it is possible to choose codes of different rates and lengths depending on the fading frame size $N_f$. In general, for a frame size of $N_f$, any AR4JA code of length $N = lN_f$ can be used where $l$ is an integer. For example, if the frame size is 256, any of the codes described above can be deployed. If the frame size is 512, the rate-1/2 or 3/4 codes can be used in the proposed system.

4.6 Simulation Results

Here the simulation results that evaluate the performance of the proposed system are presented. We assume a quasi-static block Rayleigh fading channel with frame sizes of 512 and 1024 bits. The AR4JA codes deployed in our system have been constructed using the algorithm presented in [64]. For each code, the number of transmitted bits is equal to the fading frame size. Thus, rate 1/2 code has length 640 with 512 transmitted bits and rate 3/4 code has length 1152 with 1024 transmitted bits.

It is generally assumed that the users have a strong interuser link. However, we wish to evaluate the performance of the proposed system in different interuser channel qualities.
In our simulations, we assume cases where the interuser channel is perfect, i.e., there is an error-free link between the two users, as well as the cases where the interuser channel frame-error rate (FER) is equal to 0.01, 0.1, 0.5 and 0.9.

The simulation results plotted in Figure 4.4 illustrate the FER versus $E_b/N_0$ performance curves for the proposed system, where a rate $\frac{1}{2}$ AR4JA code is employed in a channel with frame size of 512 bits and different interuser channel qualities are assumed. Here the effective rate is $\frac{1}{4}$ since 256 information bits are transmitted in two successive blocks, i.e., 1024 channel uses. Recall that when decoding is unsuccessful in the partner user, the main user retransmits its message and therefore at least an order-2 diversity is expected. It is noticed that when the interuser channel quality is poor, i.e., at FERs of 0.5 and 0.9, an order-2 diversity is obtained. Nevertheless, we obtain some coding gain in the former case. When the quality of the interuser channel improves (i.e., FERs of 0.1 and 0.01),
User Cooperation Using the rate $\frac{3}{4}$ AR4JA Code and the Alamouti Scheme

![Graph showing FER performance as a function of Eb/No for different interuser channel qualities (R = $\frac{3}{4}$).]

**Figure 4.5:** Frame error-rate (FER) performance as a function of $E_b/N_0$ for different interuser channel qualities ($R = \frac{3}{4}$).

The diversity level is increased. Finally, when the interuser channel is perfect, we obtain a level-3 diversity. It is also noticed that at channel FER of 0.01, the performance of the user-BS link is comparable to the performance of the case of a perfect interuser link. To achieve a user-BS FER of $10^{-4}$, the latter scenario outperforms the former one by almost 0.8 dB. Moreover, to achieve a user-BS FER of $10^{-3}$, the scenario with the interuser FER of 0.01 outperforms the case with the interuser FER of 0.9 by 7.4 dB.

The simulation results for a rate $\frac{3}{4}$ code which has been employed over a channel with frame size of 1024 bits is shown in Figure 4.5. In this case, the effective communication rate is $\frac{3}{8}$. Simulations reveal results similar to the previous case, where a rate $\frac{1}{2}$ code was used. It is observed that at poor interuser channel qualities, a two-level degree diversity is guaranteed, whereas when the interuser channel enhances, the diversity level increases. When the interuser channel is perfect, we achieve an order-3 diversity. Even at interuser
channel FER of 0.01, an order-3 diversity is observed. In this case, the required $E_b/N_0$ to achieve a user-BS FER of $10^{-3}$ is 5.8 dB less compared to the case when the interuser channel FER is 0.9.

4.7 Conclusions

In this chapter, we presented a novel two-user cooperative scheme that deploys a class of protograph-based rate-compatible LDPC codes called AR4JA codes. In the proposed technique, the block length and code rate are determined considering the fading frame size. When it is possible, the users employ the Alamouti scheme in a distributed fashion to transmit a message that both know. The proposed technique guarantees an order-2 diversity when the interuser channel quality is poor and an order-3 diversity when the interuser link is strong.
Chapter 5

Protograph-Based Codes for cooperative communication over Slow Fading Channels

5.1 Introduction

In this chapter, we address the code design problem for the two-user cooperative scheme where channels are modeled as block-fading (BF) frequency non-selective where it is assumed that the fading coefficients are constant over the codeword length. When cooperation is not employed, the word error rate (WER) versus SNR curve has the same slope as the uncoded system at high SNR regime, i.e., diversity one, where diversity is defined as the negative slope of WER versus SNR when both are plotted logarithmically when SNR tends to infinity. Cooperative communication, however, provides diversity through user cooperation which could potentially increase the coding rate and result in higher order diversity.

Despite advances in designing capacity-approaching codes for many channel models, there is not as much progress in designing highly efficient codes for the BF channel, which is a challenging channel model. There are several reasons that contribute to this issue. For instance, Shannon capacity for the BF channel is zero and outage probability is used as a benchmark to evaluate the performance of the designed codes. There are several well-
known techniques that can be applied to design capacity-approaching low-density graph codes for the AWGN channel. However, employing these techniques does not necessarily result in outage-probability approaching low-density graph codes. Furthermore, the best codes designed for the AWGN channel cannot usually achieve full-diversity when employed over the BF channel.

Rate-compatible convolutional (RCC) codes have been employed in a number of works as the channel coding scheme in coded cooperation [19, 22]. However, it has been shown [66] that convolutional codes and short block codes cannot achieve the outage probability limit since their WER grows logarithmically as the code length increases. This is a major drawback since the WER of outage probability approaching should be independent of the block length.

Several methods have been proposed to design capacity-approaching codes for Gaussian relay channels (see [12], [67], and [8]). However, a method was proposed in [68] to design low-density graph codes that employ the so-called rootchecks, a special class of parity check nodes that provide support to the information bits in the case of a deep fade when they are represented by a class of variable nodes called rootbits. This structure is necessary to achieve full-diversity in BF channels under iterative decoding and since capacity approaching codes in Gaussian channels usually do not have this structure, they do not perform well in the BF channel.

The root structure proposed in [68] is employed and extended to include the notion of rate-compatibility in [2] which results in designing rate-compatible-root LDPC codes for coded cooperation. In particular, the proposed code construction technique does not provide more than one coding rate for cooperation. This construction technique begins with a root LDPC code of rate $R_r = 0.5$ and modifies its parity-check matrix by applying the extension technique which converts it into the parity-check matrix of a code of rate $R_c < R_r$ that is used for cooperation. Therefore, only one code is obtained for cooperation. Furthermore, the authors employ unstructured LDPC codes which are not popular for practical implementation due to their lack of structure.

Here we propose two construction methods to design full-diversity codes for cooperative communication for the BF channel. The techniques we introduce result in design procedures for rate-compatible codes in the sense that a large family of codes with different rates are constructed from a mother code. Therefore, the transmitters can select among
a wide range of rates depending on the available SNR and reliability requirements. Furthermore, our construction methods employ protograph-based codes; thus enjoying simple encoding and parallel architecture for the decoder that would result in high speed implementation of the decoder.

5.2 Preliminaries

5.2.1 System Model

We consider a model that consists of two users and a base station (BS) (see Figure 5.1). The devices are half-duplex and the users employ a time-division based scheme. Whenever cooperation is possible, each user relays its partner user’s message to the BS. Furthermore, the user-BS links are assumed to be orthogonal and therefore when cooperation is not employed, the users independently transmit their messages to the BS [19], [22], [2].

