2018

On The Design Of Physical Layer Rateless Codes

Amrit Kharel
University of Mississippi

On the Design of Physical Layer Rateless Codes

A Dissertation
presented to the Department of Electrical Engineering
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy
at the
University of Mississippi

Amrit Kharel

May 2018

ABSTRACT

Codes that are capable of generating any number of encoded symbols from a given number of source symbols are called rateless codes. Luby transform (LT) codes are the first practical realization of rateless codes while Raptor codes are constructed by serially concatenating LT codes with high-rate outer low-density parity-check (LDPC) codes. Although these codes were originally developed for binary erasure channel (BEC), due to their rateless feature, they are being investigated and designed for their use in noisy channels. It is known that LT codes are the irregular non-systematic rateless counterpart of low-density generator-matrix (LDGM) codes. Therefore, the first part of our work is focused on LDGM codes and their serially concatenated scheme called serially concatenated LDGM (SCLDGM) codes. Though single LDGM codes are asymptotically bad codes, the SCLDGM codes are known to perform close to the Shannon limit. We first study the asymptotic behaviour of LDGM codes using a discretized density evolution (DDE) method. We then show that the DDE method can be used in two-steps to provide the detailed asymptotic performance analysis of SCLDGM codes. We also provide the detailed error-floor analysis of both the LDGM and SCLDGM codes. We also prove a necessary condition for the successful decoding of such concatenated codes under sum-product (SP) decoding in binary input additive white Gaussian noise (BIAWGN) channels. Based on this necessary condition, we then develop a DDE-based optimization approach which can be used to optimize such concatenated codes in general. We present both the asymptotic performance and simulation results of our optimized SCLDGM codes that perform within 0.26 dB to the Shannon limit in BIAWGN channels.

Secondly, we focus on the asymptotic analysis and optimization design of LT and Raptor codes over BIAWGN channels. We provide the exact asymptotic performance of LT codes using the DDE method. We apply the concept of the two-step DDE method to the Raptor codes and obtain their exact asymptotic performance in BIAWGN channels. We show that the existing Raptor codes using solely the same output degree distribution can perform within 0.4 dB to the Shannon limit for various realized code-rates. We then develop a DDE-based optimization technique to optimally design such physical layer Raptor codes. Our optimized Raptor codes are shown to perform within 0.2 dB to the Shannon limit for most of the realized code-rates. We also provide the asymptotic curves, decoding thresholds, and simulation results showing that our optimized Raptor codes outperform the existing Raptor codes in BIAWGN channels. Finally, we present the asymptotic analysis and optimization design of systematic version of these codes namely systematic LT and systematic Raptor codes as well.

LIST OF ABBREVIATIONS

APP a posteriori probability. 16

BEC binary erasure channel. ii, 1

BER bit error rate. 1

BIAWGN binary input additive white Gaussian noise. ii, 2

BPA belief propagation algorithm. 7, 16

BPSK binary phase shift keying. 19

CNs check nodes. 14

DDE discretized density evolution. ii, 2

DE density evolution. 2, 20

i.i.d independent and identically distributed. 21

LDGM low-density generator-matrix. ii, 2

LDPC low-density parity-check. ii, 1

LLR log-likelihood ratio. 17

LR likelihood ratio. 17

LT Luby transform. ii, 1

MPA message passing algorithm. 16

ODD output degree distribution. 4, 6

PCLDGM parallel concatenated LDGM. 27

PDF probability density function. 20

PMF probability mass function. 9

RSD robust soliton distribution. 9

SCLDGM serially concatenated LDGM. ii, 2, 27

SLT systematic LT. iii, 5, 70

SNR signal to noise ratio. 20

SP sum-product. ii, 2

SPA sum-product algorithm. 2, 16

SR systematic Raptor. iii, 5, 70

VNs variable nodes. 14

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my advisor Dr. Lei Cao for the continuous support, guidance, and encouragement during the course of my Ph.D. studies. I am especially grateful for his insightful suggestions and inputs over numerous discussions that encouraged me to accomplish the research objectives.

I am grateful to Dr. John N. Daigle and Dr. Ramanarayanan "Vish" Viswanathan for all the valuable discussions we had during our departmental presentation sessions. I am thankful to Dr. Feng Wang for consenting to be a member of my dissertation defense committee.

This work was supported in part by NASA EPSCoR program under grants NNX13AB31A and NNX14AN38A.

Lastly, I would like to thank my family. Without their love and continuous support, I would not have succeeded. I am thankful to my loving and caring wife Rojina with whom I shared all the joy and difficulties, and ups and downs of Ph.D. studies.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Background

Codes that are capable of generating any number of encoded symbols (bits) from given number of source symbols are called rateless codes. Let k and n represent the number of source and encoded symbols respectively. Rateless means code rates of any positive real number between 0 and 1 can be realized on the fly, i.e., the code-rate can simply be changed from k/n to $k/(n+1)$ by just generating one more encoded symbol on the fly which is equivalent to adding one more column in the generator (G) matrix of the code. Fountain codes [1, 2] such as Luby transform (LT) codes [3] and Raptor codes [4] are rateless codes. These codes have near Shannon performance in binary erasure channel (BEC). Due to their rateless property, these codes are being investigated and improved for their use in noisy channels as well [5, 6, 7, 8, 9, 10, 11]. These codes were first investigated over noisy channels in [5, 6], and it was reported that LT codes perform very poorly and suffer huge error-floor. It was also mentioned that the Raptor codes which are constructed by serially concatenating LT codes with a high-rate error-correcting precode such as low-density parity-check (LDPC) codes [12] can actually enhance the overall bit error rate (BER) performance. Since then these codes have attracted a lot of research interests for their use over noisy channels. Since LT codes are the inner codes of Raptor codes, improving these inner codes plays a great role in designing better Raptor codes.

1.2 Motivation and Objective

The decoding algorithm used by rateless codes over binary input additive white Gaussian noise (BIAWGN) channel is sum-product algorithm (SPA)[12], which was originally developed for LDPC codes. discretized density evolution (DDE) analysis [13, 14], which is a computationally efficient discretized version of density evolution (DE) [15] that exactly tracks the behaviour of SPA was developed for the asymptotic analysis and design of LDPC codes [13, 14, 16]. With the help of this DDE method, optimized irregular LDPC codes that perform extremely close to the Shannon limit are successfully developed [13, 14, 16]. The DDE method is also used to asymptotically analyse the generator-matrix based error correcting codes called low-density generator-matrix (LDGM) codes [17] that also use SPA as the decoding algorithm [18, 19]. Due to some differences in the decoding graph between LDPC and LDGM codes, the DDE method developed for LDGM codes slightly differ from those developed for LDPC codes. However, the central idea of evolution of message densities as the sum-product (SP) decoding progresses remains the same. Like LDGM codes, LT codes are also sparse generator-matrix based codes. Moreover, LT codes are known to be irregular non-systematic rateless counterpart of the LDGM codes. Like LDGM and LDPC codes, LT codes also use SPA as their decoding algorithm when used over noisy channels [6, 20]. Therefore, the successful implementation of DDE for LT codes is very important to analyse the asymptotic behaviour of such codes. Furthermore, extending DDE method for concatenated codes is of huge importance. It is known that the serially concatenated LDGM (SCLDGM) codes have near Shannon performance. Furthermore, the Raptor codes which are the serial concatenation of LT codes and high-rate LDPC codes are known to have better performance than LT codes. Such serially concatenated codes have two-stages of encoding and two-stages of decoding as shown in Figure 3.3. During the decoding of such codes, first the decoding of inner code is conducted and then the decoding of outer code is conducted. For both the SCLDGM and Raptor codes, both the inner and outer decoders use the SP

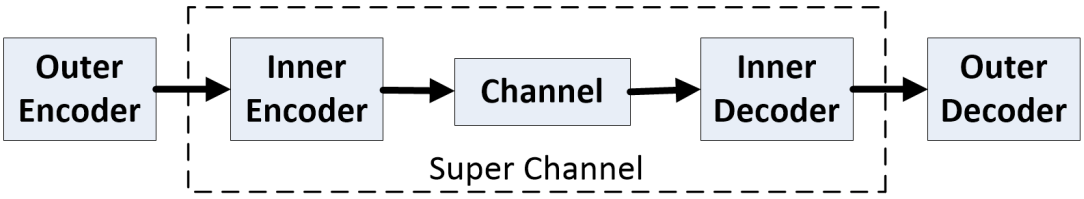

Figure 1.1. Concatenated Codes

decoding. Hence, in order to obtain the exact asymptotic performance of such concatenated codes, we should be able to implement DDE in two-stages to exactly track the two-stage SP decoding. The successful implementation of such two-step DDE method further allows us to develop DDE-based optimization approach to design capacity approaching error-correcting concatenated codes. Accomplishing these tasks can potentially lead us to design optimized physical layer error-correcting rateless codes that are capacity approaching.

The overall objective of our work is to design capacity approaching rateless codes for noisy channels. Since the Raptor codes are concatenated codes and have LDGM-based LT codes as inner codes, we first thoroughly study and analyse the SCLDGM codes. We develop two-step DDE method for the asymptotic analysis of SCLDGM codes and also develop an optimization technique to optimally design such concatenated codes in general. After that, we accomplish the following tasks regarding the physical layer Raptor codes: (a) implementation of the two-step DDE for the Raptor codes to analyse their asymptotic behaviour in BIAWGN channels, (b) development of a DDE-based optimization method to design Raptor codes for BIAWGN channels that perform very close to the Shannon limit, (c) analysis and design of systematic version of these physical layer rateless codes.

1.3 Contributions and Outline

Chapter 2 provides some details on LT and Raptor codes for BEC. It also discusses about LDPC codes, their decoding algorithm called SPA, and the DDE method that is used to exactly track the behaviour of SPA.

Chapter 3 completely focuses on LDGM and SCLDGM codes. We give some background on these codes. Since LDGM codes also use SP decoding, the DDE method is used to obtain their asymptotic performance. The main contribution of this chapter is two-fold. First, we show that since the SCLDGM codes are decoded in two-stages using SP decoding, their exact asymptotic performance can be obtained by using the DDE method in two-stages as well. We develop a two-step DDE method in which the DDE is first applied to the inner code and then to the outer code to obtain the asymptotic performance of the SCLDGM codes. We also provide the detailed error-floor analysis of both the LDGM and SCLDGM codes. The second contribution is about the optimization design of such concatenated codes. We theoretically prove a necessary condition for the successful decoding of such concatenated codes. The necessary condition indicates that the inner decoder must at least produce a certain BER called critical BER so that the outer decoder can produce an extremely small BER required for the successful decoding. It means that handing decoding to the outer decoder without inner decoder achieving the critical BER is just the waste of decoding. We finally propose a critical-BER based DDE optimization approach to design capacity approaching SCLDGM codes. We also present the asymptotic curves and simulation results of our optimized SCLDGM codes.

Chapter 4 focuses on the analysis and design of physical layer Raptor codes. It discusses about the implementation of DDE for the Raptor codes which have a rateless inner LT code. We develop a two-step DDE method for such rateless codes and use it to provide the detailed asymptotic analysis of the Raptor codes in BIAWGN channels. We also give the asymptotic performance of LT codes when used alone. We show that Raptor codes of various realized code-rates can use the same output degree distribution (ODD) and yet provide decoding performance within 0.4 dB to the Shannon limit. We also show that the necessary condition for the successful decoding of SCLDGM codes also holds true for the Raptor codes. We then develop and use the critical-BER based optimization approach to design optimized

Raptor codes. We show that the optimized Raptor codes that perform within 0.2 dB to the Shannon limit can be designed for various realized code-rates. We provide the asymptotic curves, decoding thresholds, and simulation results showing that our optimized Raptor codes outperform the existing Raptor codes. Finally, in this chapter, we extend our investigation to the systematic version of these codes. We provide the detailed asymptotic performance of systematic LT (SLT) and systematic Raptor (SR) codes. We also design the optimized SR codes that largely outperform the existing SR codes.

Finally, chapter 5 concludes our work.

CHAPTER 2

LT, RAPTOR AND LDPC CODES

In this chapter, we present an overview of LT and Raptor codes considering BEC. We also provide some background on LDPC codes, their decoding algorithm used in BIAWGN channels, and the discretized density evolution method used to analyse the asymptotic performance of such codes.

2.1 LT Codes

LT codes, presented in [3], were developed by Michael Luby as random rateless codes for erasure channels. These codes are called rateless codes due to the fact that any number of encoded symbols can be generated from the given source symbols as needed and on the fly. The original k source symbols can be recovered from any set of $n > k$ encoded symbols with high probability. The output degree distribution used for generating the encoded symbols lies at the heart of LT codes. They have sparse generator matrices, i.e., the number of source symbols contributing to an encoded symbol is relatively small compared to the total number of source symbols. A simple graphical representation of LT codes is presented in Figure 2.1, where $s_1, s_2, \cdots, s_k$ represent the source symbols while $c_1, c_2, \cdots, c_n$ represent the encoded symbols. A source symbol is of arbitrary length, from one-bit binary symbol to l-bit symbol.

2.1.1 Encoding Process

For a given set of k source symbols, an encoded symbol is generated by bitwise XORing of a randomly chosen subset of source symbols. The process of generating an encoded symbol is as follows:

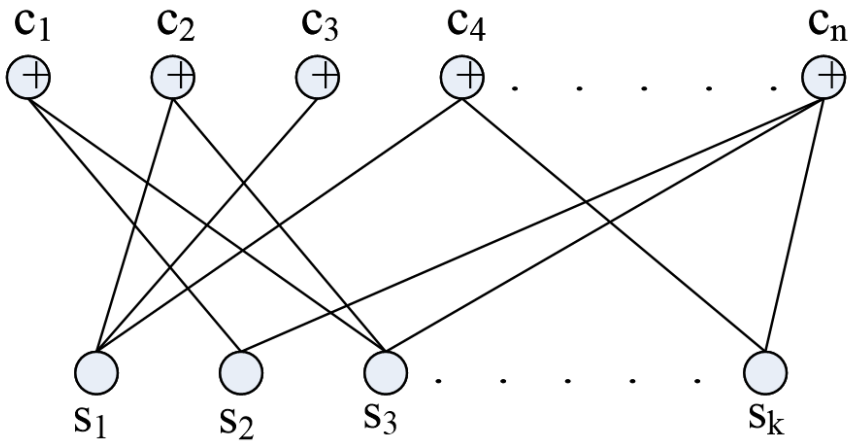

Figure 2.1. Graphical Representation of an LT code

- An integer d between 1 and k called the *degree* of an encoded symbol is randomly chosen for each encoded symbol based on an ODD.
- A set of d source symbols is chosen uniformly at random from the set of k source symbols. These d source symbols are called the neighbors of the encoded symbol to be generated.
- The d source symbols are bitwise XOR-ed to produce an encoded symbol of degree d.

By this process, an arbitrary number of encoded symbols can be generated. Each encoded symbol is generated independently.

2.1.2 Decoding Process

Decoding algorithms such as belief propagation algorithm (BPA) or peeling solvers are commonly used for the decoding of LT codes over BEC [3]. The decoding steps include

1. The decoder identifies all encoded symbols of degree one (those connected to a single source symbol) in the bipartite graph representing LT codes such as Figure 2.1. If there is at least one encoded symbol that has exactly one neighbor, then the neighbor

is recovered immediately since it is a copy of the encoded symbol. If there exists no degree one encoded symbols, then the decoding fails.

2. The encoded symbols of degree one and the associated edges are erased from the graph.

3. The value of the recovered source symbol is XORed with any remaining encoded symbols that also have that source symbol as a neighbor. The corresponding edges are erased from the graph thus decreasing the degree of such neighboring encoded symbols by one. The encoded symbols whose degree is reduced to one during the decoding process are called the reduced degree-one encoded symbols.

4. The decoder repeats steps $1-3$ using the new reduced graph.

If the decoder fails to recover all the k source symbols from the n received encoded symbols, a decoding failure occurs. Failure can occur due to either of the following reasons:

1. There exist source symbols that are not connected to any encoded symbol in the bipartite graph.

2. Absence of reduced degree-one encoded symbols at any stage of the decoding until the successful completion of the decoding process.

Few key terms regarding the decoding of LT codes in BEC are as follows

- **Recovered source symbol**: Initially, all the source symbols are unknown at the decoder. During the decoding process as described in Section 2.1.2, more and more source symbols are decoded gradually. Therefore, any source symbol that is decoded is called the recovered source symbol.

- **Reduced degree-one encoded symbol**: When a source symbol is recovered, it is XORed with every encoded symbol it is connected to and its all the connected edges in the decoding graph are erased. During this process, any encoded symbol whose degree gets reduced to one is called reduced degree-one encoded symbol.

- **Ripple**: A set of source symbols connected to reduced degree-one encoded symbols is called the ripple. At the beginning of the decoding process, all the sources symbols which are neighbors of degree-one encoded symbols are in the ripple.

- **Premature termination**: If the ripple size ceases to zero before the completion of the decoding process, then such situation is termed as premature termination. This can also be understood as the situation where reduced degree-one encoded symbols are absent. In such condition, using the decoding process as described in Section 2.1.2, no remaining source symbols can be recovered even though they are connected to encoded symbols.

2.1.3 Output Degree Distribution

The ODD of an LT code gives the probability of generating degree-d, $d = 1, 2, \cdots, k$ encoded symbols. A specially designed ODD called robust soliton distribution (RSD) is proposed in [3]. The RSD ensures that the average number of reduced degree-one encoded symbols is large enough at each stage during the decoding process ensuring the successful termination of decoding process with high probability.