We will assume that the codeword transmitted by user $i$ ($i = 1, 2$) is in the form $C_i = (C_{i,1}, C_{i,2})$ of length $N = N_1 + N_2$, where $N_1$ and $N_2$ are the lengths of $C_{i,1}$ and $C_{i,2}$, respectively. The level of cooperation is defined as $\beta = N_2/N$. In this work, we select $N_1 = N_2$ and thus, $N = 2N_1$ and $\beta = 0.5^1$. We assume that each block consists of two frames of equal length. First, let us consider user 1. In the first frame, user 1 broadcasts $C_{1,1}$. Both user 2 and the BS receive the transmitted signal. However, only user 2 attempts to decode the received signal as we assume that the interuser channel is much stronger than the user-BS channel. If user 2 is successful in decoding user 1’s message, it will transmit $C_{1,2}$ to the BS on behalf of user 1. Otherwise, it will notify user 1 using a Cyclic Redundancy Check (CRC) code and therefore user 1 will continue by transmitting $C_{1,2}$. The BS decodes user 1’s message using both $C_{1,1}$ and $C_{1,2}$. In the next block, user 1 and user 2 change roles. The time division scheme employed by the users is shown in Figure 5.2.

Let us assume that user $i$ transmits the modulated signal $s_i[k]$ at time $k$. The received

---

1In fact, it was shown in [2] that in order to achieve a second order diversity in a 2-user cooperative scheme over the BF channel, the coding rate must be less than or equal to $\min(\beta, (1 - \beta))$. Therefore, we select $\beta = 0.5$ to communicate at the highest coding rate that is possible.
signal at user $j$ ($j = 1, 2$ and $j \neq i$), denoted by $r_{ij}$, is given by

$$r_{ij}[k] = h_{ij} s_i[k] + n_j[k]$$

(5.1)

where $h_{ij}$ is the Rayleigh distributed channel coefficient between user $i$ and user $j$, with $E[h_{ij}^2] = 1$, and $n_j[k]$ is the additive noise at user $j$, where $n_j[k] \sim \mathcal{N}(0, \sigma^2)$. Similarly, for the signal received at the BS, we have

$$r_{i0}[k] = h_{i0} s_i[k] + n_0[k]$$

(5.2)

where $h_{i0}$ is the Rayleigh distributed channel coefficient between user $i$ and the BS, with $E[h_{i0}^2] = 1$, and $n_0[k]$ is the additive noise at the BS and $n_0[k] \sim \mathcal{N}(0, \sigma^2)$. It is assumed that the channel model is quasi-static and the fading coefficients are constant during a communication block or two consecutive frames.

We assume BPSK signaling in this work and denote the total energy per information bit-to-noise ratio by $E_b/N_0$. Note that the channel model considered here is memoryless and it satisfies the channel symmetry condition [69]; i.e., $p(r_{ij}[k]|h_{ij}, s_i[k] = 1) = p(-r_{ij}[k]|h_{ij}, s_i[k] = -1)$.

**Definition 2.** The diversity order [61], [62] that can be obtained by a code is defined as

$$d = - \lim_{\gamma \to +\infty} \frac{\log P_{ew}}{\log \gamma}$$

(5.3)

where $\gamma$ is the SNR and $P_{ew}$ is the word error probability.
Figure 5.2: Time division scheduling for transmission of each user's message.

Definition 3. In a system with \( N_u \) cooperative users, a code \( C \) is called full diversity if it achieves a diversity order of \( d = N_u \).

It has been shown in [70] that for a full diversity code with diversity order \( d \), the maximum coding rate is given by \( R_{\text{max}} = 1/d \). Therefore, for a full diversity code \( C \) in a 2 user cooperative scheme, we obtain \( R_{\text{max}} = 0.5 \). Furthermore, it has been shown in [2] that in a two user system employing coded cooperation over the BF channel with cooperation level \( \beta \), any coding rate greater than \( \min(\beta, (1 - \beta)) \) results in single order diversity.

5.2.2 Root LDPC Codes

Unstructured root LDPC codes were introduced in [68], [71], and [72] as a class of codes that achieve full diversity in a BF channel where each block consists of two frames of equal lengths and the codeword is spanned over the length of a block. Channel coefficients are assumed to be constant over a block length, but change independently from one block to another block.

Root LDPC codes can achieve full diversity under iterative decoding through the introduction of the so-called rootchecks. A rootcheck is a special type of check node that is connected to variable nodes transmitted in both frames. However, what distinguishes rootchecks from other check nodes is the restrictions imposed on the connections they have with their leavebits. Assume a root-LDPC code transmitted over two frames. A rootcheck is called type-i \( (i = 1, 2) \) if it is connected to only one variable node transmitted in Frame \( i \). A variable node in Frame \( i \) connected to a rootcheck of type \( i \) is called a rootbit. Figure 5.3 shows the parity-check matrix of a rate-1/2 root-LDPC code. As it is shown in this figure, the construction of parity-check matrix of root-LDPC codes consists of four classes of variable
nodes and two classes of check nodes. In a rate-1/2 root-LDPC code, the information bits are divided into two classes that are denoted by $1i$ and $2i$ and the parity bits are divided into two classes denoted by $1p$ and $2p$. Variable nodes $1i$ and $2i$ are rootbits that are connected to rootchecks $3c$ and $4c^2$ through an identity matrix, respectively. Therefore, if variable nodes of type $1i$ or $2i$ are erased due to a deep fade in frame 1 or 2, they can be retrieved through their connections in the other frame. Note that variable nodes $1p$ and $2p$ are not rootbits and hence do not benefit from the protection provided due to the root structure of these codes.

In this configuration, check nodes $3c$ are connected to variable nodes $2i$ and $2p$ through matrices $H_{2i}$ and $H_{2p}$, respectively. These matrices belong to unstructured designs and may be optimized separately or jointly. Similarly, check nodes $4c$ are connected to variable nodes $1i$ and $1p$ through matrices $H_{1i}$ and $H_{1p}$, respectively.

### 5.2.3 Rate-Compatible Root LDPC Codes

Rate-compatible root-LDPC (RCR-LDPC) codes were introduced in [2] as a class of root-LDPC codes that provide rate-compatibility through the extension technique. The

\[\begin{array}{cccc}
\hline
\text{Frame 1} & \text{Frame 2} \\
\hline
1 & 1 & 0 & H_{2i} \\
\cdots & 1 & \cdots & \cdots \\
H_{1i} & H_{1p} & \cdots & 0 \\
\hline
\end{array}\]

\[\begin{array}{cccc}
\hline
3c & 1 & \cdots & \cdots \\
\cdots & \cdots & \cdots & \cdots \\
1 & 1 & \cdots & \cdots \\
\hline
\end{array}\]

\[\begin{array}{cccc}
\hline
4c & 1 & \cdots & \cdots \\
\cdots & \cdots & \cdots & \cdots \\
1 & \cdots & \cdots & \cdots \\
\hline
\end{array}\]

\[\begin{array}{cccc}
\hline
\text{Figure 5.3:} \text{ The Parity-Check Matrix of a Root-LDPC Code of Rate 0.5.} \\
\end{array}\]
Figure 5.4: The Parity-Check Matrix of an RCR-LDPC Code of Rate 0.33.

parity-check matrix of an RCR-LDPC code is shown in Figure 5.4. The rate-compatibility is provided through the introduction of two sets of variable nodes that are denoted by \( p'_1 \) and \( p'_2 \) and two sets of check nodes denoted by \( 1c \) and \( 2c \). Therefore the RCR-LDPC code of Figure 5.4 is the combination of the root-LDPC code of Figure 5.3 and another code of rate \( R_1 \) whose degree distributions are characterized by the parity-check matrices \( H_{1s} \) and \( H_{1r} \). The rate \( R_c \) of the overall code is given by \( R_c = R_r R_1 \). The zeros padded in the \( H \)-matrix of the RCR-LDPC code serve to obtain a balanced construction that maintains the root structure of the overall code and the desired rate.

RCR-LDPC codes are employed for user cooperation using the same way that was described in Section 5.1. Therefore, the source user transmits only the encoded bits of Frame 1. However, the relaying user is required to decode the entire coded message using only half of the total number of information bits; i.e., only those bits that it has received during the first frame. This may sound counter-intuitive. However, as a result of the root structure of RCR-LDPC codes, information bits \( 2i \) and parity bits \( 2p \) and \( p'_2 \) are determined using the
values of the variable nodes received in the first block.

Designing RCR-LDPC codes consists of two steps: Optimizing the degree distributions of the root-LDPC code through designing $H_{1i}$, $H_{1p}$, $H_{2i}$, and $H_{2p}$ and optimizing the degree distributions of the code that provides rate-compatibility through designing $H_{1s}$ and $H_{1r}$. For simplicity, it is assumed that $H_{1i} = H_{2i}$, $H_{1p} = H_{2p}$, and $H_{1s} = H_{1r}$. In this case, designing an RCR-LDPC code consists of optimizing a pair of degree distributions $(\lambda_1(x), \rho_1(x))$ for the root-LDPC code and a pair of degree distributions $(\lambda_2(x), \rho_2(x))$ for the rate-compatible code.

The RCR-LDPC codes constructed using the above technique are categorized as multi-edge type LDPC codes ([41]) and a density evolution analysis for these codes is developed in [2] which has been used in [73] to design irregular RCR-LDPC codes.

The construction technique and the analysis introduced in [2] for RCR-LDPC codes focuses on unstructured LDPC codes. However, due to several practical limitations such as extensive memory requirements and difficult scalability, unstructured LDPC codes are very hard to implement in practical systems. On the other hand, although using the above construction it is possible to obtain a desired cooperative rate $R_c$ through the selection of $R_1$, it does not offer a family of rate-compatible codes and the optimized RCR-LDPC code performs only at a fixed rate. Therefore, when it is desired to select a different coding rate for the overall cooperative code, a new code has to be designed for the rate-compatible part.

To address the first problem, we propose employing protograph-based LDPC codes that help implementing high-speed decoding and simple encoding. Moreover, we propose two methods to design families of rate-compatible protograph-based root LDPC (RCPB-R-LDPC) codes. These techniques make it possible to design codes for cooperation that offer a broad range of rates, while belonging to the same code family. Therefore, their implementation becomes very efficient in practice.
5.3 Constructions of Protograph-Based Codes for Coded Cooperation

In this section, we propose two methods to design RCPB-R-LDPC codes for block fading channels. Codes designed using our techniques offer a diversity of order 2 and extensive rate-compatibility.

We begin by defining root protographs.

Definition 4. Let $G_r = (V, C, E)$ be an $(N, M)$ protograph. In a root protograph, the set of variable nodes $V$ comprises of four subsets, i.e., $V = \{v_{1i}, v_{1p}, v_{2i}, v_{2p}\}$, and the set of check nodes comprises of two subsets, i.e., $C = \{c_{3c}, c_{4c}\}$, where sets $v_{1i}$, $v_{2i}$, $c_{3c}$, and $c_{4c}$ have the same number of elements. Furthermore, every element in $v_{1i}$ ($v_{2i}$) is connected to every element in $c_{3c}$ ($c_{4c}$) through a single edge and elements of $v_{1p}$ ($v_{2p}$) are not connected to elements of $c_{3c}$ ($c_{4c}$).

Figure 5.5 represents a $(16, 8)$ root-protograph code that has four variable nodes of each of the classes $1i$, $1p$, $2i$, and $2p$. Additionally, it has four check nodes of class $3c$ and four check nodes of class $4c$. After lifting, variable nodes of classes $1i$ and $1p$ are transmitted in Frame 1 and variable nodes of classes $2i$ and $2p$ are transmitted in Frame 2. Note that check nodes of class $3c$ are rootchecks for rootbits of class $1i$ and check nodes of class $4c$ are rootchecks for rootbits of class $2i$. In the protograph of Figure 5.5, every rootbit is connected to only one rootcheck through a single edge. Therefore, when the protograph is lifted, every rootbit in Frame 1 is connected to only the variable nodes transmitted in Frame 2 through a rootcheck and vice versa. This statement is true regardless of the type of the permutations chosen.

Definition 5. Let $G_{rcr} = (V, C, E)$ be an $(N, M)$ protograph. A rate-compatible root protograph comprises of a root protograph that has been extended. More precisely, it has two additional sets of variable nodes $v_{p'1}$ and $v_{p'2}$ and two additional sets of check nodes $c_{1c}$ and $c_{2c}$, where variable nodes $v_{p'1}$ ($v_{p'2}$) are only connected to check nodes $c_{1c}$ ($c_{2c}$) and check nodes $c_{1c}$ ($c_{2c}$) are only connected to variable nodes $v_{1i}$, $v_{1p}$, $v_{p'1}$ ($v_{2i}$, $v_{2p}$, $v_{p'2}$).

Figure 5.6 represents a rate-compatible root protograph of rate 0.33. Note that the root subgraph of this protograph has rate 0.5 and consists of variable nodes $1i$, $1p$, $2i$, and
2p and check nodes 3c and 4c. The rate-compatibility is provided through extension by introducing check nodes 1c and 2c and variable nodes \( p'_1 \) and \( p'_2 \) that are transmitted in Frame 1 and Frame 2, respectively.

Constructions that are introduced in this chapter offer a family of rate-compatible codes, i.e., it is possible to design a RCR-PB-LDPC code for coded cooperation that offers more than one rate. This property is not present in unstructured RCR-LDPC codes introduced in [2].

Moreover, the protograph structure of the proposed constructions makes it possible to optimize the designed codes in two steps: Optimization of the root protograph and optimization of the extended graph which provides rate-compatibility. In this work, we only focus on code construction rather than protograph optimization and the selection of the protographs at each design step is based on computer search. Furthermore, we are interested in circulant permutations and choose the circulants according to the technique introduced in [74] so that the lifted graph has no cycle of length 4.

We focus on designing protograph-based codes with a minimum distance that grows
linearly with the codeword length [63]. This property is often called the _linear minimum distance growth property_. Codes that benefit from this property often have good error-floor performance. Usually, the normalized logarithmic asymptotic weight distribution of a protograph code is used to find _typical minimum distance ratio_ of the code which is used to determine whether it has linear minimum distance growth. Studies have shown that several structures of protograph codes [75] have this property. We review the method to obtain the typical minimum distance ratio of a protograph code and structures with linear minimum distance growth in the following section.