The RSD is represented by $\mu(\cdot)$. For constants $c > 0$ and $\delta \in (0, 1]$, the probability mass function (PMF) $\mu(\cdot)$ is given by

$$\mu(i) = \frac{\rho(i) + \tau(i)}{\beta}, \quad \text{for } 1 \leq i \leq k, \tag{2.1}$$

where

$$\beta = \sum_{i=1}^{k} \left(\rho(i) + \tau(i)\right). \tag{2.2}$$

The PMFs $\rho(i)$ and $\tau(i)$ are given by

$$\rho(i) = \begin{cases} \frac{1}{k}, & \text{for} \quad i = 1, \\ \frac{1}{i(i-1)}, & \text{for} \quad 2 \leq i \leq k, \end{cases} \tag{2.3}$$

$$\tau(i) = \begin{cases} \frac{S}{ik}, & \text{for} \quad 1 \leq i \leq \frac{k}{S} - 1, \\ \frac{S \ln(\frac{S}{\delta})}{k}, & \text{for} \quad i = \frac{k}{S}, \\ 0, & \text{otherwise.} \end{cases} \tag{2.4}$$

The parameter S represents the average number of degree-one encoded symbols and is defined as

$$S = c \cdot \sqrt{k} \cdot \ln\left(\frac{k}{\delta}\right). \tag{2.5}$$

It is suggested in [3] that the k source symbols can be recovered from any set of n encoded symbols with probability of at least $1 - \delta$, and n is given by

$$\begin{aligned} n &= k\beta \\ &= k + c \cdot \sqrt{k} \cdot \ln^2(k/\delta). \end{aligned} \tag{2.6}$$

As shown above, the RSD has two components: $\rho(\cdot)$ and $\tau(\cdot)$. The support of $\rho(i)$ is extending over the entire range of degrees from 1 through k while $\tau(i)$ is restricted to i in the range 1 through k/S. The distribution $\mu(i)$ has the spikes at two points: $i = 2$ and $i = k/S$. The spike in $\tau(i)$ at $i = k/S$ is to ensure that each source symbol is highly likely to be connected to at least one encoded symbol. The plots for $\rho(\cdot)$, $\tau(\cdot)$, and $\mu(\cdot)$ are shown in Figures 2.2, 2.3, and 2.4 respectively.

2.1.4 Overhead in LT codes

If $n = k + h$ encoded symbols are required for successfully decoding k source symbols, then h is called the overhead. The fractional overhead ϵ is defined as

$$\epsilon = \frac{n}{k} - 1 = \frac{h}{k}. \tag{2.7}$$

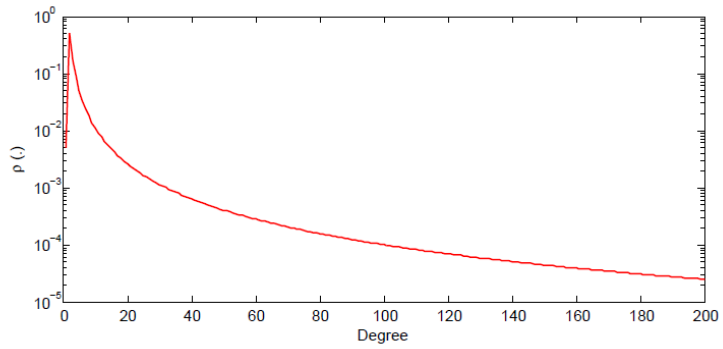

Figure 2.2. $\rho(\cdot)$ **component of RSD for** $k = 200, c = 0.05, \delta = 0.5$**.**

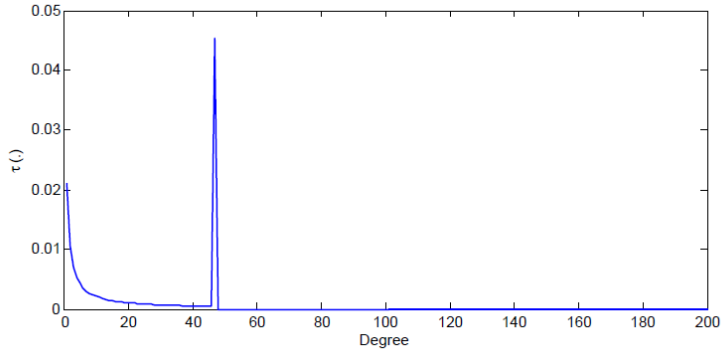

Figure 2.3. $\tau(\cdot)$ **component of RSD for** $k = 200, c = 0.05, \delta = 0.5$**.**

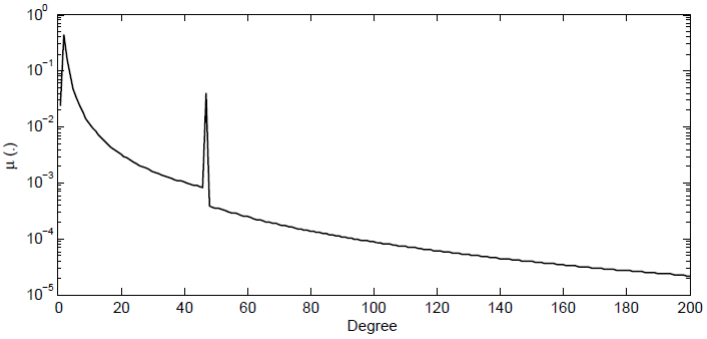

Figure 2.4. RSD $\mu(\cdot)$ **for** $k = 200, c = 0.05, \delta = 0.5$**.**

2.2 Raptor Codes

Raptor codes [4] are extensions of LT codes, and thus are rateless codes. It is constructed by serially concatenating an LT code with a high-rate LDPC code. These codes were de-

veloped with the objective of improving the encoding and decoding complexity. Raptor codes require a small constant average number of symbol operations per encoded symbol generated, and a similar small constant number of symbol operations per source symbol recovered. Thus, these codes achieve a linear encoding and decoding complexity within a constant factor.

2.2.1 Encoding and Decoding

Raptor codes are normally constructed using a high-rate code to precode the source symbols. These pre-coded symbols are called intermediate symbols. Then, a suitable LT-code with constant average output degree is applied to the intermediate symbols to produce the encoded symbols. As shown in Figure 2.5, at first the source symbols represented as $\boldsymbol{s} = [s_1, s_2, \cdots, s_k]$ are encoded using a high rate LDPC code to produce intermediate symbols represented as $\boldsymbol{m} = [m_1, m_2, \cdots, m_{k'}],\ k' > k$, where $(k' - k)$ is a small fraction of k. Since the LDPC precode used is mostly systematic in nature, the first k intermediate symbols are the original source symbols, and the last $(k' - k)$ intermediate symbols are called the redundant symbols. These intermediate symbols are further encoded using an LT code to produce as many LT encoded (output) symbols as required. Let $\boldsymbol{o} = [o_1, o_2, \cdots, o_n],\ n > k$ represent n such output symbols. There are $(k' - k)$ constraints that define the relationship between the source symbols and the redundant symbols among the intermediate symbols. These constraints are viewed as symbols called *constraint symbols*. The value of each constraint symbol is zero, i.e., the constraint symbol constrains the sum of its neighboring intermediate symbols to be equal to zero.

During the decoding of Raptor codes, once the LT decoder finishes its operation, a small fraction of the intermediate symbols may still be unrecovered. If the precode is chosen appropriately, then this set can be recovered using an erasure decoding algorithm for the precode. For example, if the precode is capable of correcting up to a δ fraction of erasures among the intermediate symbols, then the LT decoder only needs to recover a $(1 - \delta)$ fraction

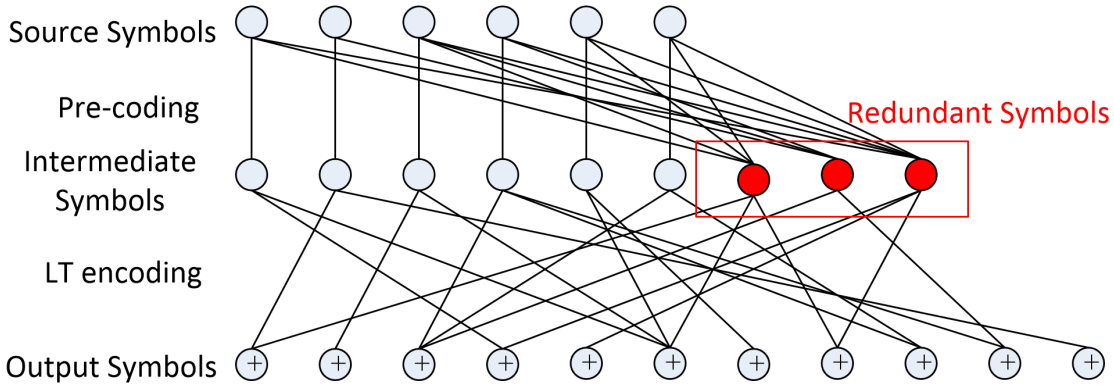

Figure 2.5. Graphical representation of a Raptor code.

of the intermediate symbols from the received output symbols.

2.2.2 Output Degree Distribution

The ODD associated with LT codes is represented as $\Omega(x) = \sum_{i=1}^{k} \Omega_i x^i$, where Ω_i is the probability of generating a degree-i LT encoded symbol and $\sum_{i=1}^{k} \Omega_i = 1$. An optimization technique is provided in [4] to obtain optimized degree distributions with an average output degree of $O(\log(1/\epsilon))$, thus making the encoding and the decoding cost constant. An example of such distribution is

$$\begin{aligned}\Omega_{\mathrm{T}}(x) = .007969x^1 + .49357x^2 + .1662x^3 + .072646x^4 + .082558x^5 + \\ .056058x^8 + .037229x^9 + .05559x^{19} + .025023x^{65} + .003135x^{66}\end{aligned} \tag{2.8}$$

The average output degree of (2.8) is 5.87, and is independent of k.

2.3 LDPC Codes

LDPC codes are a class of linear block codes that provide near capacity performance on a large collection of data transmission and storage channels while simultaneously admitting implementable decoders. LDPC codes were first proposed by Gallager in his 1960 doctoral

dissertation [12]. The study of LDPC codes was resurrected almost after 35 years with the work of MacKay, Luby, and others [21, 22].

2.3.1 Representation of LDPC Codes

The LDPC code is given by the null space of a $(n-k) \times n$ parity-check (H) matrix that has a low density of 1s. The k and n represent the number of original source bits and encoded bits respectively. Tanner in 1981 generalized the LDPC codes and introduced a graphical representation of these codes, now called Tanner graphs [23]. The two types of nodes in a Tanner graph are variable nodes (VNs) and check nodes (CNs). The Tanner graph of a code is drawn according to the following rule: A CN $\mathrm{C_j}$ is connected to a VN $\mathrm{V_i}$ whenever the element h_{ji} in H-matrix is a 1. So, there are $n-k$ CNs, one for each check equation, and n VNs, one for each encoded bit. An example of H-matrix for a $(2,4)$-LDPC code is given as (2.9). The corresponding Tanner graph is shown in Figure 2.6. For each CN, the sum of the neighboring VNs is zero in GF(2), i.e., the value of each CN is always zero.

$$H = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix} \tag{2.9}$$

The number of edges connecting to a node is called the degree of the node. An LDPC code is called *regular* if all its VNs have the same degree, and all the CNs have the same degree. Otherwise, it is *irregular* LDPC code.

2.3.2 Degree Distributions

Degree distributions associated with CNs and VNs of LDPC codes govern the performance of these codes. Let $\Omega(x) = \sum_{i=1}^{d_r} \Omega_i x^i$ be the degree distribution function associated

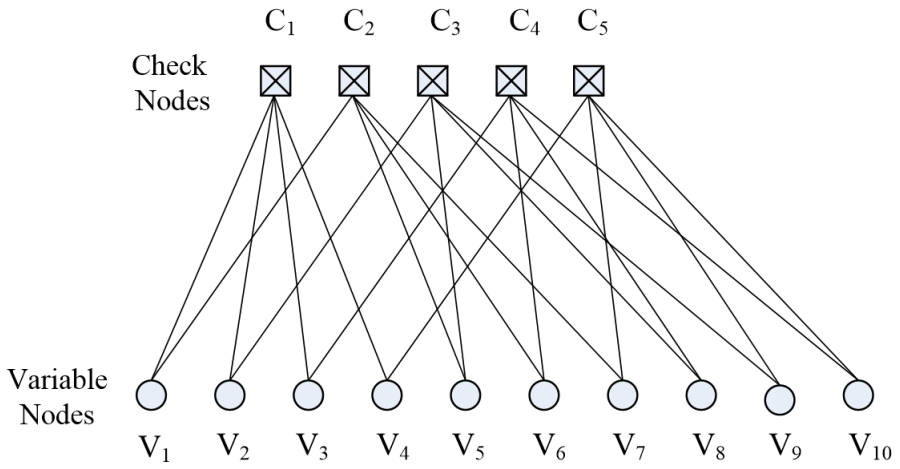

Figure 2.6. Tanner graph for an example LDPC code.

with the CNs, where Ω_i represents the fraction of degree i CNs, d_r is the maximum CN degree, and $\sum_{i=1}^{d_r} \Omega_i = 1$. Let $\Lambda(x) = \sum_{i=1}^{d_l} \Lambda_i x_i$ be the degree distribution function associated with the VNs, where Λ_i represents the fraction of degree i VNs, d_l represent the maximum VN degree, and $\sum_{i=1}^{d_l} \Lambda_i = 1$.

The total number of edges in the decoding graph can be calculated as

$$N_e = (n-k)\sum_{j=1}^{d_r} j\Omega_j = n\sum_{i=1}^{d_l} i\Lambda_i. \tag{2.10}$$

Let $\omega(x)$ and $\lambda(x)$ be the degree distributions associated with the CNs and VNs from the edge perspective respectively. They are represented as [12]

$$\begin{aligned} \omega(x) &= \sum_{i=1}^{d_r} \omega_i x^{i-1}, \\ \lambda(x) &= \sum_{i=1}^{d_l} \lambda_i x^{i-1}, \end{aligned} \tag{2.11}$$

where ω_i is the probability that an uniformly chosen edge is connected to a CN of degree i. Similarly, λ_i is the probability that an uniformly chosen edge is connected to a VN of degree

i. ω_i and λ_i are calculated as

$$\omega_i = \frac{i\Omega_i}{\sum_{j=1}^{d_r} j\Omega_j},$$
$$\lambda_i = \frac{i\Lambda_i}{\sum_{j=1}^{d_l} j\Lambda_j}. \tag{2.12}$$

Therefore, (2.12) can be rewritten as

$$\omega(x) = \frac{\Omega'(x)}{\Omega'(1)},$$
$$\lambda(x) = \frac{\Lambda'(x)}{\Lambda'(1)}, \tag{2.13}$$

where $\Omega'(x)$ and $\Lambda'(x)$ represent the derivative of $\Omega(x)$ and $\Lambda(x)$ respectively.

2.3.3 Iterative Decoding Algorithm

In addition to introducing LDPC codes, Gallager also provided a decoding algorithm that is typically near optimal. Since that time, other researchers have independently discovered that algorithm and related algorithms. These algorithms perform iterative computations in graph-based models and comes under different names depending on the context. These names include: the sum-product algorithm, the belief propagation algorithm, and the message passing algorithm (MPA). The term "message passing" usually refers to all such iterative algorithms, including the SPA, BPA, and its approximations. The concept of such algorithm is as follows:

The a posteriori probability (APP) that a given bit in a transmitted codeword $\boldsymbol{c} = [c_1 c_2 \cdots c_n]$ equals 1, given the received word $\boldsymbol{y} = [y_1 y_2 \cdots y_n]$, is computed initially. So, for the decoding of bit c_i, the APP is computed as

$$Pr(c_i = 1|y). \tag{2.14}$$

The APP ratio (also called likelihood ratio (LR)) is calculated as

$$l(c_i) \triangleq \frac{Pr(c_i = 0|y)}{Pr(c_i = 1|y)}. \tag{2.15}$$

The more numerically stable computation of the log-APP ratio, also called the log-likelihood ratio (LLR) is calculated as

$$L(c_i) \triangleq \log\left(\frac{Pr(c_i = 0|y)}{Pr(c_i = 1|y)}\right). \tag{2.16}$$

Hereafter, the natural logarithm is assumed for LLRs.

The SPA for the computation of $Pr(c_i = 1|y)$, $l(c_i)$, or $L(c_i)$ is an iterative algorithm which is based on the code's Tanner graph. Specifically, it is imagined that the VNs represent processors of one type, CNs represent processors of another type, and the edges represent message paths. In one-half iteration, each variable node processes its input messages and passes its resulting output messages up to neighboring CNs (two nodes are said to be neighbors if they are connected by an edge). This is depicted in Figure 2.7 for the message $m_{\uparrow 14}$ from VN V_1 to CN C_4 (the subscripted arrow indicates the direction of the message, keeping in mind that the Tanner graph convention places CNs above VNs). The information passed concerns $Pr(c_1 = b \mid \text{input messages})$, $b \in \{0, 1\}$, the ratio of such probabilities, or the logarithm of the ratio of such probabilities. Note in the figure that the information passed to CN C_4 is all the information available to VN V_1 from the channel and through its neighbors, excluding CN C_4, i.e., only extrinsic information is passed. Such extrinsic information $m_{\uparrow ij}$ is computed for each connected VN-CN pair V_i/C_j at each half-iteration.