### 5.3.1 Asymptotic Weight Distribution for Protograph Codes

Asymptotic weight distributions of unstructured LDPC codes were studied in [76] for regular LDPC codes and in [77–80] for irregular LDPC codes. Moreover, asymptotic weight distributions of serially concatenated codes and turbo-like codes were studied in [81, 82]. Asymptotic weight distributions of protograph codes were studied first in [83, 84], and later in [85]. Here we review the important results of this subject based on its treatment in [75].

Let denote the code length by $n$ and Hamming weights of the code by $w$’s, where $0 \leq w \leq n$. Therefore, the normalized codeword weights, denoted by $\delta$ is defined by $\delta = \frac{w}{n}$. Ensemble weight enumerator for codes of length $n$ is then denoted by $A_w = A_{n\delta}$.

For protograph based code ensembles, we use the following notations and definitions: We denote the number of variable nodes in the protograph by $n_v$, the number of transmitted variable nodes by $n_t$, the number of punctured variable nodes by $n_u$, and the number of check nodes by $n_c$. Further, let denote the lifting factor by $N$. Therefore, each node in the protograph has $N$ copies. The partial Hamming weight associated with the $N$ copies of $v_j$,
Let denote the subvector of \( n_t \) partial weights corresponding to the transmitted nodes by \( w_t \) and the subvector of \( n_t \) normalized partial weights of the transmitted variable nodes by \( \delta_t \triangleq w_t/N \). Similarly, subvector of \( n_u \) normalized partial weights of the punctured variable nodes is denoted by \( \lambda_u \).

The degree of variable node \( v_j \), \( j = 1, \ldots, n_v \) and degree of the check node \( c_i \), \( i = 1, \ldots, n_c \) are denoted by \( q_{v} \) and \( q_{c} \), respectively. Moreover, subvector of \( q_{c} \) partial Hamming weights corresponding to the \( q_{c} \) variable nodes connected to check node \( c_i \) is defined as \( w_i = (w_1, \ldots, w_{q_{c}}) \). Then \( \delta_i \triangleq w_i/N \). If we denoted the number of sequences with weight vector \( w_i \) for check node \( c_i \) by \( A_{w_i} \), the weight vector enumerator \( A_w \) for protograph-based code ensembles is obtained by summing contributions from partial weight vectors \( w \) that yield total codeword weight \( w \):

\[
A_w = \sum_{w:|w|=w} \left[ \prod_{i=1}^{n_c} A_{w_{c_i}} \right] \prod_{j=1}^{n_v} \left( \frac{N}{w_j} \right)^{q_j-1} \tag{5.4}
\]

Let \( r_N(\delta) = \frac{\ln(s)}{N} \). Therefore,

\[
\frac{\ln(s)}{N} = \frac{\ln A_{N\delta}}{N} - \frac{\sum_{j=1}^{n_v} (q_j - 1) \ln \left( \frac{N}{N\delta_j} \right)}{N} \tag{5.5}
\]

A corresponding normalized logarithmic asymptotic weight distribution for each vector \( \delta \) of partial weights is

\[
r(\delta) = \lim sup_{N \to \infty} r_N(\delta) \tag{5.6}
\]

Let \( a^c(\delta_i) \triangleq \lim sup_{N \to \infty} \frac{\ln(A_{N\delta_i})}{N} \) be a normalized logarithmic asymptotic weight distribution for check node \( c_i \) with normalized partial weight vector \( \delta_i \) and \( H(x) \) be the binary entropy function. Then

\[
r(\delta) = \sum_{i=1}^{n_c} a^c(\delta_i) - \sum_{j=1}^{n_v} (q_j - 1) H(\delta_j) \tag{5.7}
\]
For a degree-3 check node, any triplet of normalized partial weights \((\epsilon_1, \epsilon_2, \epsilon_3)\) satisfying the check node constraint must satisfy the condition \(\max\{\epsilon_1, \epsilon_2, \epsilon_3\} \leq (\epsilon_1 + \epsilon_2 + \epsilon_3)/2 \triangleq \sigma \leq 1\) for which \(a^c(\cdot, \cdot, \cdot)\) is evaluated explicitly as
\[
a^c(\epsilon_1, \epsilon_2, \epsilon_3) = H_4(\sigma - \epsilon_1, \sigma - \epsilon_2, \sigma - \epsilon_3, 1 - \sigma)
= H_4\left(\frac{-\epsilon_1 + \epsilon_2 + \epsilon_3}{2}, \frac{\epsilon_1 - \epsilon_2 + \epsilon_3}{2}, \frac{\epsilon_1 + \epsilon_2 - \epsilon_3}{2}, 1 - \frac{\epsilon_1 + \epsilon_2 + \epsilon_3}{2}\right)\tag{5.8}
\]
where \(H_4(x_1, x_2, x_3, x_4) = -\sum_{i=1}^{4} x_i \ln x_i\).

We can obtain \(r_t(\delta_t)\) for any subvector \(\delta_t\) of normalized partial weights corresponding to transmitted variable nodes by unconstrained maximization of \(r(\delta)\) over subvector \(\lambda_u\) of normalized partial weights corresponding to punctured variable nodes, i.e.,
\[
r_t(\delta_t) = \max_{\lambda_u} r(\delta)\tag{5.9}
\]

Finally, the normalized logarithmic asymptotic weight distribution \(r(\delta)\) for the protograph code ensemble is evaluated as
\[
r(\delta) = \max_{\delta:|\delta|=n} r(\delta)/n_t = \max_{\delta_t:|\delta_t|=n_t} r_t(\delta_t)/n_t
= \max_{\delta:|\delta|=n} \max_{\lambda_u} r(\delta)/n_t\tag{5.10}
\]

Note that the normalized codeword weight \(\delta\) and code ensemble’s normalized logarithmic asymptotic weight distribution \(r(\delta)\) are both normalized by codeword length \(n = N n_t\), whereas the vector-based quantities \(\delta\) and \(r(\delta)\) are normalized by the lifting factor \(N\).

If the first zero crossing of \(r(\delta)\) exists for \(\delta = \delta_{\text{min}} > 0\), then \(r(\delta) < 0\) for any \(\delta < \delta_{\text{min}}\) and \(\delta_{\text{min}}\) is called the typical minimum distance ratio.

It is proved in [85] that if a protograph-based code has a typical minimum distance ratio, (i.e., when \(\delta_{\text{min}} > 0\)), then the minimum distance of a code in the ensemble grows linearly with the block length with probability close to one. Therefore, \(\delta_{\text{min}}\) of protograph codes may be considered during the protograph design procedure.

Figure 5.7 shows the plot \(r(\delta)\) vs. \(\delta\) curve for the root-protograph of Figure 5.5 and the RCPB-R-LDPC code of Figure 5.6. For a number of LDPC code families, \(\delta_{\text{min}}\) is known to be greater than zero. For example, it was shown in [86] that a sufficient condition for a protograph to have typical minimum distance ratio greater than zero is that the minimum variable node degree in the protograph to be at least 3.
However, LDPC codes with some degree-2 variable nodes are known to have very good thresholds in the AWGN channel. A technique was presented in [63] that shows how to introduce degree-2 variable nodes to a protograph with linear minimum distance growth property and yet preserve it in the modified protograph. This technique is called the check node splitting technique and since it is used in our methodology to construct RCPB-R-LDPC codes, we review this method in the following.

Essentially, the check node splitting technique is used to reduce the degree of a check node by introducing new checks and distributing the edges of the original check node among the new set of check nodes. Assume we are interested in applying the check node splitting technique to a check node of degree $q^c > 3$ such that its degree is equal to 3 and the degree of each newly added check is at most 3. This requires splitting the check node $q^c - 3$ times. Each time a check node is split, it is connected to the newly added check using a
Figure 5.8: Applying the check node splitting technique to a degree-5 check node.

degree-2 untransmitted variable node. Therefore, $q^c - 3$ new check nodes and $q^c - 3$ new
degree-2 untransmitted variable nodes are added to the graph. Note that puncturing the
newly added variable nodes is essential to maintain the rate. However, if we chosen to
transmit one or more of the added variable nodes, the rate decreases. Figure 5.8 shows
the application of the check node splitting technique to a degree-5 check node. The circles
with solid background represent transmitted variable nodes and circles with dashed line
represent punctured variable nodes. In this example, two additional check nodes and two
untransmitted variable nodes are added to the graph and therefore, the rate remains fixed.
However, if we wish to decrease the rate, we may transmit either one or both variable nodes.
We will employ this technique in the construction of our RCPB-R-LDPC codes.

Puncturing the variable nodes that are introduced to the graph through the check
node splitting technique results in maintaining the typical linear minimum distance ratio.
However, if one or more variable nodes are transmitted, this ratio is reduced by no more
than the reciprocal of the number of transmitted variable nodes in the new graph and hence
the linear minimum distance growth property is preserved ( [75]).
5.3.2 Method I

Since we are interested in designing RCPB-R-LDPC codes with linear minimum distance growth property, in this method we set the constraint that the minimum variable node degree in our codes is 3. However, in the second method we propose, degree-2 variable nodes are added to the graph by applying the check node splitting technique.

Figure 5.9 shows the general protograph of the RCPB-R-LDPC code of Method I. In order to design an RCPB-R-LDPC code, first a root protograph of rate $R_r \leq 0.5$ that satisfies the linear minimum distance growth property is designed. Techniques to design codes with this property are introduced in [75]. In this construction, the information bits are mapped onto variable nodes of classes 1 and 2 and the parity bits are mapped onto the variable nodes of classes 1 and 2. Further, there are two types of check nodes, i.e., 3c and 4c, where check nodes of type 3c are connected to variable nodes of types 1, 2, and 2p and check nodes of type 4c are connected to variable nodes of types 2i, 2p, and 1i.

As we will discuss later in this chapter, the performance of codes designed for the BF channel model is usually compared with the outage probability of this channel where Shannon capacity is zero as there is a non-zero probability that the channel is in deep fade. However, if the constructed code has root structure—which guarantees its full-diversity—it is advantageous to take the achievable threshold of the designed protograph into consideration while designing RCPB-R-LDPC codes. Therefore, any of the known techniques of designing low-threshold protograph codes for the AWGN channel may be considered while designing the initial root-protograph. (See [75] for a number of examples of capacity-approaching protograph-based codes.) There are also several tools to determine the threshold of the designed protographs. In particular, the protograph-based EXIT chart technique proposed in [87] is a fast and reliable method to determine the threshold of protograph codes in the AWGN channel.

The next step is to modify the designed protograph code in a way that it would become rate-compatible. The proposed technique to provide rate-compatibility in this work is based on extension, where information blocks are fixed and additional parity bits are introduced to the protograph. To achieve this, a new variable node and a new check node are added to each side of the root-protograph, where the variable-check node pairs at the
Figure 5.9: RCPB-R-LDPC code construction (Method I).

right side and at the left side are connected through \( r_1^v \) and \( l_1^v \) single edges, respectively, where \( r_1^v, l_1^v \geq 3 \) (see Figure 5.9.) The new check nodes at the right and left sides of the graph are also connected to the root-protograph using \( r_1^c \) and \( l_1^c \) edges, respectively, where the check node connections are selected so that a low threshold protograph is obtained.

Note that the check nodes added to the left and right sides of the root-protograph are of types \( 1c \) and \( 2c \), respectively and the variable nodes are of types \( p_1' \) and \( p_2' \), respectively. Further, the number of additional parity bits added to each side of the root-protograph are equal since the number of transmitted variable nodes in both frames are equal.

Now let us investigate the impact of the extension method described above on the code rate. Assume that the root-protograph has \( n_t \) transmitted variable node, \( n_u \) untransmitted
variable nodes, and \( m_r \) check nodes. The rate of the root-protograph code is then given by

\[
R_r = 1 - \frac{m_r - n_u}{n_t} \tag{5.11}
\]

If \( n_e \) variable–check node pairs are added to each side of the root-protograph and all of the new variable nodes are transmitted, the rate of the RCPB-R-LDPC code family constructed using the above method is given by

\[
R_c = 1 - \frac{m_r + 2n_e - n_u}{n_t + 2n_e}, \quad n_e = 1, 2, \ldots \tag{5.12}
\]

For instance, if \( n_u = 0, n_t = 4 \) and \( m_r = 2 \), then \( R_r = 0.5 \). For \( n_e = 1, 2, 3 \), we obtain \( R_c = 0.33, 0.25, 0.2 \), respectively.

The first method is summarized as follows:

1. Design a root protograph code of rate \( R_r \leq 0.5 \). Set the minimum variable node degree to at least 3 to ensure that the designed code has the linear minimum distance growth property.

2. Extend the root protograph by adding a pair of connected variable and check nodes to each side of the protograph. The new variable node should have degree at least 3. The check nodes connections to the rest of the graph are chosen to decrease the achievable threshold of the overall protograph in the AWGN channel.

3. Go to step 2 if the desired rate is not achieved yet.

4. Select the lifting factor based on the desired codeword length and the circulants so that the small cycles are avoided when the graph is lifted using the algorithm of [74].

5. Lift the protograph using the selected circulants to obtain the code of desired length.

### 5.3.3 Method II

The second proposed method employs the check node splitting technique which preserves the linear minimum distance growth property if the original graph possesses it. This method is implemented in two steps. In the beginning, a root protograph of rate \( R \leq 0.5 \) with linear minimum distance growth property is designed. The remarks stated for Method
I apply here too. Next, the root protograph is extended in two steps. First, a new variable node of class $p'_1$ and degree $l^v \geq 3$ is connected to the left side of the protograph through a new check node of class $1c$ and degree $l^v + l^c \geq 4$ (Figure 5.10). Similarly, a new variable node of class $p'_2$ and degree $r^v \geq 3$ is connected to the right side of the protograph through a new check node of class $2c$ and degree $r^v + r^c \geq 4$. This decreases the rate of the overall code. The next step is to obtain lower rate codes through the application of the check node splitting technique to the check nodes of classes $1c$ and $2c$. Assume we apply this technique to each check node $n_s$ times. Therefore, $n_s$ check nodes of classes $1p$ and $2p$ and $n_s$ degree-2 untransmitted variable nodes of classes $1p'$ and $2p'$ are introduced to the protograph. Whenever two such variable nodes are transmitted, a code of lower rate is obtained. The number of degree-2 variable nodes introduced to the graph through the application of the check node splitting technique is determined by the desired rates.