In the other half iteration, each CN processes its input messages and passes its resulting output messages down to its neighboring VNs. This is depicted in Figure 2.8 for the message $m_{\downarrow 14}$ from CN C_1 down to VN V_4. The information passed concerns $Pr(\text{check equation } C_1 \text{ is satisfied} \mid \text{input messages})$, $b \in \{0, 1\}$, the ratio of such probabilities, or the logarithm of

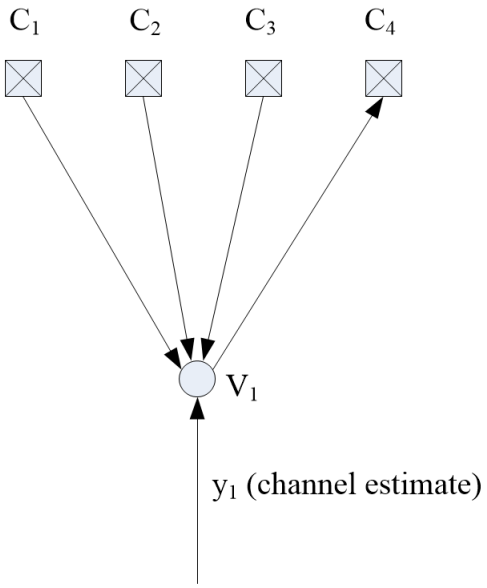

Figure 2.7. Subgraph of a Tanner graph corresponding to an H matrix whose first column is $[1\ 1\ 1\ 1\ 0\ \cdots\ 0]$**. The arrows indicate message passing from node** V_1 **to node** C_4**.**

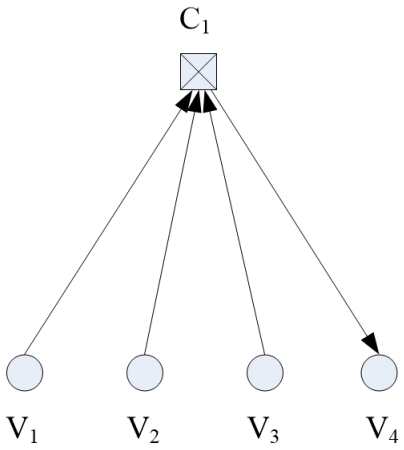

Figure 2.8. Subgraph of a Tanner graph corresponding to an H matrix whose first row is $[1\quad 1\quad 1\quad 1\quad 0\quad \cdots\quad 0]$**. The arrows indicate message passing from node** C_1 **to node** V_4**.**

the ratio of such probabilities. Note that only extrinsic information is passed to VN V_4 as in the previous case. Such extrinsic information $m_{\downarrow ji}$ is computed for each connected CN-VN pair C_j/V_i at each half-iteration.

After a prescribed maximum number of iterations or after some stopping criterion has

been met, the decoder computes the APP, the LR, or the LLR for each VN from which the decision of each transmitted bit c_i is made.

2.3.4 Decoding of LDPC codes

This section presents the decoding of LDPC codes over BIAWGN channel. The modulation assumed is the binary phase shift keying (BPSK) modulation. The channel is modelled as $y = x + w$, where y represents the received corrupted version of the transmitted codeword, x represents the modulated codeword, and w represents a vector of zero-mean white Gaussian noise samples with variance $\sigma^2 = N_0/2$, where N_0 is the one-sided noise power spectral density. The channel estimates are calculated as $2y/\sigma^2$, which are used to start the decoding process.

Sum-Product Algorithm

The SPA is run over the decoding graph associated with the LDPC codes constructed using the associated H-matrix. The messages are exchanged between VNs and CNs along the connected edges. Let $L_{\mathrm{C_jV_i}}$ represent the LLR message that the CN $\mathrm{C_j}$ passes to its neighboring VN $\mathrm{V_i}$. Similarly, let $L_{\mathrm{V_iC_j}}$ represent the LLR message that the VN $\mathrm{V_i}$ passes to its neighboring CN $\mathrm{C_j}$. Let $L_{\mathrm{V_i}}$ represent the channel estimate associated with the VN $\mathrm{V_i}$. Hence, $L_{\mathrm{V_i}} = 2y_i/\sigma^2$. Initially, each VN passes its channel estimate to all of its neighboring CNs through the connecting edges. Then, the updates $L_{\mathrm{C_jV_i}}$ and $L_{\mathrm{V_iC_j}}$ are exchanged at each iteration which are calculated as

$$L_{C_jV_i} = 2\tanh^{-1}\left(\prod_{i'\in N_{C_j}-\{i\}} \tanh\left(\frac{1}{2}L_{V_{i'}C_j}\right)\right), \tag{2.17}$$

$$L_{V_iC_j} = L_{V_i} + \sum_{j'\in N_{V_i}-\{j\}} L_{C_{j'}V_i}, \tag{2.18}$$

where N_{C_j} denotes the set of VNs connected to the jth CN and N_{V_i} denotes the set of CNs connected to the ith VN.

We call (2.17) and (2.18) as *product rule* and *sum rule* respectively. This algorithm is run over some predefined number of iterations or until certain stopping criteria is met. After that, the final decision about each VN is made based on the following rule which we call as *decision rule*.

$$L_D = L_{V_i} + \sum_{j' \in N_{V_i} - \{j\}} L_{C_{j'} V_i}. \tag{2.19}$$

2.3.5 Density Evolution for LDPC Codes

A general tool to analyze LDPC codes that uses an iterative sum-product decoder is called density evolution [15, 16]. It tracks the evolution of the probability density function (PDF) of the messages being passed through the edges of the decoding graph, where the messages are modelled as random variables. Using this method, one can find the minimum signal to noise ratio (SNR) or the maximum noise parameter (σ^2) beyond which the iterative decoder fails to achieve an extremely small bit error probability. This value is called the decoding threshold.

Discretized Density Evolution

Discretized density evolution [13, 14] is a computationally efficient discretized version of the density evolution which considers discretized LLR messages. The DDE accurately tracks the behaviour of the SP decoding. The method is explained here.

Let v be a LLR message through a randomly chosen edge from a degree d_v VN to a CN. Under SP decoding v is calculated as

$$v = \sum_{i=0}^{d_v - 1} u_i, \tag{2.20}$$

where $u_i,\ i = 1, \cdots, d_v - 1$ are the incoming LLRs from the neighboring CNs of the VN except the CN that gets the message v, and u_0 is the observed LLR of the VN.

Let u be a LLR message from a degree d_c CN to a VN. Then, the message update rule

for the CN is calculated as

$$\tanh\frac{u}{2}=\prod_{j=1}^{d_c-1}\tanh\frac{v_j}{2}, \tag{2.21}$$

where $v_j,\ j=1,\cdots,d_c-1$ are the incoming LLRs from the neighboring VNs of the CN except the VN that gets the message u.

To perform the DDE, the messages are discretized in the following way. Let $Q(w)$ be the quantized message of w, i.e.,

$$Q(w)=\begin{cases}\left\lfloor\frac{w}{\Delta}+\frac{1}{2}\right\rfloor\Delta, & \text{if } w\geq\frac{\Delta}{2}\\ \left\lceil\frac{w}{\Delta}-\frac{1}{2}\right\rceil\Delta, & \text{if } w\geq-\frac{\Delta}{2}\\ 0, & \text{otherwise;}\end{cases} \tag{2.22}$$

where Q is the quantization operator, Δ is the quantization interval, $\lfloor x\rfloor$ is the largest integer not greater than x, and $\lceil x\rceil$ is the smallest integer not less than x.

Discretized SP decoding is defined as SP decoding with all input and output messages quantized in this way. Under discretized SP decoding, (2.20) becomes $\bar{v}=\sum_{i=0}^{d_v-1}\bar{u}_i$, where $\bar{v}=Q(v)$ and $\bar{u}_i=Q(u_i)$ for $i=0,\cdots,d_v-1$. The PMF of a quantized message $\bar{w}$ is denoted as $p_{\bar{w}}[k]=\Pr(\bar{w}=k\Delta)$ for $k\in\mathbb{Z}$. Then, the PMF for the quantized message $\bar{v}$ is calculated as [14]

$$p_{\bar{v}}=\bigotimes_{i=0}^{d_v-1}p_{\bar{u}_i}, \tag{2.23}$$

where $\bigotimes$ is defined as the convolution operation while the superscript represents the number of times the convolution is operated. Since the $\bar{u}_i$ are independent and identically distributed (i.i.d) for $1\leq i\leq d_v$, (2.23) is written as

$$p_{\bar{v}}=p_{\bar{u}_0}\bigotimes\left(\bigotimes^{d_v-1}p_{\bar{u}}\right), \tag{2.24}$$

where $p_{\bar{u}}=p_{\bar{u}_i}, 1\leq i\leq d_v$.

Let a two-input operator R be defined as

$$R(a,b) = Q\left(2\tanh^{-1}\left(\tanh\frac{a}{2}\tanh\frac{b}{2}\right)\right), \tag{2.25}$$

where a and b are quantized messages. Using this operator, the quantized message $\bar{u}$ of (2.21) is calculated as

$$\bar{u} = R(\bar{v}_1, R(\bar{v}_2, \cdots, R(\bar{v}_{d_c-2}, \bar{v}_{d_c-1})). \tag{2.26}$$

The PMF of $R(a,b)$ denoted as p_c is calculated as

$$p_c[k] = \sum_{(i,j):k\Delta = R(i\Delta, j\Delta)} p_a[i]p_b[j]. \tag{2.27}$$

For simplicity, (2.27) is written as $p_c = R(p_a, p_b)$.

Since for any CN, all the incoming messages from its neighboring VNs have the same PMF, we can write $p_{\bar{v}} = p_{\bar{v}_i}$ for any $1 \leq i \leq d_c$. Thus, we write the following

$$\begin{aligned} p_{\bar{u}} &= R(p_{\bar{v}}, R(p_{\bar{v}}, \cdots, R(p_{\bar{v}}, p_{\bar{v}}))) \\ &= R^{d_c-1}p_{\bar{v}}. \end{aligned} \tag{2.28}$$

We know that the message densities get evolved as the iteration number increases. Let, $p_{\bar{u}}^{(l)}$ and $p_{\bar{v}}^{(l)}$ be the PMFs corresponding to $\bar{u}$ and $\bar{v}$ respectively at the completion of the lth iteration.

We define, for any PMF p,

$$\begin{aligned} f_\lambda(p) &= \sum_{i=2}^{d_v} \lambda_i \left(p_{\bar{u}_0} \bigotimes \left(\bigotimes^{i-1} p\right)\right), \\ f_\omega(p) &= \sum_{j=2}^{d_c} \omega_j \left(R^{j-1}p\right). \end{aligned} \tag{2.29}$$

The $f_\lambda(p)$ and $f_\omega(p)$ are in fact the average PMFs of the message $\bar{u}$ and $\bar{v}$ respectively.

Then, the evolution of message densities $p_{\bar{v}}^{(l)}$ and $p_{\bar{u}}^{(l)}$ can be calculated iteratively using following formulas

$$\begin{aligned} p_{\bar{v}}^{(l)} &= f_{\lambda}\left(f_{\omega}\left(p_{\bar{v}}^{(l-1)}\right)\right), \\ p_{\bar{u}}^{(l)} &= f_{\omega}\left(f_{\lambda}\left(p_{\bar{u}}^{(l-1)}\right)\right), \end{aligned} \tag{2.30}$$

where the initial PMFs $p_{\bar{v}}^{(0)}$ and $p_{\bar{u}}^{(0)}$ have all mass at zero.

These formulations are used in the design of capacity approaching LDPC codes. Given a code-rate, designing LDPC codes means finding a degree distribution pair $\lambda(x)$ and $\omega(x)$ such that $p_{\bar{u}}^{(l)} \to 0$ as $l \to \infty$ for the minimum SNR or the maximum σ.

Using this DDE method, an optimized half-rate irregular LDPC code has been designed with a decoding threshold within 0.0045 dB to the Shannon limit [13, 14].

CHAPTER 3

DESIGN OF SERIALLY CONCATENATED LDGM CODES

In this chapter, we first provide some background on LDGM codes and its concatenated schemes. We then completely focus on the asymptotic performance analysis, error-floor analysis, and optimization design of SCLDGM codes using DDE method for BIAWGN channels.

3.1 LDGM Codes

LDGM codes are systematic linear codes with low-density generator matrices. Because of the sparseness of the generator matrix, the amount of processing required in the encoder is comparable to that of turbo codes and much less than that of the standard LDPC codes [17]. These codes also use SPA as the decoding algorithm. Hence, their decoding complexity is as good as that of LDPC codes and much better than Turbo codes.

3.1.1 Encoding

Let a packet to be transmitted over a communication channel is of length k-bits. These k-bits called source/information bits are encoded to produce a code-word of length n-bits which after modulation are transmitted over a communication channel. The ratio k/n is the code-rate of the code. Let $\boldsymbol{s} = [s_1, s_2, \cdots, s_k]$ be the original k-bits. Due to the systematic nature of LDGM encoding, the encoded bits also called as output bits consist of k original source bits and $(n-k)$ parity bits. Therefore, the output bits are represented

as $\boldsymbol{o} = [s_1, s_2, \cdots, s_k, p_1, p_2, \cdots, p_{n-k}]$. The generator matrix used to generate such output bits is represented as G = [I P], where I represents $k \times k$ identity matrix and P represents $(n-k) \times k$ sparse matrix used to generate parity bits $[p_1, p_2, \cdots, p_{n-k}]$. The graphical representation of a LDGM code is presented in Figure 3.1. A LDGM code is called a regular (X, Y)-LDGM code if the number of edges connected to each source and parity bit are X and Y respectively. Therefore, the code represented by the Figure 3.1 is a regular $(3, 3)$ LDGM code.

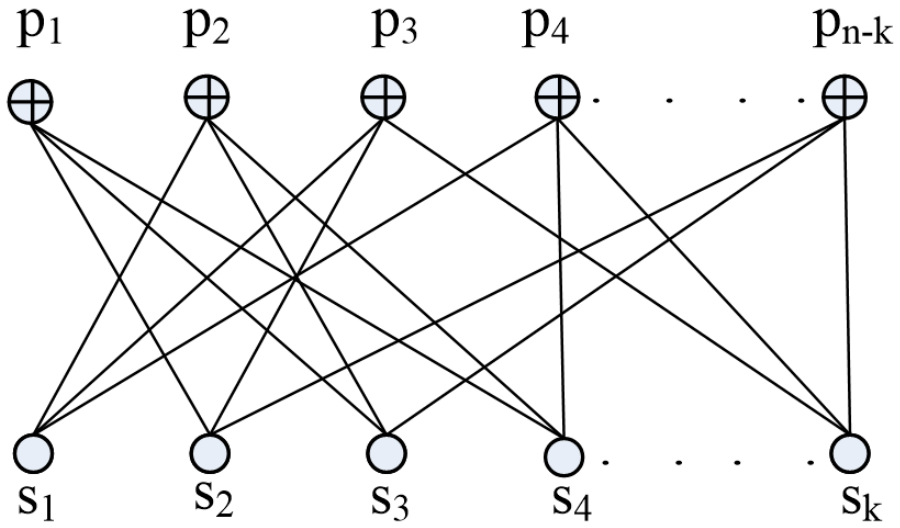

Figure 3.1. An example of a LDGM code.

3.1.2 Decoding of LDGM Codes over BIAWGN channel

A bi-partite graph as shown in Figure 3.2 is used to conduct SP decoding of LDGM codes in BIAWGN channels. The VNs $V_1, V_2, \cdots, V_k$ represent the original source bits while the CNs $C_1, C_2, \cdots, C_{n-k}$ represent the parity bits. We have considered the BPSK modulation. The channel is modelled as $y_i = x_i + w_i$, where x_i represent the transmitted bit and w_i represent the zero mean Gaussian noise. The corresponding channel estimate is calculated as $2y_i/\sigma^2$, where σ^2 is the variance of the Gaussian noise. Let L_{V_i} and L_{C_j} represent the channel estimates associated with the ith VN and the jth CN respectively.

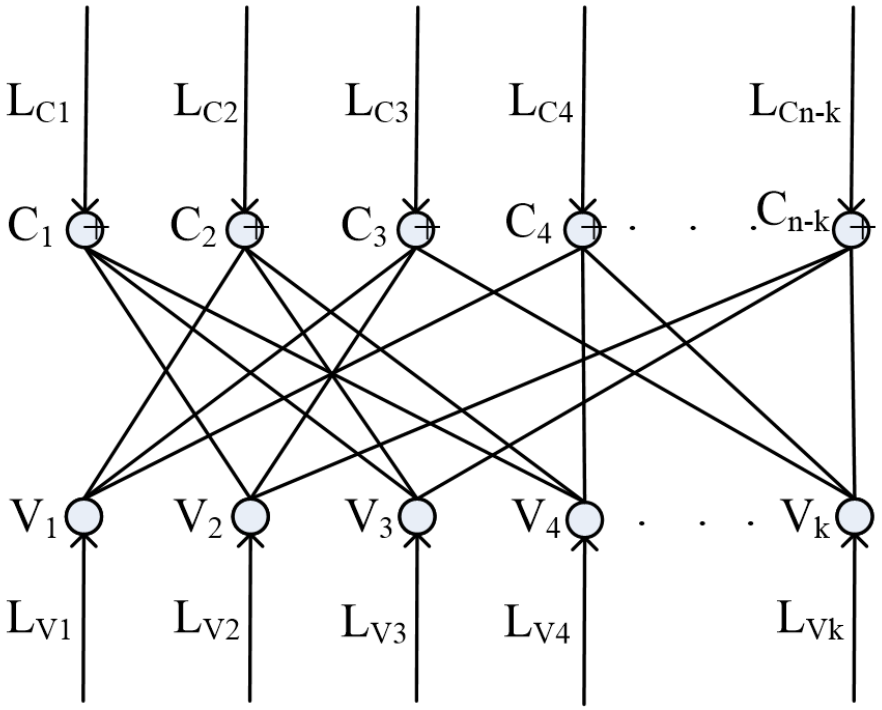

Figure 3.2. Bipartite graph used for the decoding of LDGM codes using SPA.

The product and sum rule used for the decoding of LDGM codes in BIAWGN channels are as follows

$$L_{C_jV_i} = 2\tanh^{-1}\left(\left(\tanh\frac{L_{C_j}}{2}\right)\prod_{i'\in N_{C_j}-\{i\}}\tanh\left(\frac{1}{2}L_{V_{i'}C_j}\right)\right), \tag{3.1}$$

$$L_{V_iC_j} = L_{V_i} + \sum_{j'\in N_{V_i}-\{j\}} L_{C_{j'}V_i}. \tag{3.2}$$

where N_{C_j} denotes the set of VNs connected to the jth CN and N_{V_i} denotes the set of CNs connected to the ith VN.

The decision rule used is

$$L_D = L_{V_i} + \sum_{j'\in N_{V_i}-\{j\}} L_{C_{j'}V_i}, \tag{3.3}$$

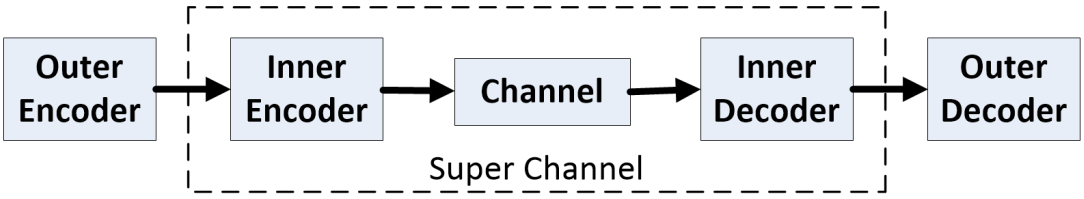

Figure 3.3. Concatenated Codes

Compared to the SPA of LDPC decoding, the sum and decision rule remain the same for the LDGM decoding while the product rule has an extra term L_{C_j} representing the channel estimate associated with the ith parity bit. The LLR messages are exchanged between VNs and CNs through their connected edges using these update rules. The decision of each VN is made based on the gathered LLRs summed using the decision rule.

3.2 Serially Concatenated LDGM codes

The concept of concatenated codes was first introduced by David Forney in his Ph.D. dissertation in 1965 [24]. These codes employ two stages of encoding and decoding [24, 25] as shown in Figure 3.3. During encoding, the original source bits are first encoded using an outer code to produce intermediate bits which are further encoded using an inner code to produce final output bits that are transmitted over a communication channel. During decoding, first the decoding of the inner code and then the decoding of the outer code is conducted. The serially concatenated LDGM codes that use the same two-stages of encoding and decoding are proposed in [17]. Although single LDGM codes are asymptotically bad [21, 17], the SCLDGM codes are shown to have impressive performance over noisy channels [17, 26]. Another concatenated scheme called parallel concatenated LDGM (PCLDGM) codes were proposed in [19]. Since SCLDGM codes are known to have better decoding performance than the PCLDGM codes, we only consider SCLDGM codes in this work.

3.2.1 Encoding

In the two stage encoding of SCLDGM codes, at first, k source bits are encoded using a high rate outer LDGM code to produce k' intermediate bits which are further encoded using an inner LDGM code to produce n output bits. Hence, the code-rates of the outer and inner codes are $\mathrm{R_O} = k/k'$ and $\mathrm{R_I} = k'/n$ respectively, making the SCLDGM code of code-rate $\mathrm{R} = \mathrm{R_I R_o} = k/n$. Let $\boldsymbol{s} = [s_1, s_2, \cdots, s_k]$ represent the original source bits. For the SCLDGM code, encoding at each stage is systematic. Therefore, the intermediate and the final output bits are $\boldsymbol{m} = [s_1, s_2, \cdots, s_k, p^o_1, \cdots p^o_{k'-k}]$ and $\boldsymbol{o} = [s_1, s_2, \cdots, s_k, p^o_1, \cdots p^o_{k'-k}, p^i_1, \cdots p^i_{n-k'}]$ respectively , where $\boldsymbol{p}^o = [p^o_1, \cdots p^o_{k'-k}]$ and $\boldsymbol{p}^i = [p^i_1, \cdots p^i_{n-k'}]$ represent outer and inner parity bits respectively. Let n_i and n_o represent the number of inner and outer parity bits. Thus, $n = k + n_i + n_o$.

3.2.2 Decoding over BIAWGN channels

We consider transmission over the BIAWGN channel using BPSK modulation. For the inner decoder, the decoding bipartite graph is constructed using inner parity bits as CNs and intermediate bits as VNs [26]. Then the SP decoding as discussed in Section 3.1.2 is conducted over the decoding graph associated with the inner code. After running SP decoding for some pre-defined number of iterations, the LLRs from the inner decoder are gathered to calculate the final LLR of each VN using the decision rule. These values are then used as the initial LLRs for the decoding of the outer code with the decoding graph that is constructed using the outer parity bits as the CNs and the source bits as the VNs [17, 26].

3.3 Asymptotic Analysis of SCLDGM Codes using DDE

In [17], simulation results are used to show that the SCLDGM codes perform close to the Shannon limit, however, no asymptotic analysis are presented for BIAWGN channels. Asymptotic analysis for the single and the parallel concatenation of regular LDGM codes are

presented in [19]. In [18], error floor and DDE predictions of SCLDGM codes are presented. It is more focused on the Gaussian approximation [14] of DE. Extrinsic information transfer (EXIT) functions [27] are used in [26] to optimally design SCLDGM codes and also provided decoding thresholds. However, none of these works presented the exact asymptotic curves for SCLDGM codes using DDE.

Both the inner and the outer codes of the SCLDGM codes use SP decoding. Knowing that the SP decoding can be exactly tracked using DDE, we can apply DDE to both the inner and outer codes. Therefore, in this section, we show how the DDE method can be applied in two stages to complete the asymptotic analysis of the SCLDGM codes. We first apply the DDE to the inner code and then to the outer code. We assume that the all-one BPSK codeword is transmitted. Therefore, the received LLRs (channel estimates) over a BIAWGN channel are known to be symmetric Gaussian distributed with mean $2/\sigma^2$ and variance $4/\sigma^2$, i.e., $\mathcal{N}\left(2/\sigma^2, 4/\sigma^2\right)$, where the variance is twice the mean and this symmetric condition has been proved to be preserved under SP decoding [15]. The channel condition E_b/N_o, noise parameter σ, and overall code-rate R are related as $E_b/N_o = 10\log_{10}\left(1/\left(2\mathrm{R}\sigma^2\right)\right)$.

3.3.1 DDE for Inner Code

Degree Distributions

For the inner LDGM code, the degree distributions of CNs and VNs from the node perspective are given as $\Omega(x) = \sum_{i=1}^{k'} \Omega_i x^i$, where $\sum_{i=1}^{k'} \Omega_i = 1$ and $\Lambda(x) = \sum_{i=1}^{n_i} \Lambda_i x^i$, where $\sum_{i=1}^{n_i} \Lambda_i = 1$ respectively. Similarly, the degree distributions of CNs and VNs from the edge perspective are given as $\omega(x) = \sum_{i=1}^{k'} \omega_i x^{i-1}$, where $\sum_{i=1}^{k'} \omega_i = 1$ and $\lambda(x) = \sum_{i=1}^{n_i} \lambda_i x^{i-1}$, where $\sum_{i=1}^{n_i} \lambda_i = 1$ respectively. The conversion between the node and edge degree distributions are done using $\omega_i = i\Omega_i / \sum_{j=1}^{k'} j\Omega_j$, and $\lambda_i = i\Lambda_i / \sum_{j=1}^{n_i} j\Lambda_j$ respectively.

Implementation

Let $\bar{v}$ be a quantized LLR message through a randomly chosen edge from a degree d_v VN to a CN and let $\bar{u}$ be that from a degree d_c CN to a VN. Under SP decoding, $\bar{v}$ is calculated

as $\bar{v} = \sum_{i=0}^{d_v-1} \bar{u}_i$, where $\bar{u}_i,\ i = 1, \cdots, d_v - 1$ are the incoming quantized LLRs from the neighboring CNs of the VN except the CN that gets the message $\bar{v}$, and $\bar{u}_0$ is the observed LLR of the VN. As in [13], the PMF of $\bar{v}$ is calculated as $p_{\bar{v}} = p_{\bar{u}_0} \bigotimes \left(\bigotimes^{d_v-1} p_{\bar{u}} \right)$, where $\bigotimes$ is discrete convolution while the superscript represents the number of times the convolution is operated, and $p_{\bar{u}}$ is the PMF of $\bar{u}$. Similarly, for the discretized decision rule $\bar{D} = \sum_{i=0}^{d_v} \bar{u}_i$, the associated PMF is $p_{\bar{D}} = p_{\bar{u}_0} \bigotimes \left(\bigotimes^{d_v} p_{\bar{u}} \right)$.

As in [18], $\bar{u}$ is calculated as $\bar{u} = R(\bar{v}_0, R(\bar{v}_1, R(\bar{v}_2, \cdots, R(\bar{v}_{d_c-2}, \bar{v}_{d_c-1}))))$, where $v_j,\ j = 1, \cdots, d_c - 1$ are the incoming LLRs from the neighboring VNs of the CN except the VN that gets the message $\bar{u}$, and $\bar{v}_o$ is the observed LLR of the CN. It is important to note here that this $\bar{v}_o$ is absent in the LDPC code. As in LDPC codes, the two-input operation R is defined as $R(a, b) = \mathbb{Q}\left(2 \tanh^{-1}\left(\tanh \frac{a}{2} \tanh \frac{b}{2}\right)\right)$, where a and b are quantized messages, and $\mathbb{Q}$ is the quantization operator. Hence, the PMF $p_{\bar{u}}$ is computed as $p_{\bar{u}} = R(p_{\bar{v}_0}, R(p_{\bar{v}}, R(p_{\bar{v}}, \cdots, R(p_{\bar{v}}, p_{\bar{v}})))) = R(p_{\bar{v}_0}, R^{d_c-1} p_{\bar{v}})$, where the PMF of $R(a, b)$ denoted as p_c is computed as $p_c[k] = \sum_{(i,j):k\Delta = R(i\Delta, j\Delta)} p_a[i] p_b[j]$, where Δ is the quantization step.

By defining $f_\lambda(p) = \sum_{i=1}^{d_v} \lambda_i \left(p_{\bar{u}_0} \bigotimes \left(\bigotimes^{i-1} p \right) \right)$ and $f_\omega(p) = \sum_{j=1}^{d_c} \omega_j \left(R(p_{\bar{v}_0}, R^{j-1} p \right)$, the evolving PMF of $\bar{u}$ and $\bar{v}$ at the lth iteration can then be calculated as $p_{\bar{v}}^{(l)} = f_\lambda \left(f_\omega \left(p_{\bar{v}}^{(l-1)} \right) \right)$ and $p_{\bar{u}}^{(l)} = f_\omega \left(f_\lambda \left(p_{\bar{u}}^{(l-1)} \right) \right)$ respectively, where the initial PMFs $p_{\bar{v}}^{(0)}$ and $p_{\bar{u}}^{(0)}$ have all mass at zero.

Probability of Decoding Error for Inner Code

For the inner code with all VNs having same degree, i.e., for a regular inner LDGM code, the associated PMF of the decision rule at the lth iteration is $p_{\bar{D}}^{(l)} = p_{\bar{u}_0} \bigotimes \left(\bigotimes^{d_v} p_{\bar{u}} \right)$. However, for irregular inner code with the maximum VN degree of d_{max}, we can calculate $p_{\bar{D}}^{(l)}$ as

$$p_{\bar{D}}^{(l)} = \sum_{i=1}^{d_{max}} \Lambda_i \left(p_{\bar{u}_0} \bigotimes \left(\bigotimes^{i} p_{\bar{u}}^{(l)} \right) \right). \tag{3.4}$$

Let $[-L_a, L_a]$ be the range of LLRs used. Thus, the quantization step is calculated as $\Delta = 2L_a/2^{n_b}$, where n_b is the number of quantization bits used. Then, the decoding error probability of the inner decoder can be calculated at the completion of each iteration as

$$E_{in}^{(l)} = \sum_{\bar{d}} p_{\bar{D}}^{(l)}\left(\bar{d}\right) \quad \text{for} \quad \bar{d} \in [-L_a, -L_a + \Delta, \cdots, 0]. \tag{3.5}$$

3.3.2 DDE for Outer Code

Let $q_{\bar{u}_0}$ be the PMF of the initial LLRs for the outer decoder. We know that after the completion of the inner decoding, the final LLRs of all the VNs of inner code are calculated using the decision rule and are then fed into the outer decoder. This means $p_{\bar{D}}^{(l)}$ serve as the PMF of the input LLRs to the outer decoder. Since the condition that the variance of $p_{\bar{D}}^{(l)}$ is twice its mean is preserved under DE, $p_{\bar{D}}^{(l)}$ is known to be $\mathcal{N}\left(M_{\bar{D}}^{(l)}, 2M_{\bar{D}}^{(l)}\right)$, where $M_{\bar{D}}^{(l)}$ is the associated mean. Knowing that $p_{\bar{D}}^{(l)}$ is the PMF of the LLRs fed into outer decoder, we can write $q_{\bar{u}_0} = p_{\bar{D}}^{(l)}$. For the outer code, let $\Omega^{(o)}(x)$ and $\Lambda^{(o)}(x)$ be the degree distributions of the CNs and VNs from the node perspective, and $\omega^{(o)}(x)$ and $\lambda^{(o)}(x)$ be those from the edge perspective respectively. Let $q_{\bar{u}}^{(l)}$ be the PMF of the message through a randomly chosen edge from a CN to a VN at the lth outer decoding iteration. For the outer code with all VNs having same degree $d_v^{(o)}$, the associated PMF of the decision variable at the lth outer iteration is $q_{\bar{D}}^{(l)} = q_{\bar{u}_0} \bigotimes \left(\bigotimes^{d_v^{(o)}} q_{\bar{u}}\right)$. However, for irregular outer code with the maximum VN degree of $d_{max}^{(o)}$, we can calculate $q_{\bar{D}}^{(l)}$ as

$$q_{\bar{D}}^{(l)} = \sum_{i=1}^{d_{max}^{(o)}} \Lambda_i^{(o)} \left(q_{\bar{u}_0} \bigotimes \left(\bigotimes^{i} q_{\bar{u}}^{(l)} \right) \right). \tag{3.6}$$

Finally, we can calculate the overall decoding error probability of the SCLDGM code as

$$E^{(l)} = \sum_{\bar{d}} q_{\bar{D}}^{(l)}\left(\bar{d}\right) \quad \text{for} \quad \bar{d} \in [-L_a, -L_a + \Delta, \cdots, 0]. \tag{3.7}$$

Throughout this chapter, unless otherwise mentioned, the number of quantization bits

used is 10, the range of LLRs used is $[-50, 50]$, and the maximum number of iterations used is 200 for all the asymptotic results presented.

3.3.3 Asymptotic Curves

We present here the exact asymptotic curves for the single LDGM codes as well as the SCLDGM codes using the DDE method. The asymptotic performance of various half-rate single LDGM codes is presented in Figure 3.4. The figure clearly depicts that the single LDGM codes have bad performance in BIAWGN channels.

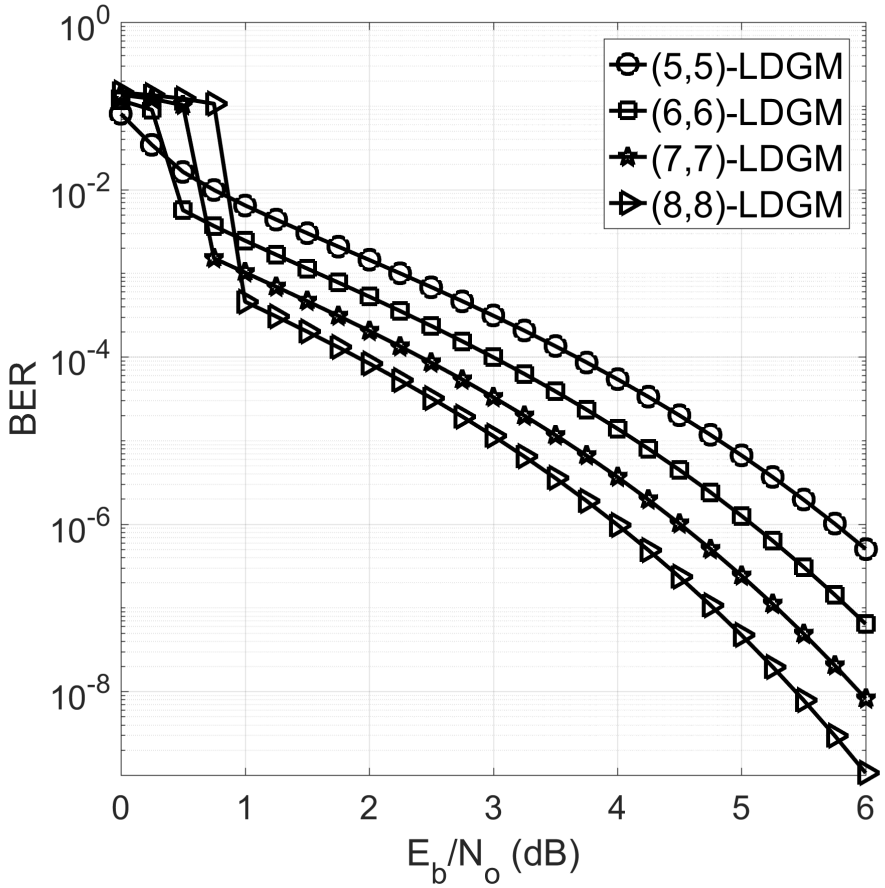

Figure 3.4. Asymptotic performance of LDGM codes. Code-rate=1/2.