Let us denote the number of transmitted and untransmitted variable nodes in the root-protograph by $n_t$ and $n_u$, respectively, and the number of check nodes by $m_r$. Suppose we apply the check-node splitting technique $n_s$ times. Therefore, $2n_s$ additional untransmitted variable nodes and $2n_s$ additional check nodes are introduced to the protograph. If we select and transmit $2n_{t,s}, n_{t,s} \leq n_s$ variable nodes, then achievable rates of the designed code family is given by

$$R_c = 1 - \frac{m_r + 2n_{t,s} - n_u}{n_t + 2n_{t,s}}, \quad n_{t,s} = 1, 2, \ldots, n_s$$

(5.13)

For instance, if $n_t = 16$, $n_u = 0$, and $m_r = 8$, then $R_c = 0.5$. For $n_{t,s} = 1, 2, 3$, we obtain $R_c = 0.44, 0.40, 0.36$, respectively.

The second design method is summarized as follows:

1. Design a root protograph code of rate $R_r \leq 0.5$. Set the minimum variable node degree to at least 3 to ensure that the designed code has linear minimum distance growth property.

2. Extend the root protograph by adding a pair of connected variable and check nodes to each side of the protograph. The degrees of the new variable nodes are $l^v \geq 3$ and $r^v \geq 3$ and the degrees of the new check nodes are $l^c \geq l^v + 1$ and $r^c \geq r^v + 1$. The check node connections to the rest of the graph are chosen in a way to decrease the achievable threshold of the overall protograph.
3. Split the new check nodes $n_s$ times. This will produce $n_s$ degree-2 untransmitted variable nodes.

4. Obtain a new rate by converting the status of an untransmitted variable node to a transmitted one from each side of the protograph. Repeat this step until the desired rate is achieved.

5. Select the lifting factor based on the desired codeword length and the circulants so that the small cycles are avoided when the graph is lifted.

6. Lift the protograph using the selected circulants to obtain the code of desired length.

The proposed methods differ in a number of ways. For example, the construction technique used in Method I is similar to constructing codes with Incremental Redundancy.
such as Raptor codes [88], while Method II requires introducing degree-2 variable nodes. As we mentioned in Chapter 2, degree-2 variable nodes usually decrease the convergence speed of iterative decoding. Furthermore, they decrease the typical linear minimum distance ratio of the code. However, they also decrease the minimum achievable threshold of the code which is a desired property because this results in achieving higher coding gains.

**Design Example 1:**

Let us illustrate Method I with an example. If it is decided that the protograph is symmetric with respect to both frames (i.e., the protographs corresponding to Frame 1 and Frame 2 are identical), it is sometimes helpful to embed high performance protographs that do not have root structure into the protograph corresponding to both frames in Figure 5.9. Note that the variable nodes of classes $1i$ and $2i$ should be connected to their corresponding root checks. A similar approach is applied in [2] where unstructured LDPC codes are embedded into the parity-check matrices of root-LDPC code.

However, it is important to select variable and check nodes of different classes so that the resulting protograph would perform well. Further, since the extended protograph is a new code, its performance is not necessarily similar to the performance of the original protograph and usually it is required to modify the extended protograph to obtain the desired performance.

The first design example follows the method described above. Figure 5.11(a) illustrates the protograph of a root-LDPC code $C_1$ of rate $R_{1,r} = 0.5$ that is constructed by embedding two rate-1/2 R4JA codes [75] into the construction of Figure 5.9. The resulting root protograph has a symmetric construction and consists of two variable nodes and two check nodes of each class. Although this protograph has two degree-2 variable nodes, its minimum distance grows linearly with the block length because it can be shown that both degree-2 variable nodes can be obtained by applying the check-node splitting technique to a graph with no variable nodes of degree less than 3 and since the remaining variable nodes in the graph also have degree at least 3, this protograph has the linear minimum distance growth property.

The graph is extended in Figure 5.11(b) to obtain the root protograph of the cooperative code $C_{1,c1}$ of rate $R_{1,c1} = 0.4$. As it is shown in this figure, the root-protograph is
Figure 5.11: An example of a family of RCPB-R-LDPC codes constructed using Method I.

extended by adding a variable node of class $p'_1$ to Frame 1 and a variable node of class $p'_2$
to Frame 2 which are connected to the rest of graph through two new check nodes, i.e., 1c and 2c, respectively. In this example, both variable nodes have degree 3. The check node connections, however, are chosen so that the checks are connected to both variable nodes of class 1i and of class 2i. The presented permutations in this example are selected through simulating a number of possible connections.

The protograph of Figure 5.11(b) is further extended in Figures 5.11(c-e) to obtain the root protographs of the cooperative codes $C_{1,c2}$, $C_{1,c3}$, and $C_{1,c4}$ with overall rates $R_{1,c2} = 0.33$, $R_{1,c3} = 0.28$, and $R_{1,c4} = 0.25$, respectively. For all extensions, variable nodes of degree 3 or higher are chosen. Furthermore, check nodes connections to parity bits of classes 1p and 2p are also considered and the linear minimum distance growth is preserved in all cases.

**Design Example 2:**

In this example, construction method II is employed to design a family of RCPB-R-LDPC codes. At the beginning, a root-protograph code $C_{2,r}$ is designed which has 16 variable nodes and 8 check nodes and has rate $R_{2,r} = 0.5$ (Figure 5.13 (a)). Then the protograph of the lower rate code $C_{2,c1}$ is obtained by adding a variable node of degree 3 and a check node to each side of the root-protograph. The extended protograph has rate $R_{2,c1} = 0.44$ (Figure 5.13 (b)).

To obtain codes of lower rates, the check node splitting technique is employed. In this example, we obtain two more codes whose rates are calculated using Eq. (5.13). Thus, the check node splitting techniques is applied to check nodes 1c and 2c twice to obtain two additional degree-2 untransmitted variable nodes in each frame. As it is shown in Figure 5.13 (c), the new variable nodes are of classes $p_1'$ and $p_2'$. Furthermore, the new check nodes are of class 1c and 2c, respectively. It is required here to optimize the connections of the new check nodes. In this example, check node connections are chosen through simulations.

The second cooperative code $C_{2,c2}$ is obtained by transmitting variable nodes $A_1$ and $B_1$ in Frame 1 and Frame 2, respectively. This code has rate $R_{2,c2} = 0.40$. Finally, if we transmit variable nodes $A_2$ and $B_2$, a new cooperative code $C_{2,c3}$ of rate $R_{2,c3} = 0.36$ is obtained. This procedure can be repeated to obtain codes of rates $R_{2,c4} = 0.33$, $R_{2,c5} = 0.30$ and so on.
5.4 Analysis of Outage Probability

The outage probability of the block fading channel is defined as

\[ P_o = \Pr(I(h, \gamma) < R) \]  

(5.14)
where $I(h, \gamma)$ is the instantaneous mutual information of the channel as a function of fading coefficient $h$ and the SNR $\gamma$. Assuming BPSK signaling and given the coefficient $h$, the instantaneous mutual information between the channel input $\mathcal{S}$ and its output $\mathcal{R}$ in the presence of AWGN noise can be obtained using the result given in [89] as

$$I(\mathcal{S}; \mathcal{R}|h) = E_{\mathcal{R}|h} \left[ \log_2 \left( 1 + \exp \left( \frac{-2\gamma h}{\sigma^2} \right) \right) \right]$$

(5.15)

where $E[\cdot]$ is the mathematical expectation. It is necessary to study the outage behavior of cooperative communication to evaluate the performance of the designed codes. Outage probability limit is a full-diversity curve and it is required that full-diversity codes would approach the outage probability limit regardless of the codeword length [90], [91]. The outage probability limit depends on the signaling technique and the cooperative scheme employed by the users. In our model only the message of one user is transmitted in each block where one of the two cases mentioned earlier in this chapter may occur. Thus, the outage event related to each user in this scheme is the union of the outage events of both cases. Assuming BPSK signaling, the outage events of user 1 may be calculated using the following Proposition:

**Proposition 5.4.1.** The outage event of the above cooperative scheme under BPSK signa-
ing is given as:
\[ E_o = \left[ \left( I_{12} > \frac{R}{1 - \beta} \right) \cap (I_{10}(1) < R) \right] \]
\[ \cup \left[ \left( I_{12} < \frac{R}{1 - \beta} \right) \cap (I_{10}(2) < R) \right] \]  
(5.16)

where \( I_{10}(1) \) is \( I_{10} \) in case 1 and \( I_{10}(2) \) is \( I_{20} \) in case 2. Further,
\[ I_{12} = 1 - E_{R\mid h_{12}} \left[ \log_2 \left( 1 + \exp \left( \frac{-2r_{12}h_{12}}{\sigma_{12}^2} \right) \right) \right] , \]  
(5.17)
\[ I_{10}(1) = 1 - (1 - \beta) E_{R\mid h_{10}} \left[ \log_2 \left( 1 + \exp \left( \frac{-2r_{10}h_{10}}{\sigma_{10}^2} \right) \right) \right] \]
\[ - \beta E_{R\mid h_{20}} \left[ \log_2 \left( 1 + \exp \left( \frac{-2r_{20}h_{20}}{\sigma_{20}^2} \right) \right) \right] , \]  
(5.18)
\[ I_{10}(2) = 1 - E_{R\mid h_{10}} \left[ \log_2 \left( 1 + \exp \left( \frac{-2r_{10}h_{10}}{\sigma_{10}^2} \right) \right) \right] \]  
(5.19)

Proof. The expression of \( E_o \) is the union bound of the outage events in Case 1 and Case 2. The expression of \( I_{12} \) follows from (5.16). The equations for \( I_{10}(1) \) and \( I_{10}(2) \) are obtained from the capacity results for parallel Gaussian channels where the capacities add together. The weights of the terms in the expression for \( I_{10}(1) \) are due to the cooperation of the users. \( \square \)

The outage probability is then given as
\[ P_o = \int \int \int_{E_o} p(h_{12}, h_{10}, h_{20}) dh_{12} dh_{10} dh_{20} \]  
(5.20)
where \( p(h_{12}, h_{10}, h_{20}) \) is the joint probability density function of the channel coefficients.

### 5.5 Simulation Results

In this section, we evaluate the performance of the designed codes in two different scenarios over the BF channel through finite length simulations using codes with block length close to 5000. This length is chosen to make a fair comparison with codes designed in [2] possible. However, since in protograph-based codes, the block length is equal to the length of the protograph multiplied by the lifting number, it was not possible to obtain a block length of 5000 for all rates in our examples. However, as we will see, the block length in all cases is between 4994 and 5008.
A) Scenario 1:

In this scenario, we assume that the average SNRs on the source-destination and relay-destination links are equal. Furthermore, it is assumed that the interuser channel is perfect which means that the source-relay link is error free. As a result, the success rate of decoding the source user’s message in the relay user is 1 and cooperation is always possible.

We employ the codes constructed using the first method in this scenario, i.e., the $C_1$ code family. Figure 5.16 illustrates the performance of code $C_{1,c2}$ (Figure 5.11 (c)) with $R_{1,c2} = 0.33$, the regular (3, 9, 3, 6) RCR-LDPC code of rate 0.33 reported in [2] which is an unstructured code of length 5000 (Figure 5.15), and a rate 0.33 protograph-based code with no root structure. The protograph of this code is shown in Figure 5.14. Code $C_{1,c2}$ has been lifted 417 times and the resulting code has length 5004 bits whereas the protograph of Figure 5.14 has been lifted 833 times and the resulting code has length 4998. In both cases, circulant permutations are used and the circulants are chosen so that the parity-check matrix has no cycles of length 4.

Our simulations indicate that the protograph-based code with no root structure is not full-diversity, while the codes with root structure achieve a diversity of order 2 and have a similar performance.

We then calculated the BPSK outage probability of the cooperative model for all of the rates that correspond to the $C_1$ code family by performing Monte Carlo simulations and compared the results with the performance of the $C_1$ code family. For this purpose, the protographs of Figure 5.11 have been lifted first with the following lifting numbers and
Figure 5.15: The parity-check matrix of the regular (3, 9, 3, 6) RCR-LDPC code of rate 0.33 reported in [2] where the dots represent 1’s.

derived lengths:

- Protograph of Figure 5.11 (a) ($R_{1,c1} = 0.50$): Lifting number = 625; code length = 5000.
- Protograph of Figure 5.11 (b) ($R_{1,c2} = 0.40$): Lifting number = 500; code length = 5000.
- Protograph of Figure 5.11 (c) ($R_{1,c3} = 0.33$): Lifting number = 417; code length = 5004.
- Protograph of Figure 5.11 (d) ($R_{1,c4} = 0.28$): Lifting number = 357; code length = 4998.
Figure 5.16: Comparison of the performance of three rate-0.33 codes: The regular (3, 9, 3, 6) RCR-LDPC code reported in [2]; the code $C_{1,c2}$ of Figure 5.11(c); and the protograph-based code of Figure 5.14 with no root structure.

- Protograph of Figure 5.11 (e) ($R_{1,c5} = 0.25$): Lifting number = 313; code length = 5008.

Then the derived codes have been simulated over the cooperative model of Scenario 1 to obtain the corresponding word error-rate (WER) curves. The performance of each code has been compared with the corresponding BPSK outage probability and the results are illustrated in Figures 5.17, 5.18, and 5.19.

The results indicate that all of the codes in this family achieve second order diversity. Furthermore, they perform within 1.5 dB of the outage probability limit.

B) Scenario 2:

In this scenario, we assume that the average SNRs on the source-destination and relay-destination links are equal. However, the interuser channel is not assumed to be error free in
this case. Instead, we assume that the SNR of this link is 5 dB higher than the SNR of the user-destination link. Thus, there are instances where the source user transmits the second frame of its message to BS directly. However, in most cases cooperation would be possible.

We have evaluated the performance of \( C_2 \) code family of Figures 5.12 and 5.13 under this scenario. This code family consists of three different rates and Monte Carlo simulations have been used to calculate the BPSK outage probability of the cooperative model at each rate.

For \( C_2 \) code family, the following lifting numbers and derived lengths have been used:

- Protograph of Figure 5.12 (a) \((R_{2,c1} = 0.50)\): Lifting number = 313; code length = 5008.
- Protograph of Figure 5.12 (b) \((R_{2,c2} = 0.44)\): Lifting number = 278; code length = 5004.
- Protograph of Figure 5.13 \((R_{2,c3} = 0.36)\): Lifting number = 227; code length = 4994.