Using the two-step DDE method presented in Section 3.3, we obtained the asymptotic performance of for four SCLDGM codes namely code A, B, C, and D as depicted in Figure

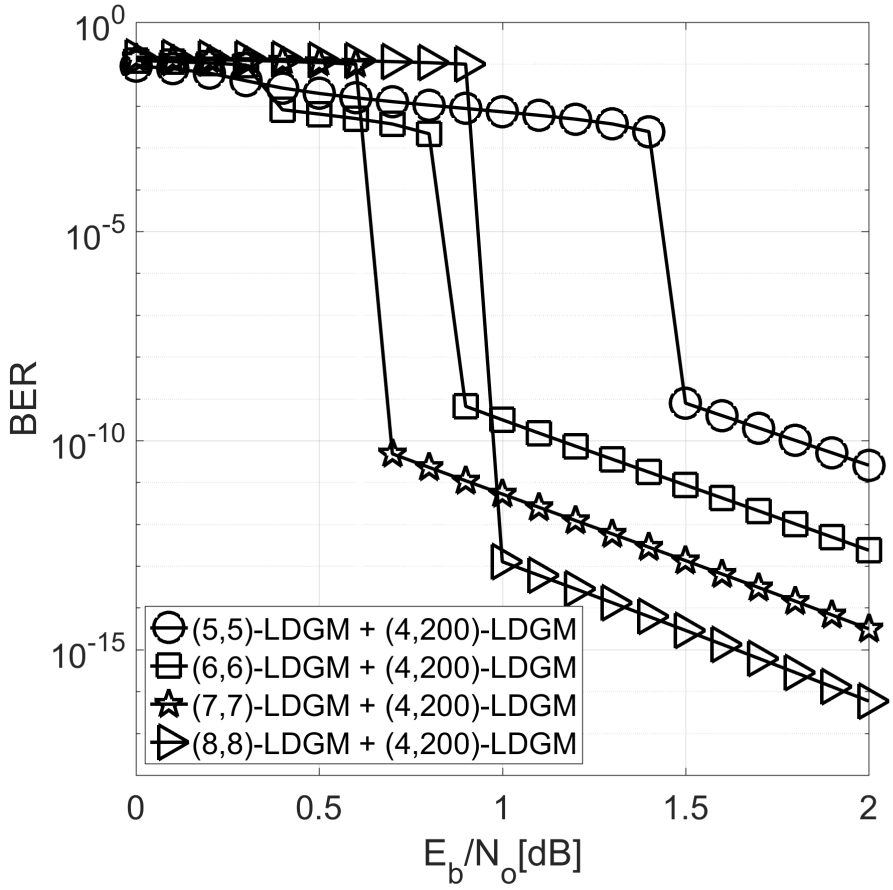

Figure 3.5. Asymptotic performance of SCLDGM codes. Inner code used is rate half (d_v, d_c)-LDGM code and outer code used is $50/51$ rate (4,200)-LDGM code. Overall code-rate (R) = 50/102.

3.5. The inner codes used for code A, B, C, and D are the regular $(5, 5)$, $(6, 6)$, $(7, 7)$, and $(8, 8)$ half-rate LDGM codes respectively while the outer code used in each case is $50/51$ rate (4,200)-LDGM code. Hence, the overall code-rate is R = $25/51$. The outer code-rate $50/51$ is chosen since it was presented as one of the best choices in [26]. We see that by using a high-rate LDGM code as an outer code, the performance of the concatenated scheme can be drastically improved as depicted for various SCLDGM codes in Figure 3.5. The decoding threshold and the gap to the Shannon limit of various SCLDGM codes are presented in Table 3.1. We see that the Code C has the best decoding threshold of 0.68 dB which is about 0.53 dB away from the Shannon limit.

Table 3.1. Decoding threshold of various SCLDGM codes. Inner code is half-rate (d_v, d_c) LDGM code and outer code is $50/51$ rate $(4, 200)$-LDGM code. Overall code-rate (R) = 50/102.

Code	Threshold (dB)	Gap (dB)
Code A = (5,5)-LDGM + (4,200)-LDGM	1.44	1.29
Code B = (6,6)-LDGM + (4,200)-LDGM	0.82	0.67
Code C = (7,7)-LDGM + (4,200)-LDGM	0.68	0.53
Code D = (8,8)-LDGM + (4,200)-LDGM	0.99	0.84

3.4 Necessary Condition for Successful Decoding

Theorem 1: A SCLDGM code using SP decoding in a BIAWGN channel can asymptotically achieve an extremely small BER if it satisfies $E_{in}^{(l)} \leq \mathrm{Q}\left(1/\sigma_{th}^{(o)}\right)$, where $E_{in}^{(l)}$ is the BER produced by the inner decoder and $\sigma_{th}^{(o)}$ is the decoding threshold of the code used as outer code.

Proof: Throughout this proof, we consider DE method where non-discretized LLR messages are considered. In the two-stage decoding as shown in Figure 3.3, we can consider the encoder-channel-decoder chain of the inner code as a super-channel [24]. Under DE, it is known that the input LLRs to the outer decoder has the PDF of $\mathcal{N}\left(M_D^{(l)}, 2M_D^{(l)}\right)$. Therefore, the equivalent noise parameter, i.e., the noise standard deviation, associated with the super-channel assuming the unit signal power, can be calculated as $\sqrt{2/M_D^{(l)}}$. Let $\sigma_{th}^{(o)}$ be the decoding threshold of the outer code at and below which the outer code can theoretically achieve an extremely small BER. It is then obvious that for the successful decoding, the outer decoder must satisfy the following condition

$$\sqrt{\frac{2}{M_D^{(l)}}} \leq \sigma_{th}^{(0)}. \tag{3.8}$$

Using the Q-function, i.e., the upper tail function of the standard Gaussian distribution, (3.8) can be rewritten as

$$\mathrm{Q}\left(\sqrt{\frac{M_D^{(l)}}{2}}\right) \leq \mathrm{Q}\left(\frac{1}{\sigma_{th}^{(o)}}\right). \tag{3.9}$$

Knowing that $p_D^{(l)} = \mathcal{N}\left(M_D^{(l)}, 2M_D^{(l)}\right)$ is the PDF of the decision variable at the lth itera-

tion of the inner decoder and under the assumption of all-one BPSK codeword transmission, the decoding error probability for the inner decoder can be calculated as

$$E_{in}^{(l)} = \int_{-\infty}^{0} p_D^{(l)} dD = 1 - \mathrm{Q}\left(\frac{0 - M_D^{(l)}}{\sqrt{2M_D^{(l)}}}\right) = \mathrm{Q}\left(\sqrt{\frac{M_D^{(l)}}{2}}\right). \tag{3.10}$$

From (3.9) and (3.10), we get

$$E_{in}^{(l)} \leq \mathrm{Q}\left(\frac{1}{\sigma_{th}^{(o)}}\right). \tag{3.11}$$

□

In summary, (3.11) is the necessary condition that a SCLDGM code must satisfy in order to be successfully decoded in a BIAWGN channel using the SP decoding. We henceforth call $\mathrm{Q}\left(1/\sigma_{th}^{(o)}\right)$ the critical BER. It is worth pointing out that in a BIAWGN channel under BPSK modulation, $\mathrm{Q}\left(1/\sigma_{th}^{(o)}\right)$ is actually the raw input BER (i.e., the BER with direct threshold detection) of the outer code calculated at $\sigma_{th}^{(o)}$. Let this BER be represented by $P_{b_{th}}$. Then, (3.11) can be rewritten as $E_{in}^{(l)} \leq P_{b_{th}}$, which means that the raw input BER of the outer code at $\sigma_{th}^{(o)}$ is the critical BER. Clearly, handing the decoding to the outer decoder before this critical BER is achieved by the inner decoder is just the waste of decoding.

This theorem asserts that the decoding threshold of the overall SCLDGM code will be the minimum E_b/N_o (maximum σ) at which the inner decoder of the concatenated scheme achieves the critical BER that is determined by the outer code.

It is very important to note here that Forney discussed in [24] that the only function of the inner decoder is to bring the probability of decoding error to the range of $10^{-2} \sim 10^{-4}$, which once achieved, the outer decoder further drives the overall probability of decoding error down to an extremely small desired value. More specifically, that range for BIAWGN channel is predicted to be $10^{-2} \sim 10^{-3}$ [24]. In this section, utilizing the two-step DDE, we proved that $\mathrm{Q}\left(\frac{1}{\sigma_{th}^{(o)}}\right)$ is the exact value in that range that the inner decoder must at least

produce for the successful decoding of the SCLDGM codes.

3.4.1 Finding the Critical BER

Figure 3.6 presents the asymptotic performance obtained using DDE for three regular (d_v, d_c)-LDGM codes of code-rate 50/51. Their decoding thresholds are presented in Table 3.2. Therefore, if these high rate LDGM codes are to be used as the outer codes in a SCLDGM code, the inner decoder must at least produce in each case the respective critical BER as presented in Table 3.2 in order to secure an overall successful decoding. Figure 3.6 also shows that the $(3, 150)$ code is inferior to the $(4, 200)$ code in both error floor and decoding threshold, while the $(4, 200)$ code has better decoding threshold with a slightly higher error floor than the $(5, 250)$ code. We pick the $(4, 200)$ code as the outer code to further study the performance of SCLDGM codes.

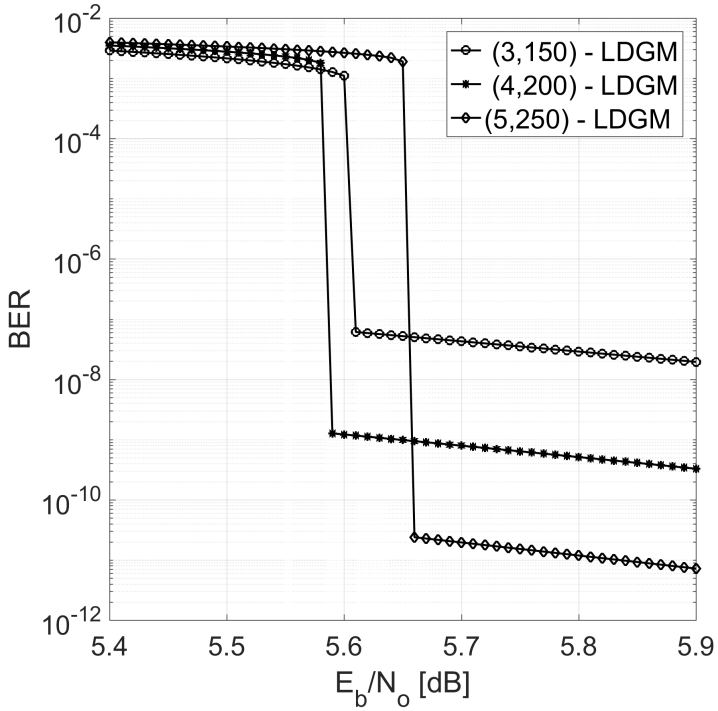

Figure 3.6. Asymptotic performance of (d_v, d_c)- LDGM codes, code-rate = 50/51.

Table 3.2. DDE threshold and critical BER.

(d_v, d_c)	$(E_b/N_o)_{th}$	$\sigma_{th}^{(o)}$	$\mathrm{Q}\left(1/\sigma_{th}^{(o)}\right)$
$(3, 150)$	5.61	0.374	3.778×10^{-3}
$(4, 200)$	5.59	0.375	3.848×10^{-3}
$(5, 250)$	5.66	0.372	3.608×10^{-3}

3.4.2 Verification of the Necessary Condition

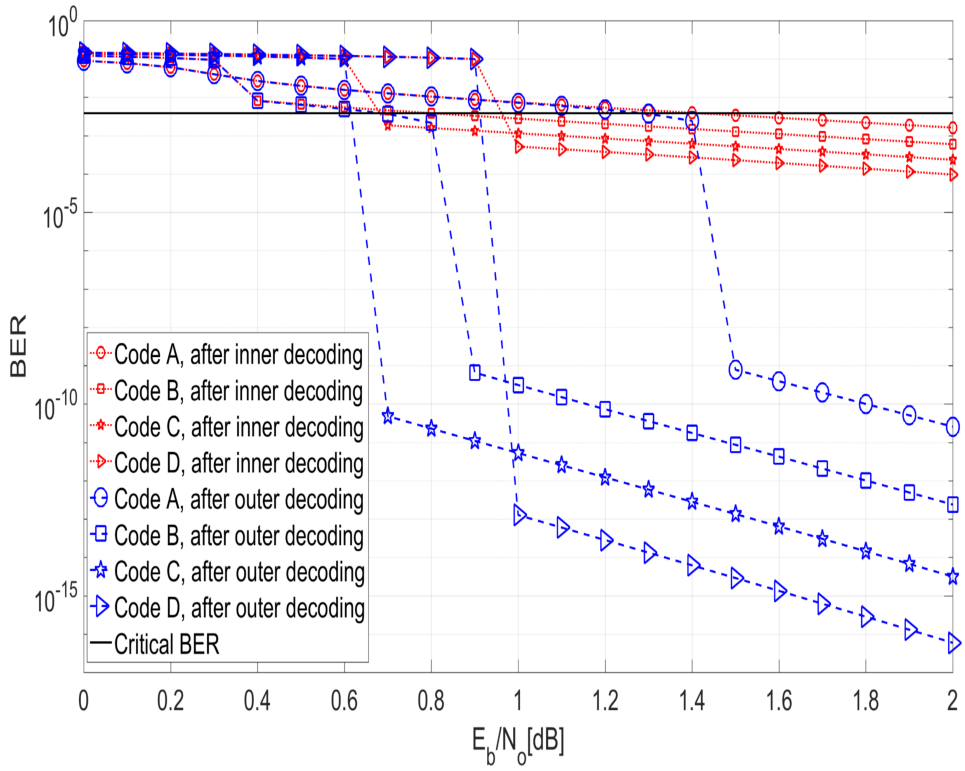

Figure 3.7. Asymptotic performance of various SCLDGM codes.

In Figure 3.7, we provide the asymptotic curves showing the decoding error probability produced by both the inner and outer decoders of four different SCLDGM codes namely code A, B, C, and D. The curves clearly depict that once the inner decoder produces a BER below the critical BER, the outer decoder in each case drastically reduces the BER to the level enough to declare successful decoding. For example, for code A, the BERs computed after inner decoding from 0 to 1.4 dB are above the critical BER value of 3.848×10^{-3}. The BERs further computed after outer decoding in that range mostly remain the same

and only show some sign of improvement near 1.4 dB. However, at 1.5 dB and beyond, the BERs computed after inner decoding start falling below the critical BER, which after outer decoding immediately fall below 10^{-9} level. We observe such facts holding true in the case of each code, which validate the result presented in the theorem.

3.4.3 Convergence of the Inner Decoder

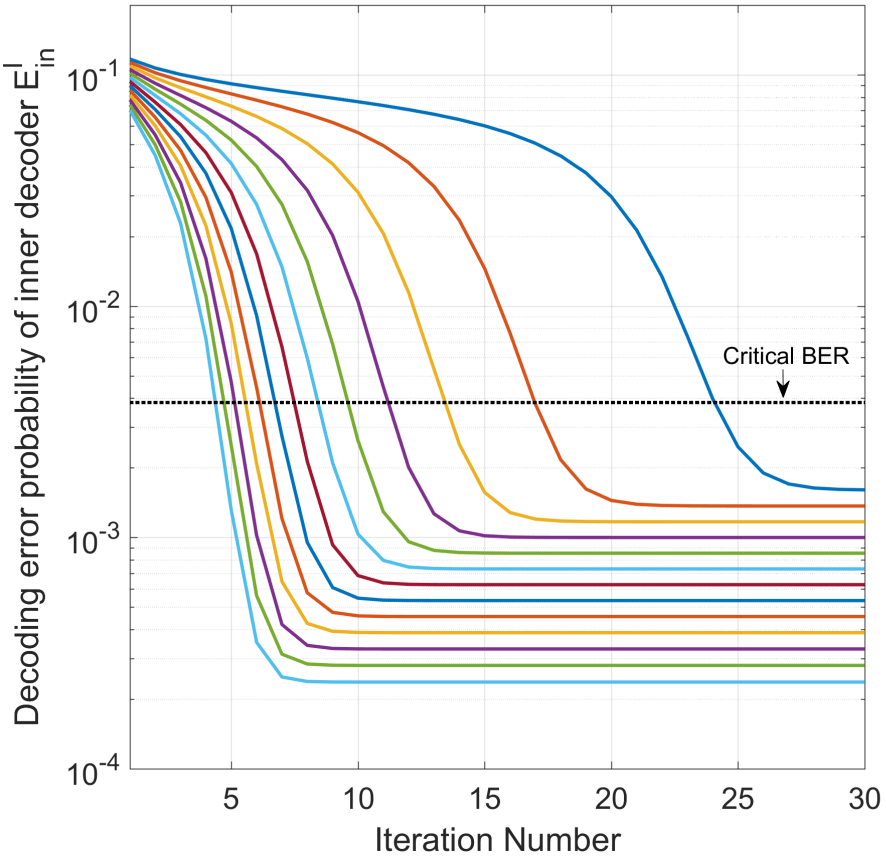

Figure 3.8. Decoding error probability of the inner decoder $\left(E_{in}^{(l)}\right)$ vs. iteration number for a SCLDGM code (Code C) at different E_b/N_o values. E_b/N_o used are 0.8 to 2.0 dB with a gap of 0.1 dB ordered from right to left in the graph.

The bi-partite graph used for the decoding of the outer LDGM code is very small. In addition, it is known that once the critical BER is achieved by the inner decoder, the number of iterations required by the outer decoder to provide successful decoding is very small. Due

to these reasons, the decoding complexity added by the outer decoder is not much. On the other hand, due to the low code-rate of inner code, its bi-partite graph used for the decoding is large. Therefore, the most of the decoding complexities of the SCLDGM codes are associated with the number of iterations required by the inner LDGM codes to achieve the critical BER. Practically, the faster the inner decoding converges below the critical BER, the smaller is the decoding complexity. The asymptotic curves plotted in Figure 3.8 clearly depict that the inner decoder of the SCLDGM codes converges very fast, i.e., the number of iterations required by the inner decoder to provide BER well below the critical BER is small. Hence, this faster convergence behaviour makes the SCLDGM codes practically suitable codes capable of providing lower latency.

3.5 Error-Floor Analysis of LDGM and SCLDGM codes

Single LDGM codes are asymptotically bad and suffer from huge error floors. The SCLDGM codes drastically improve the decoding performance and have decoding performance close to the Shannon limit. However, they also exhibit error-floor behaviour, but at very lower BER level as shown in Figure 3.7. In this section, we provide the reasons behind such error floors and give the lower bound formulas for both the single LDGM and SCLDGM codes. For the error-floor analysis purpose, we have considered both the inner and outer LDGM codes as regular codes with VN degree d_v and $d_v^{(o)}$ respectively.

3.5.1 Lower Bounds for single LDGM Codes

The message u passing through a randomly chosen edge from a degree d_c CN to a VN is calculated using the product rule as given by (3.1), which can be rewritten as

$$u = 2\tanh^{-1}\prod_{j=0}^{d_c-1}\tanh(v_j/2), \tag{3.12}$$

where v_o is the channel estimate of the CN and $v_j, j = 1, 2, \cdots, d_c - 1$ are the incoming LLRs from the neighboring VNs.

Under min-sum decoding, (3.12) is written as

$$u \approx \left(\prod_{j=0}^{d_c-1} \text{sign}(v_j) \right) \cdot \min |v_j|. \tag{3.13}$$

From (3.13), we can write $|u| \approx \min |v_j|$. It is important to note that during the iterative decoding of LDGM codes, the messages $v_j,\ j = 1, \cdots, d_c - 1$ can continuously evolve, however, the message v_0 remains the same throughout the decoding process. Due to this v_0, no matter how large the incoming LLRs are, the following condition holds true throughout the decoding process

$$|u| \leq |v_o| + \epsilon, \tag{3.14}$$

where ϵ is a small positive real number. We neglect ϵ for the rest of the analysis.

It is known that $p_{v_0} \sim \mathcal{N}(2/\sigma^2, 4/\sigma^2)$. Under Gaussian approximation of DE, the message u is also assumed to be Gaussian distributed and consistent throughout the decoding iterations, i.e., $p_u^{(l)} \sim \mathcal{N}\left(M_u^{(l)}, 2M_u^{(l)}\right)$, where $M_u^{(l)}$ is the mean associated with u at the lth iteration. In terms of mean value, (3.14) can be written as

$$M_{|u|}^{(l)} \leq M_{|v_o|} \Rightarrow M_u^{(l)} \leq M_{v_o} \tag{3.15}$$

We can calculate the decoding error probability of u as $e_u^{(l)} = Q\left(\sqrt{M_u^{(l)}/2}\right)$. Using (3.15), we obtain the following inequality

$$e_u^{(l)} \geq Q\left(\sqrt{\frac{M_{v_0}}{2}}\right) \tag{3.16}$$

It is now obvious that even if $l \to \infty$, the magnitude of the message that a CN passes is always bounded by its initial channel estimate and hence its decoding error probability.

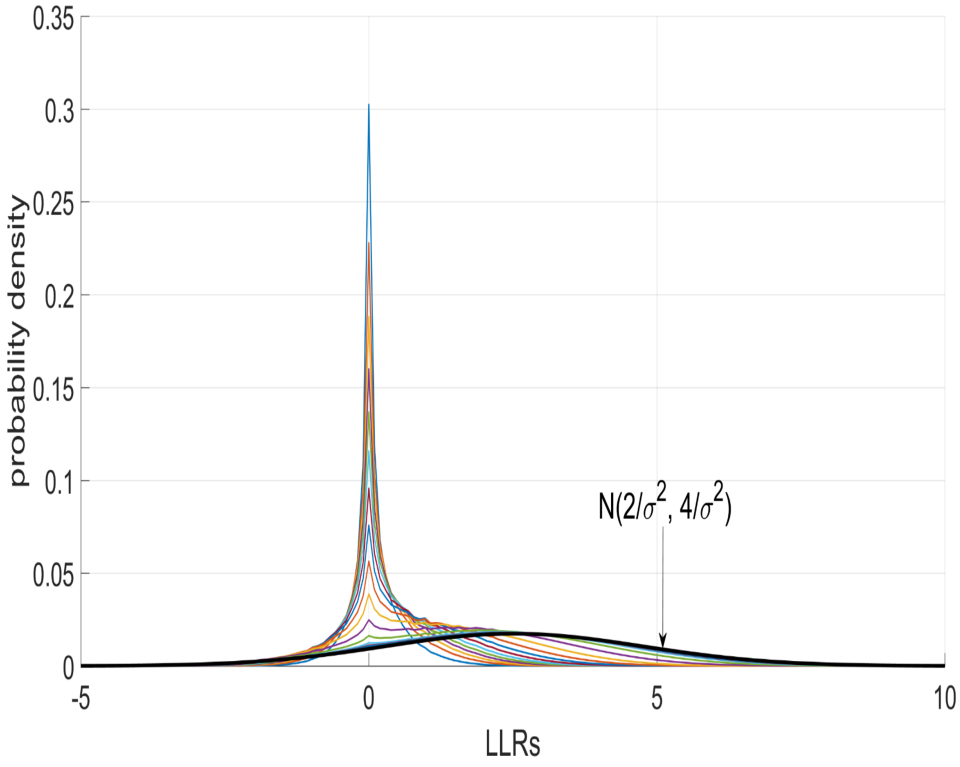

Figure 3.9. Evolution of $p_u^{(l)}$ towards right as l increases for a $(7,7)$-LDGM code, $E_b/N_o = 1$dB.

Knowing from (3.15) that the best possible $M_u^{(l)}$ during the DE is M_{v_0}, it is obvious that the $p_u^{(l)}$ can never evolve beyond $\mathcal{N}\left(2/\sigma^2, 4/\sigma^2\right)$. This fact is clearly depicted in Figure 3.9, where $p_u^{(l)}$ for 50 iterations are plotted along with p_{v_0}. We see that $p_u^{(l)}$ gradually evolves towards right as iteration increases, however, never evolves beyond p_{v_0}.

We know that the decision of each VN is made using the decision rule $D = \sum_{i=0}^{d_v} u_i = u_0 + \sum_{i=1}^{d_v} u_i$. From DE, we know that $u_1, u_2, \cdots, u_{d_v-1}$ are i.i.d Gaussian random variables and $p_{u_i} = p_u \ \forall i$. Hence, the mean associated with the D at the lth iteration is $M_D^{(l)} = M_{u_0} + d_v M_u^{(l)}$. Since, $M_{u_0} = M_{v_0}$ and $M_u^{(l)} \leq M_{v_0}$, we can write the following

$$M_D^{(l)} \leq (d_v + 1)\, M_{u_0}. \tag{3.17}$$

Since $p_D^{(l)} \sim \mathcal{N}\left(M_D^{(l)}, 2M_D^{(l)}\right)$, it is clear from (3.17) that $p_D^{(l)}$ can never evolve beyond $\mathcal{N}\left(2(d_v+1)/\sigma^2, 4(d_v+1)/\sigma^2\right)$ which is clearly depicted in Figure 3.10, where $p_D^{(l)}$ for 50

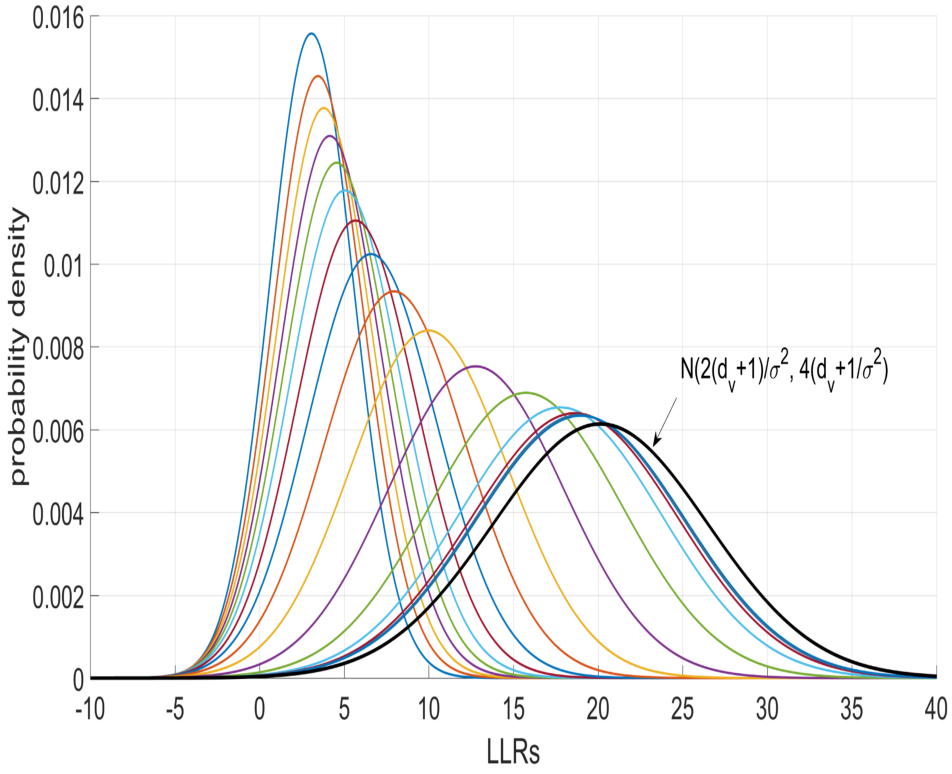

Figure 3.10. Evolution of $p_D^{(l)}$ towards right as l increases for a $(7,7)$-LDGM code, E_b/N_o = 1dB.

iterations are plotted. As can be seen, initially the $p_D^{(l)}$ gradually evolves towards right as iteration increases, however, is bounded by $\mathcal{N}\left(2(d_v+1)/\sigma^2, 4(d_v+1)/\sigma^2\right)$.

The decoding error probability of D calculated at the lth iteration is known to be $e_D^{(l)} = Q\left(\sqrt{M_D^{(l)}/2}\right)$. Using (3.17), we get the following inequality

$$\begin{aligned} e_D^{(l)} &\geq Q\left(\sqrt{\frac{(d_v+1)M_{u_0}}{2}}\right) \\ &\geq Q\left(\sqrt{\frac{(d_v+1)}{\sigma^2}}\right). \end{aligned} \tag{3.18}$$

Hence, the decoding error probability of D, i.e., the BER of the LDGM codes is lower bounded by $Q\left(\sqrt{(d_v+1)/\sigma^2}\right)$. In Figure 3.11, the exact asymptotic performance of various regular half-rate LDGM codes obtained using the DDE is compared with their respective

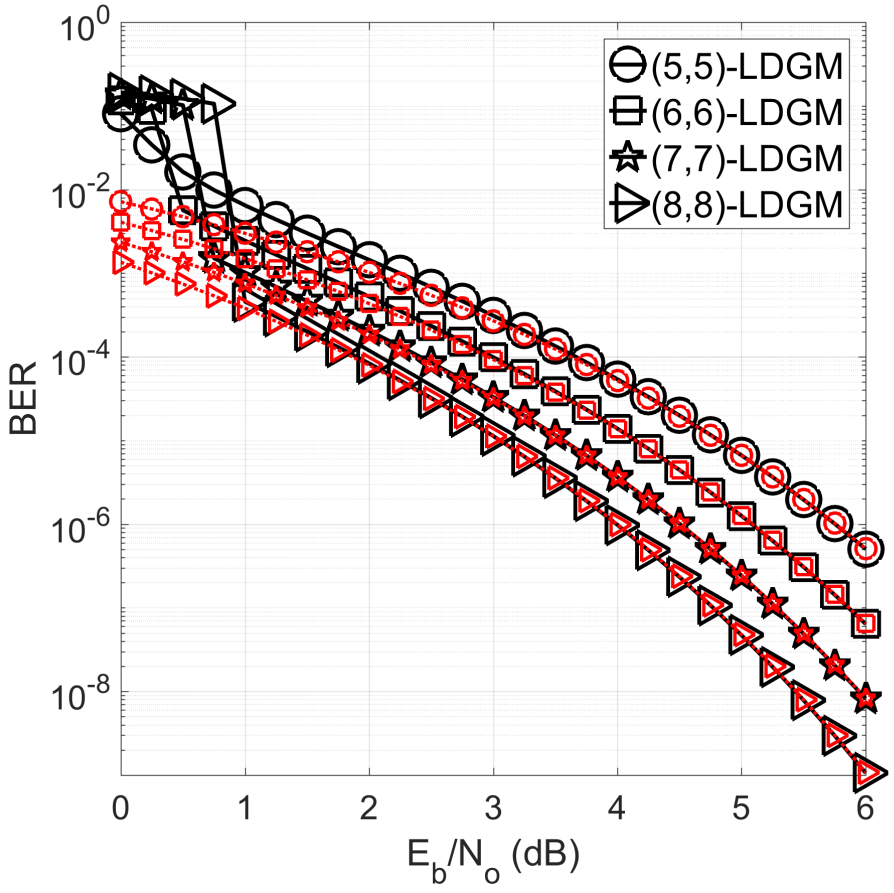

Figure 3.11. Asymptotic performance (solid lines) and lower bounds (dotted lines) of (d_v, d_c)-regular half-rate LDGM codes.

lower-bounds calculated using $Q\left(\sqrt{(d_v+1)/\sigma^2}\right)$. We observe that at higher E_b/N_o values, the exact BER of LDGM codes is closely approximated by the lower bound formula, i.e., $e_D \approxeq Q\left(\sqrt{(d_v+1)/\sigma^2}\right)$ while they significantly differ at lower E_b/N_o values.

3.5.2 Lower Bounds for SCLDGM Codes

Since the outer code of a SCLDGM code is also a LDGM code, the message that the CNs of the outer decoder pass are also bounded by their respective initial estimates, and hence also suffer from error floors as we already observed in Figure 3.7.

It is known from the DE analysis of the SCLDGM codes under two-step decoding that the PDF of the input LLRs to the outer decoder is Gaussian with variance twice the mean.

Let σ_{sup} be the noise parameter associated with the super-channel. Hence, the input LLRs to the outer decoder has the PDF of $\mathcal{N}\left(2/\sigma_{sup}^2, 4/\sigma_{sup}^2\right)$. Hence, based on the analysis from Section 3.5.1, we can write the following

$$E^{(l)} \geq Q\left(\sqrt{\frac{(d_v^{(o)}+1)}{\sigma_{sup}^2}}\right). \tag{3.19}$$

For the inner LDGM code, we knew that $M_D^{(l)} \leq (d_v+1)\, M_{u_0}$. In fact, this $M_D^{(l)}$ is the mean of the input LLRs to the outer decoder through the super-channel. Since $M_D^{(l)} = 2/\sigma_{sup}^2$ and $M_{u_0} = 2/\sigma^2$, we can write the following

$$\begin{aligned} \frac{2}{\sigma_{sup}^2} &\leq (d_v+1)\frac{2}{\sigma^2} \\ \sigma_{sup}^2 &\geq \frac{\sigma^2}{(d_v+1)} \end{aligned} \tag{3.20}$$

Combining (3.19) and (3.20), we get

$$E^{(l)} \geq Q\left(\sqrt{\frac{(d_v^{(o)}+1)(d_v+1)}{\sigma^2}}\right). \tag{3.21}$$

Hence, the decoding error probability of a SCLDGM code under two-step SP decoding is always lower bounded by $Q\left(\sqrt{\left((d_v^{(o)}+1)(d_v+1)\right)/\sigma^2}\right)$. The exact asymptotic performance of various SCLDGM codes obtained using the two-step DDE are compared with their respective lower bounds in Figure 3.11. We observe that at higher E_b/N_o values, the exact asymptotic performance of the SCLDGM codes is well approximated by the lower bound formula while they significantly differ at lower E_b/N_o values. More specifically, we see that at any E_b/N_o value above the decoding threshold, we can just use the lower bound formula to approximately calculate the BER of the code.

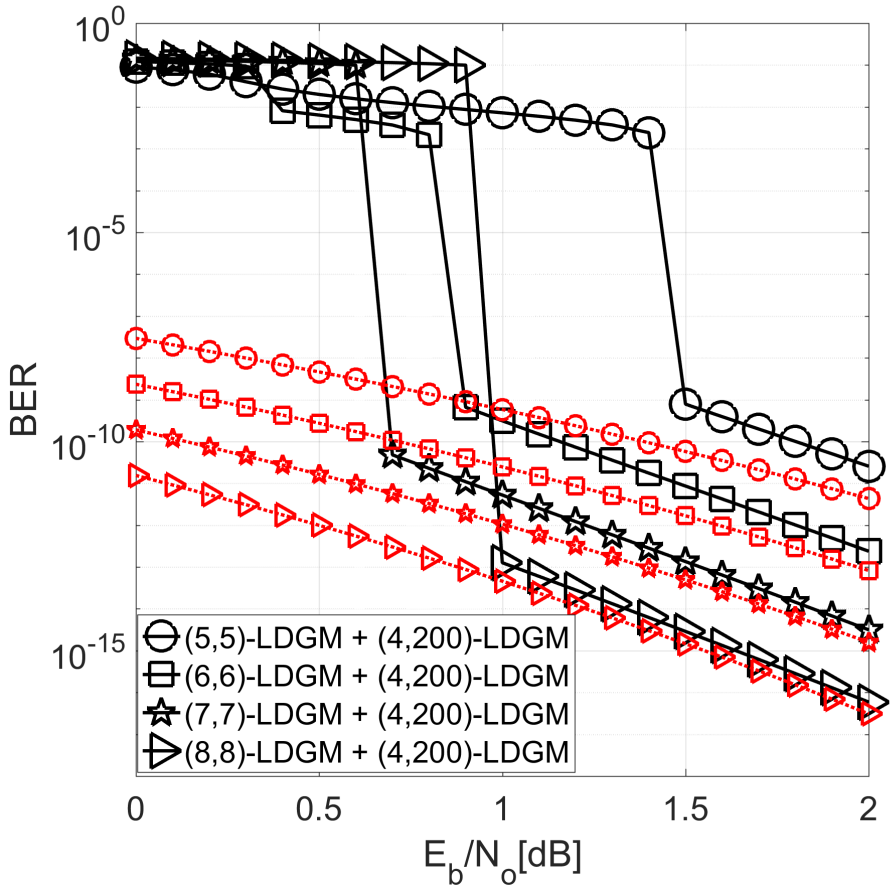

Figure 3.12. Asymptotic performance (solid lines) and lower bounds (dotted lines) of various SCLDGM codes.