The WER curves of the above codes under Scenario 2 have been compared with the outage probability limit at the corresponding rates in Figures 5.20 and 5.21. We observe that even under this scenario, the designed codes achieve diversity of order 2. In fact, the designed code approach the outage probability limit by performing within 1.3 dB from this limit.

5.6 Conclusions

In this chapter, we introduced rate-compatible protograph-based root LDPC codes for cooperative communication and proposed two methods to construct these codes. Since these constructions are based on protograph designs, they benefit from simple encoding and parallel architecture of the decoder which enables implementing high-speed decoders.

The proposed techniques are based on the extension technique and offer broad design rates. Furthermore, they are based on protograph constructions with minimum distance
growing linearly with the block length, a property that often improves the error floor performance of the designed codes.

The BPSK outage probability limit was also obtained for the cooperative scheme employed in this work where the results were compared with the WER performance of the designed codes. It was observed that these codes achieve a diversity of order 2 as a result of their root structures and they approach the outage probability limit.
Figure 5.17: Performance results for finite length $C_1$ codes of rates 0.50 and 0.40 under Scenario 1 where the interuser channel is error free.
(a) Comparison of the performance of the outage probability and the rate-0.33 code.

(b) Comparison of the performance of the outage probability and the rate-0.28 code.

**Figure 5.18:** Performance results for finite length $C_1$ codes of rates 0.33 and 0.28 under Scenario 1 where the interuser channel is error free.
(a) Comparison of the performance of the outage probability and the rate-0.25 code.

(b) Performance results for $C_1$ code family.

**Figure 5.19**: Performance results for finite length $C_1$ code of rate 0.25 under Scenario 1 and comparison of all rates of $C_1$ code family.
Figure 5.20: Performance results for finite length $C_2$ codes of rates 0.50 and 0.44 under Scenario 2 where the SNR of the interuser channel is 5 dB higher than the SNR of the user-BS channel.
(a) Comparison of the performance of the outage probability and the rate-0.33 code.

(b) Performance results for \( C_2 \) code family.

**Figure 5.21**: Performance results for finite length \( C_2 \) code of rate 0.36 under Scenario 2 and comparison of all rates of \( C_2 \) code family.
Chapter 6

Conclusion

In this dissertation, we address code design problem for cooperative communication over different channel models with emphasis on low complexity designs and structured codes that are attractive for practical implementation.

We started with the problem of designing efficient codes for the relay node in Gaussian relay channels. For a class of capacity approaching codes for this channel model, called bilayer lengthened LDPC (BL-LDPC) codes, we calculated a measure of decoding complexity as a function of the number of decoding iterations $N$. We proposed a technique to design complexity-optimized BL-LDPC codes by minimizing the complexity measure of these codes. This was made possible by generalizing the EXIT charts to the case of BL-LDPC codes. Our technique fixes the rate-optimized code of the destination node and targets minimizing the decoding complexity of the relay code. This work was motivated by the fact that there are usually stricter hardware restrictions at the relay node. Furthermore, excessive delay due to decoding high rate codes at the relay results in additional delay at the destination. Using our technique, we designed bilayer codes with noticeable reduction in decoding complexity and delay compared to the rate-optimized codes reported in the literature.

Furthermore, we designed a wide class of complexity-optimized BL-LDPC codes with different upper check node degrees in their bilayer graphs and studied their measure of complexity over a wide range of channel noise thresholds. Our results show that these codes require considerably fewer number of iterations to converge compared to the rate-optimized codes reported in the literature. When simulated using codes with long block lengths, these
codes outperform the rate-optimized codes, in terms of bit error rates, at mid-to-high signal to noise ratios.

Next, we studied the achievable rates for the DF relaying strategy for the Rayleigh fading relay channel where the links have independent normalized Rayleigh fading coefficients and the channel side information is perfectly known at the corresponding receivers but not at the transmitters. We designed BL-LDPC codes for this scenario for the case when the source-relay link is much stronger than the source-destination link as well as for the case when these two links have comparable SNRs.

We also proposed a novel two-user cooperation scheme for the block fading channel model that employs protograph-based LDPC codes. The proposed scenario is based on time division where each user transmits its message to the base station (BS) in two successive frames. In Frame 1, the source user sends its message to the BS as well as to its partner user. When the partner user successfully decodes the received signal, the two users cooperatively send the source user’s message using the Alamouti scheme in Frame 2. Otherwise, the main user simply retransmits its message to the BS in Frame 2. The users change roles during the next two frames. Additionally, the users encode their information over a class of protograph-based LDPC codes called Accumulate-Repeat-4-Jagged-Accumulate (AR4JA) codes that allow flexible selection of rates and code lengths. Using this scheme, a diversity order of 3 is achieved when the interuser link is strong, whereas when the interuser channel has a poor quality, the diversity order is 2.

Finally, we introduced rate-compatible protograph-based root LDPC (RCPB-R-LDPC) codes for cooperative communication over block fading channels and proposed two methods to construct these codes. Since these constructions are based on protograph designs, they benefit from simple encoding and parallel architecture of the decoder which enables implementing high-speed decoders. The proposed techniques are based on the extension technique and offer broad design rates resulting in high flexibility. Furthermore, they are based on protograph constructions with minimum distance growing linearly with the block length, a property that improves the error floor performance of the designed codes. The outage probability limit under BPSK modulation is obtained for the cooperative scheme employed in this work and was used to evaluate the WER performance of the designed codes. It was observed that as a result of their root structure, these codes achieve order-2 diversity and
approach the outage probability limit.

There are several directions for future work. For instance, BL-LDPC codes are a special case of multi-edge type LDPC (MET-LDPC) codes. An interesting problem is to derive a measure of decoding complexity for MET-LDPC codes and furthermore, to formulate an optimization problem to design complexity-optimized MET-LDPC codes.

Moreover, the methodology to design complexity-optimized BL-LDPC codes that was proposed for Gaussian relay channels can be extended to Rayleigh fading relay channels. This would be a natural extension to the work presented in Chapters 2 and 3.

The cooperation strategy proposed in Chapter 4 is based on SNR accumulation and results in logarithmic increase of the mutual information with the received power from the source and relay. By employing the RCPB-R-LDPC codes designed in Chapter 5 and modifying the proposed cooperative strategy accordingly, it is possible to obtain a method that is based on information accumulation which results in linear increase of the mutual information with the received power. For example, a possible strategy would be the following: User 1 transmits the first part of its message in Frame 1. When user 2 successfully decodes user 1’s message, it can obtain the second part of user 1’s message. Therefore, both users employ a space-time code (such as the Alamouti scheme) to transmit the second part of user 1’s message in Frame 2. Otherwise, user 1 continues with transmitting the remaining part of its message in Frame 2, and so on.

The techniques proposed in Chapter 5 to design RCPB-R-LDPC codes result in designing full-diversity codes. An interesting problem is to optimize the Shannon thresholds of the designed protograph-based root codes. This may result in codes with a smaller performance gap with the outage probability limit. Even though embedding protograph codes with excellent performance in Gaussian channels into constructions presented in Chapter 5 results in RCPB-R-LDPC codes that approach the outage probability limit, no methodology has been proposed so far to optimize the performance of a coding scheme with root structure in the Shannon sense. In fact, designing rate-compatible structured and unstructured LDPC codes with root structure that approach the Shannon capacity on the Gaussian channel is a challenging problem and tackling this problem would be an interesting direction for future work.
Bibliography