3.6 Error Floor Eradication using High-Rate LDPC Code as an Outer Code

Unlike in LDGM codes, the CNs of LDPC codes are always zero in GF(2), and the message v_0 is absent in the product rule. Hence, for LDPC codes, the message that its CN passes is not bounded by the channel estimate which rather continuously evolves as the decoding proceeds. Due to this fact, the error floors are absent in the asymptotic curves of the 50/51-rate regular LDPC code obtained using the DDE method and is also depicted in Figure 3.13. The decoding threshold of the 50/51-rate LDPC code and the corresponding critical BER as presented in Table 3.3 are almost the same as those of 50/51-rate LDGM

codes presented in Table 3.2. Hence, the overall performance of serial concatenation of inner LDGM codes and outer LDPC codes should be as good as the SCLDGM codes and due to the use of outer LDPC codes, the asymptotic curves of such concatenated codes should be free from the error-floors. As expected, we observed no error-floors for such concatenated schemes. Their asymptotic performance is depicted in Figure 3.14. The outer LDPC code used is of code-rate 50/51 with $d_v = 4$. Hence, the use of high-rate LDPC code as an outer code instead of the high-rate LDGM code help completely eliminate the error floors without sacrificing the decoding performance.

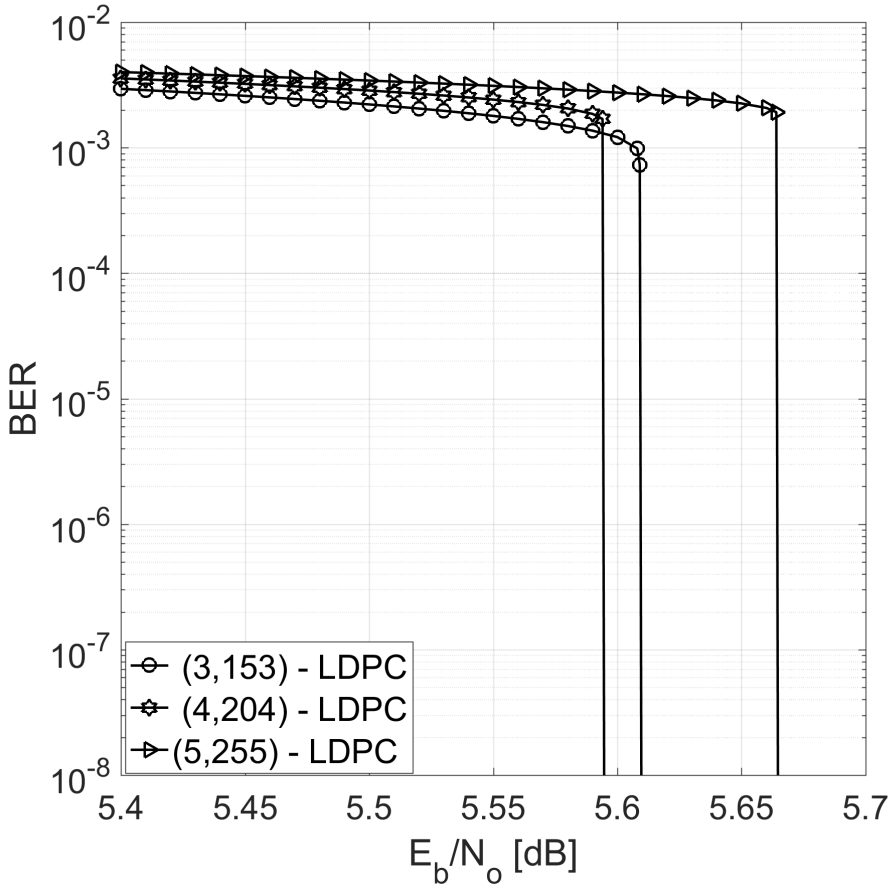

Figure 3.13. Asymptotic performance of (d_v, d_c)-regular LDPC codes, code-rate = 50/51.

Table 3.3. Decoding threshold and critical BER for $50/51$ rate regular (d_v, d_c) LDPC code.

(d_v, d_c)	$(E_b/N_o)_{th}$	$\sigma_{th}^{(o)}$	$\mathrm{Q}\left(1/\sigma_{th}^{(o)}\right)$
$(3, 153)$	5.610	0.374	3.778×10^{-3}
$(4, 204)$	5.595	0.375	3.831×10^{-3}
$(5, 255)$	5.665	0.372	3.592×10^{-3}

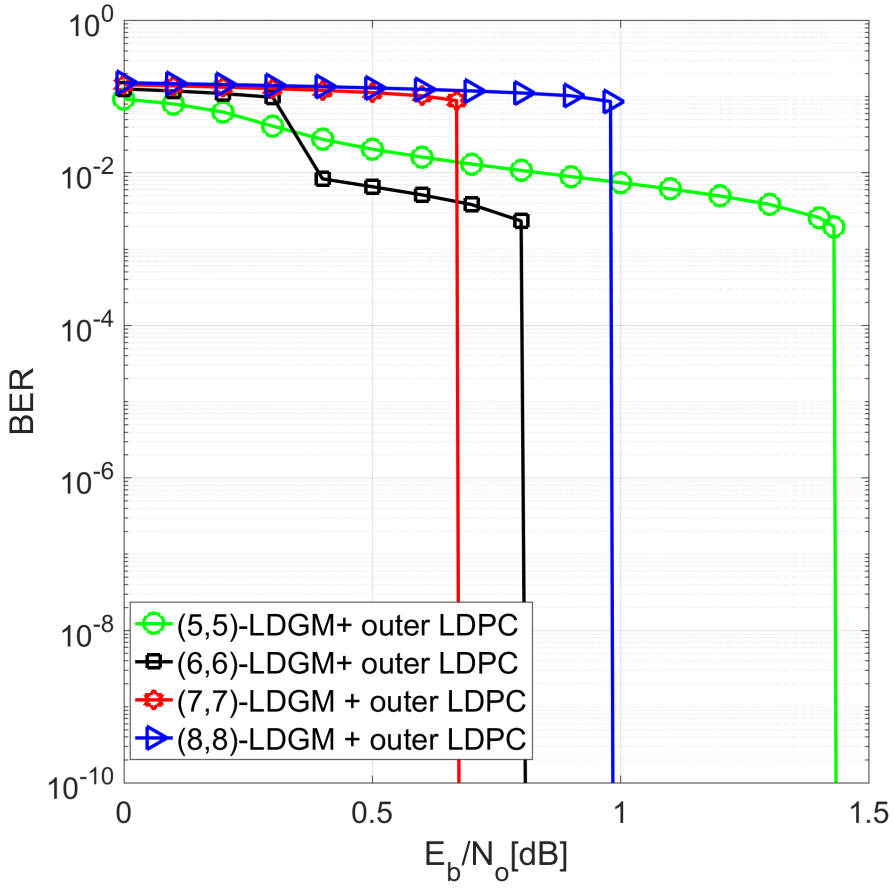

Figure 3.14. Asymptotic performance of inner LDGM concatenated with high-rate outer LDPC codes. Inner code used is rate half (d_v, d_c)-LDGM code and outer code used is $50/51$ rate (4,204)-LDPC code.

3.7 Optimization Design

DDE-based optimization method was used to design capacity approaching irregular LDPC codes in [14, 16]. The detailed two-step DDE implementation of the SCLDGM codes presented in Section 3.3 and the necessary condition for the successful decoding play vital role in the optimization design of the SCLDGM codes. For a given outer code, designing a

good SCLDGM code means finding the degree distribution pair $\Lambda(x)$ and $\Omega(x)$ of the inner code such that the overall decoding error probability $E^{(l)}$ given by (3.7) tends to zero at the lowest possible E_b/N_o. This arduous task, however, with the knowledge of the critical BER becomes finding the pair $\Lambda(x)$ and $\Omega(x)$ of the inner code such that $E_{in}^{(l)} \leq \mathrm{Q}\left(1/\sigma_{th}^{(o)}\right)$ is achieved at the lowest possible E_b/N_o. Hence, the constraints required for the optimization are (a) $\Lambda(1) = 1$, (b) $\Omega(1) = 1$, and (c) $E_{in}^{(l)} \leq \mathrm{Q}\left(1/\sigma_{th}^{(o)}\right)$. The first and second constraints guarantee the valid degree distributions while the third constraint is the necessary condition for the successful decoding.

For the purpose of optimization, we have considered 50/51 rate $(4, 200)$-LDGM code as the outer code. The code-rate 50/51 is chosen since it is known to be one of the best choices for the outer codes in such concatenated schemes [26]. $(4, 200)$-LDGM code is used since it is best regular LDGM code when the code-rate is 50/51 as shown in Table 3.2. Hence, for a SCLDGM code with this outer code, the critical BER to be achieved by the inner decoder as known from Table 3.2 is 3.848×10^{-3}. The optimization process then becomes finding the optimized $\Lambda(x)$ and $\Omega(x)$ for the inner code such that 3.848×10^{-3} is achieved at the lowest possible E_b/N_o. Actually, we focus on obtaining $\Lambda(x)$ while its pair $\Omega(x)$ is calculated as $\Omega(x) = \Omega_i x^i + \Omega_{i+1} x^{i+1}$ for some $i \geq 2$, where the coefficients and the exponents can be easily computed from the knowledge of $\Lambda(x)$ and code-rates [14]. We do the minimization of $E_{in}^{(l)}$ starting at slightly higher E_b/N_o and search for degree distributions satisfying the constraints. Once successful, we lower E_b/N_o and repeat the minimization subject to our constraints. As an example, let us consider that the code-rate of the inner LDGM code we want to optimize be one-half. Since Code C $((7, 7)$-LDGM + $(4, 200)$-LDGM) is a SCLDGM code with half-rate inner LDGM code and have the DDE threshold of 0.68 dB, the initial E_b/N_o chosen for our optimization process is close to 0.68 dB, which after each success, is lowered by 0.01 dB. Due to the monotonicity of the threshold [28], in practice, we speeded up the search process by using the bisection search at a desired level of precision. The lowest

E_b/N_o at which such search is successful is called the decoding threshold of the optimized code. The corresponding optimized degree distributions for the half-rate inner code we obtained using this process are given by (3.22). We found that the genetic algorithm based global optimization method called differential evolution [29] that has been successfully used to design optimized irregular LDPC codes in [16] and good erasure codes in [30] are equally effective in designing optimized SCLDGM codes by incorporating our critical-BER based DDE optimization approach.

$$\begin{aligned} \Lambda(x) &= 0.35624x^6 + 0.5963x^7 + 0.047452x^{100}, \\ \Omega(x) &= 0.49187x^{11} + 0.50813x^{12}. \end{aligned} \tag{3.22}$$

The asymptotic performance of our optimized SCLDGM code is depicted in Figure 3.15. The lowest E_b/N_o at which we obtained an inner code that achieved the critical BER during the optimization process was 0.41 dB. Consequently, we see from the asymptotic curves that at 0.41 dB and beyond, the overall BER of the SCLDGM code is drastically dropped, making 0.41 dB the decoding threshold of the optimized SCLDGM code. This code is within 0.26 dB to the Shannon limit, and clearly outperforms the best regular SCLDGM code (i.e, the code C) in Figure 3.7 whose decoding threshold is 0.68 dB as presented in Table 3.1. Although our optimization approach does not consider the extrinsic messages exchanged between the inner and outer decoders at each iteration as in the EXIT-based optimization used in [26], our results presented here are as good as those presented in [26] when 50/51 rate outer code is considered. This further validates the strength of our proposed critical BER-based DDE optimization approach. We also want to emphasize that $n_b = 10$ and 200 iterations were used for the DDE calculation during the optimization process. Considering more iterations as in LDPC design and higher n_b can further improve the decoding threshold.

The simulation result is presented in Figure 3.16. We have used 1000 message blocks each with k=10000 information bits. The maximum number of iterations allowed for the

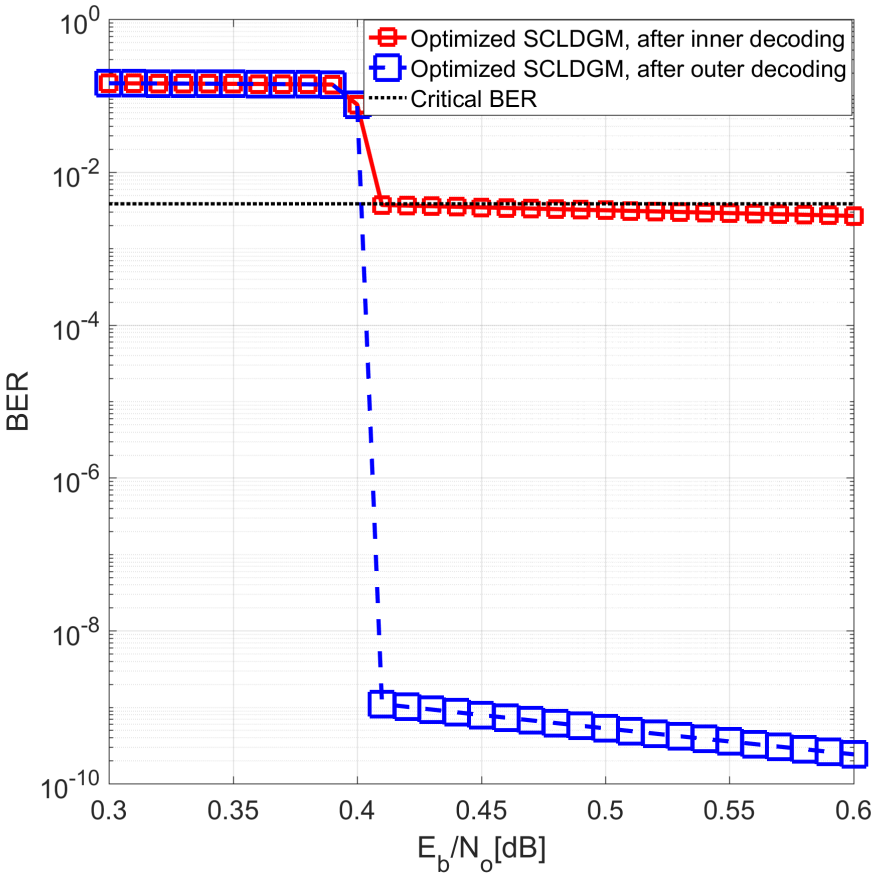

Figure 3.15. Asymptotic performance of our optimized SCLDGM code.

inner and outer decoder is 100. We see that our optimized SCLDGM code outperforms the best SCLDGM code (code C). This also verifies the asymptotic results presented. We would point out that although SCLDGM codes are used here, the concept of the critical BER and the optimization approach can also be exploited in analyzing and designing other serially concatenated error correcting codes.

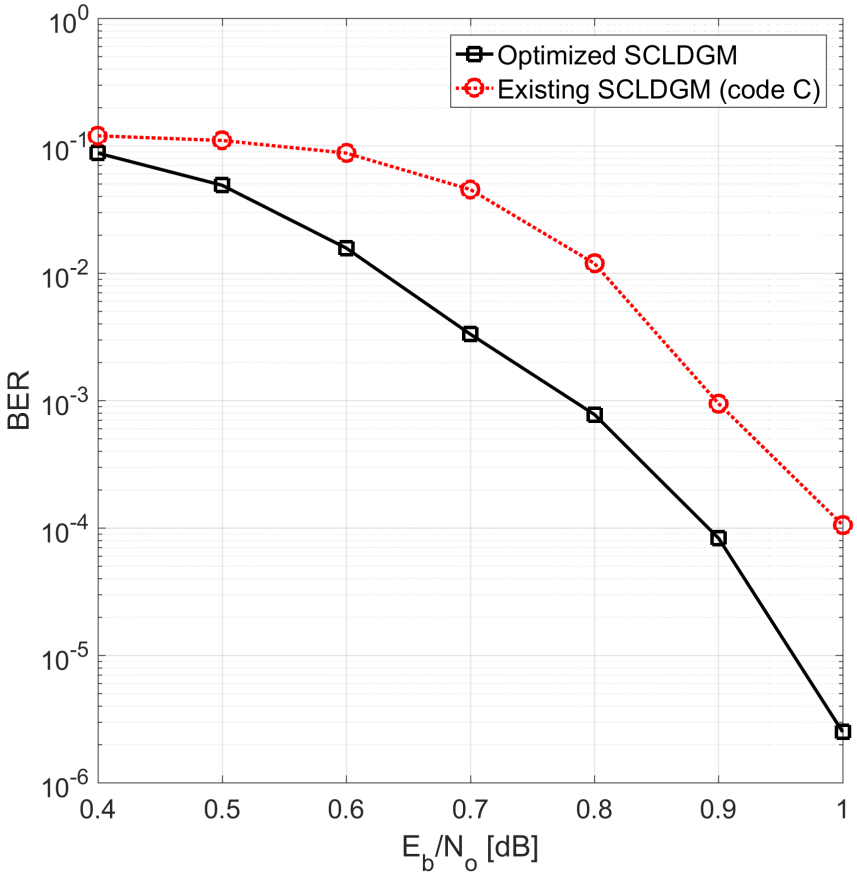

Figure 3.16. BER performance comparison of our optimized SCLDGM vs. existing SCLDGM code (code C), k=10000.

CHAPTER 4

DESIGN OF PHYSICAL LAYER RAPTOR CODES

In this chapter, we provide the detailed asymptotic analysis of LT and Raptor codes. For the Raptor codes, the two-step DDE method is used to provide the exact asymptotic curve. We provide key insights regarding the SP decoding of LT codes in BIAWGN channels and show their resemblance to the decoding process in BEC. Using critical BER-based optimization approach, we design capacity approaching Raptor codes for BIAWGN channels. We also provide the detailed asymptotic analysis and optimization design of systematic version of these codes.

4.1 Encoding and Decoding of LT codes over BIAWGN Channels

4.1.1 Encoding

Let k be the number of bits in a packet. For the transmission over noisy channels, these k bits are encoded to generate n output bits. The random encoding nature of LT codes allows generating any number of output bits as desired or required. Hence, any code-rate k/n can be realized as required. Let $\boldsymbol{s} = [s_1, s_2, \cdots, s_k]$ represent the source/information bits in the packet and $\boldsymbol{o} = [o_1, o_2, \cdots, o_n]$ be output bits to be transmitted.

4.1.2 Decoding

A bi-partite graph as shown in Figure 4.1 is used to conduct SP decoding of LT codes in BIAWGN channels. The VNs $V_1, V_2, \cdots, V_k$ represent the original source bits while the CNs $C_1, C_2, \cdots, C_n$ represent the output bits. Under BPSK modulation, the channel is modelled

as $y_j = x_j + w_j$, where x_j is the modulated version of o_j and w_j is zero mean Gaussian noise. The channel estimate associated with each CN is calculated as $L_{C_j} = 2y_j/\sigma^2$, where σ^2 is the variance of the Gaussian noise. Since only output bits are transmitted, the VNs do not have associated channel estimates. Hence, $L_{V_i} = 0,\ i = 1, 2, \cdots, k$, which is not the case in both LDGM and LDPC codes.

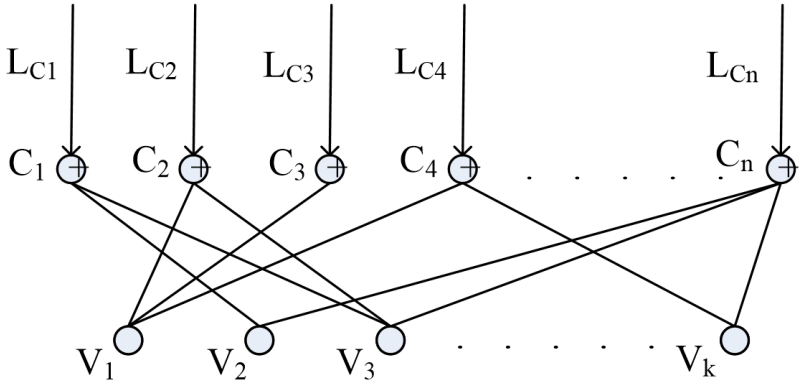

Figure 4.1. Bipartite graph used for LT decoding over BIAWGN Channel.

The product and sum rule used for LT decoding in BIAWGN channel are

$$L_{C_jV_i} = 2\tanh^{-1}\left(\left(\tanh\frac{L_{C_j}}{2}\right)\prod_{i'\in N_{C_j}-\{i\}}\tanh\left(\frac{1}{2}L_{V_{i'}C_j}\right)\right), \tag{4.1}$$

$$L_{V_iC_j} = \sum_{j'\in N_{V_i}-\{j\}} L_{C_{j'}V_i}, \tag{4.2}$$

where N_{C_j} denotes the set of VNs connected to the jth CN and N_{V_i} denotes the set of CNs connected to the ith VN.

The decision rule used is

$$L_D = \sum_{j'\in N_{V_i}-\{j\}} L_{C_{j'}V_i}. \tag{4.3}$$

4.1.3 Full Nodes and Empty Nodes

We know that none of the VNs of LT decoding graph have their channel estimates while all the CNs have their channel estimates. Based on the product rule of SP decoding, it is obvious that for a CN C_j to transmit a non-zero LLR message to a VN V_i, the CN C_j must have received non-zero LLRs from all other edges. This means that at the first iteration, only degree-one CNs are able to pass non-zero LLRs to their neighboring VNs. At the second and subsequent iterations, a degree d_c CN can pass non-zero LLR message through an edge e only if it has received non-zero LLRs from all other $d_c - 1$ edges. From VNs perspective, at the first iteration, only those VNs connected to degree-one CNs get non-zero LLRs while rest of the VNs receive zero LLRs. As the iteration proceeds, more and more VNs start getting non-zero LLRs. We denote the VNs that received non-zero LLRs from at least one of their edges as **full nodes**. Otherwise, we denote them as **empty nodes**. It is obvious that from the BEC's perspective, full nodes are equivalent to recovered source symbols while empty nodes are those source symbols which are yet to be recovered.

Referring to the graph in Figure 4.1, at the first iteration, the CN C_3 passes a non-zero message to the VN V_1 while none of the other CNs can pass the non-zero message to any of the VNs. Therefore, only V_1 turns to full node and rest of the VNs remain empty. Then, at the second iteration, VNs V_3 and V_k turn to full nodes as they are at this point able to get a non-zero message from check nodes C_2 and C_4 respectively. The VN V_2 has to wait until the third iteration to get non-zero messages from C_1 and C_n. Hence, we see that a VN V_i obtains a non-zero LLR message from its neighboring CN C_j only if all the remaining neighbors of C_j have already become full nodes.

In this way, during the decoding process, a fraction of empty nodes turn into full nodes until all the VNs connected in the decoding graph become full nodes.

4.1.4 Incomplete Decoding

From BEC's perspective, full nodes and empty nodes mean recovered and unrecovered source symbols respectively. We know from LT decoding in BEC that when all the source symbols are recovered, successful decoding occurs. However, there exists a scenario in which source symbols can no more be recovered which is popularly known as premature termination and adversely affects the decoding performance. Whether we are considering BEC or BIAWGN channel, the decoding graph associated with the code constructed using the generator (G) matrix have the same structure. Therefore, any structural issues that impair the performance of LT codes in BEC should obviously impair the performance of these codes in noisy channels as well. Regarding LT decoding in BIAWGN channel, the equivalence of premature termination is that at some point during the decoding process, we will not be able to turn any of the remaining empty nodes into full nodes. In such case, even though SP decoding keeps on running, the remaining empty nodes never get non-zero updates. We call this scenario incomplete or premature decoding.

It is known from the SP decoding of LT codes that at the beginning all the VNs are empty nodes. It means that all CNs have their all edges connected to only empty nodes. As the SP decoding proceeds, each CN may have their edges connected to either empty nodes or full nodes or both.

Based on the SP decoding of LT codes as described above, we can also redefine the terms used in BEC's case such as *ripple* and *reduced degree-one encoded symbol* from noisy channel's perspective. Any degree d_c CN which has one edge connected to empty node and other $d_c - 1$ edges connected to full nodes as shown in Figure 4.2 is the noisy equivalence of reduced degree-one encoded symbols. Let us call such CNs as single empty neighbor CNs. Then, a set of empty nodes connected to such single empty neighbor CNs are in fact the noisy equivalence of the ripple. Hence, in the next decoding iteration, all these empty nodes in the ripple turn to full nodes. Therefore, as in BEC, this ripple should never cease to zero as long as there are empty nodes connected to CNs. It implies that if no more CNs become

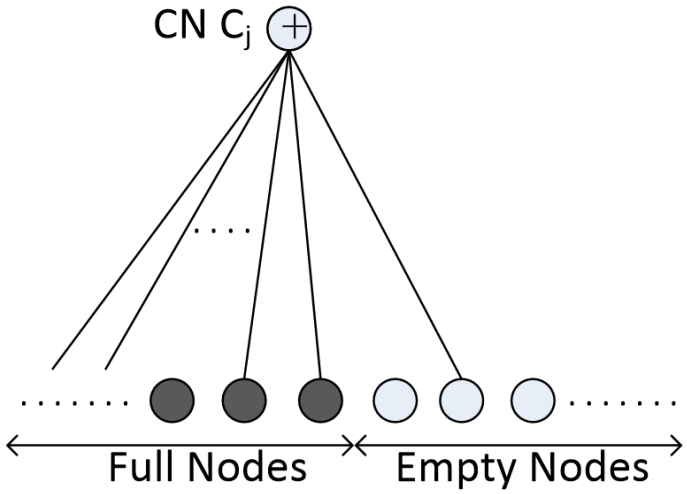

Figure 4.2. A CN with one edge connected to empty node and other edges connected to full nodes.

single empty neighbor CNs, then the ripple size ceases to zero, and no remaining empty nodes can be turned into full nodes. This is the noisy equivalence of premature termination and we call it premature/incomplete decoding.

4.1.5 Number of Full Nodes vs. Iteration Number

In this section, we present a method that can calculate the exact number of full nodes generated at each decoding iteration for any given ODD. We utilize the concept of 'And-Or' tree analysis [31] in our calculation.

Let us represent empty and full nodes as label 0 and label 1 nodes. Hence, the rule of turning empty nodes to full nodes is: *Any VN V_i which is an empty node (label 0) turns into a full node (label 1), if it has at least one neighboring CN C_j such that all neighboring VNs of C_j except V_i are full nodes.*

This rule is exactly the same as the graph substitution recovery rule presented in [31] for erasure channels which states that "A left node v with label 0 is allowed to change its label to a 1 if it has at least one right neighbor w such that all left neighbors of w except v

have label 1". The left and right node used in the rule refers to VNs and CNs.

Therefore, the formulation developed in [31] can be directly applied to our case. It is proved in [31] that the probability that a VN V_i is empty can be calculated iteratively using $Y_l = \delta \cdot \lambda(1-\omega(1-Y_{l-1}))$, where $\lambda(x)$ and $\omega(x)$ represent the degree distributions associated VNs and CNs respectively from the edge perspective, and δ is the probability that a VN V_i is empty at the beginning. Since in our case, all the VNs are empty at the beginning, we have $\delta = 1$. The iterative calculation is started with $Y_0 = \delta$. As l increases, Y_l should decrease to 0. The condition will be true if $\lambda(1 - \omega(1 - z)) < z,\ \forall z \in (0, 1]$ [31].

Under this And-Or tree analysis, we see that Y_l gives the exact fraction of VNs remaining empty by the completion of the lth iteration. Hence, for any given ODD, $(Y_{l-1} - Y_l)$ gives the exact fraction of new full nodes generated at the completion of the lth iteration.

Table 4.1. Percentages of new full nodes generated at each iteration.

Iteration	$\Omega_{\mathrm{T}}(x)$	$D(x)$
1	1.5820	14.6390
2	3.0470	21.6810
3	5.7410	26.9670
4	10.3390	23.0990
5	17.0440	11.1764
6	23.6700	2.3815
7	23.6270	0.0552
8	12.7007	0.0001
9	2.2077	0
10	0.0407	0
11	0.0001	0

$$\begin{aligned}\Omega_{\mathrm{T}}(x) = 0.008x^1 + 0.4936x^2 + 0.1662x^3 + 0.0726x^4 + 0.0826x^5 + \\ 0.0561x^8 + 0.0372x^9 + 0.0556x^{19} + 0.0250x^{65} + 0.0031x^{66}\end{aligned} \tag{4.4}$$

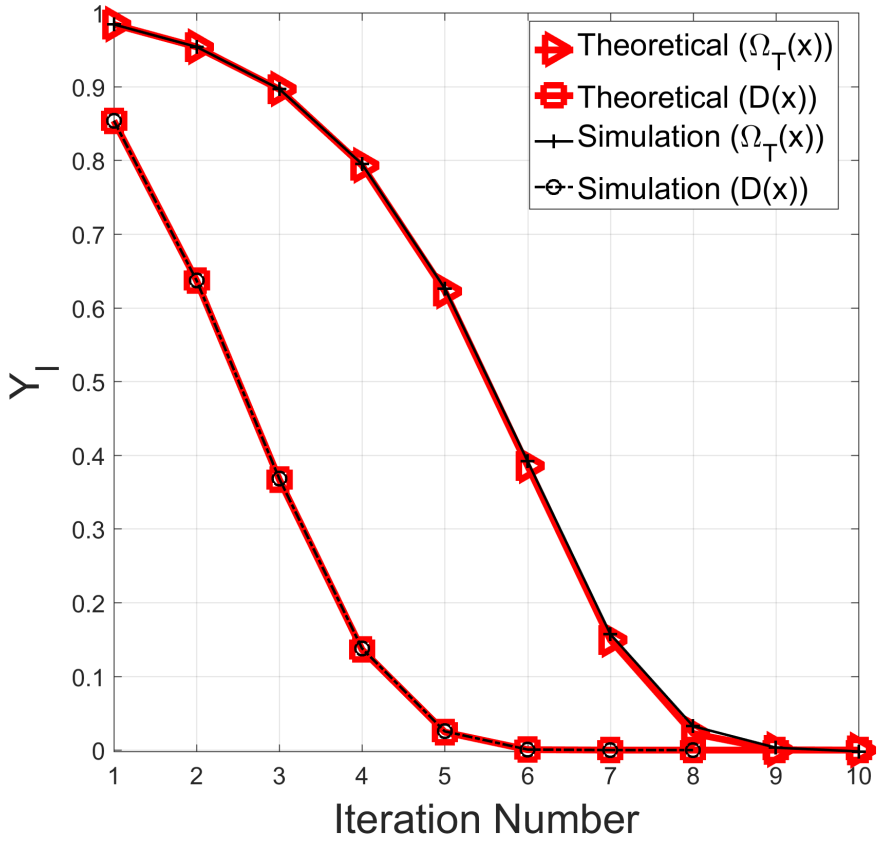

Figure 4.3. Y_l **vs. Iteration Number**

$$D(x) = 0.0791x + 0.4560x^2 + 0.1916x^3 + 0.0564x^4 + 0.0449x^5 + \\ 0.0252x^8 + 0.0376x^9 + 0.0825x^{19} + 0.0165x^{65} + 0.0102x^{66} \tag{4.5}$$

Figure 4.3 shows both the simulation and theoretical results for two different degree distributions. For both distributions, the theoretical curves are obtained using the Y_l formula while the simulation curves are obtained by averaging over simulation results of 100,000 message blocks of $k = 1000$ and code-rate of $1/2$. We can see that the theoretical results perfectly match with the simulation results.

Table 4.1 presents the percentages of the full nodes generated at each iteration for the

two different ODDs $\Omega_{\mathrm{T}}(x)$ and $D(x)$.

4.2 Encoding and Decoding of Raptor Codes over BIAWGN Channels

Raptor codes, being serially concatenated codes, involve two stages of encoding and decoding [4]. At first, source bits $\boldsymbol{s} = [s_1, s_2, \cdots, s_k]$ are encoded using a high-rate outer LDPC code to produce k' intermediate bits $\boldsymbol{m} = [m_1, m_2, \cdots, m_{k'}]$, $k' > k$. These intermediate bits are further encoded using an inner LT code to produce final output bits $\boldsymbol{o} = [o_1, o_2, \cdots, o_n]$, $n > k'$. Hence, a Raptor code of any code-rate k/n can be realized based on the number of output bits generated by the inner LT code.

During the decoding of Raptor codes, as explained in [6], we first complete the SP decoding for the inner LT code followed by the SP decoding of the outer LDPC code. The decoding graph for the inner LT code is constructed by representing intermediate bits as VNs and output bits as CNs. The received channel estimates are $L_{C_j} = 2y_j/\sigma^2$. The product and sum rules as given by 4.1 and 4.2 are used to iteratively perform the SP decoding over a predefined number of iterations. Then, the decision rule as given by 4.3 is used to sum all the incoming LLRs for each VN. These gathered LLRs are used as the channel estimates for the outer decoder. The outer decoding graph is constructed using the (H)-matrix associated with the outer LDPC code, where again VNs represent the intermediate bits while each CN is 0 in GF(2). The SP decoding of LDPC codes as discussed in Chapter 2 is then run over the corresponding decoding graph.

4.3 Asymptotic Analysis of Raptor Codes using DDE

Since both the inner and outer decoders of the Raptor codes use SPA, just like SCLDGM codes, we implement the DDE in two steps. We first apply it to the inner LT code, and then to the outer LDPC code to complete the asymptotic analysis of the Raptor codes. Let $\mathrm{R_I}$

and $\mathrm{R_O}$ be the code-rates of the inner and outer codes respectively making a Raptor code of realized code-rate $\mathrm{R} = \mathrm{R_I R_O}$. We assume that the all-one BPSK codeword is transmitted. The channel condition E_b/N_o, noise parameter σ, and overall code-rate R are related as $E_b/N_o = 10\log_{10}\left(1/\left(2\mathrm{R}\sigma^2\right)\right)$. The channel estimates are known to be $\mathrm{N}\left(2/\sigma^2, 4/\sigma^2\right)$ [14].

4.3.1 DDE for inner code

Let $\Omega(x) = \sum_{i=1}^{k'} \Omega_i x^i$, where $\sum_{i=1}^{k'} \Omega_i = 1$ be the ODD for the inner LT codes, where Ω_i represents the probability of generating a degree i output bit. Regarding the decoding graph of the inner LT code, $\Omega(x)$ is in fact the node degree distribution associated with the CNs. Let $\Lambda(x) = \sum_{i=0}^{n} \Lambda_i x^i$, where $\sum_{i=0}^{n} \Lambda_i = 1$ be the node degree distribution associated with the VNs. Due to random LT encoding, $\Lambda(x)$ is known to be binomial [6] and its coefficients are calculated as

$$\Lambda_i = \binom{n}{i} \left(\frac{a}{k'}\right)^i \left(1 - \frac{a}{k'}\right)^{n-i}, \tag{4.6}$$

where $a = \Omega'(1)$ is the average degree of the LT encoded symbols and $\Omega'(x)$ is the derivative of $\Omega(x)$.

Let $\omega(x) = \sum_{i=1}^{k'} \omega_i x^{i-1}$ and $\lambda(x) = \sum_{i=1}^{n} \lambda_i x^{i-1}$ be the edge degree distributions associated with the CNs and VNs of the inner LT code respectively. ω_i and λ_i are calculated as $\omega_i = i\Omega_i / \sum_{j=1}^{k'} j\Omega_j$, and $\lambda_i = i\Lambda_i / \sum_{j=1}^{n} j\Lambda_j$ respectively.

As seen in (4.6), $\Lambda(x)$ of inner LT code is dependent on block-length $n = k'/R_I$, and it is known that DDE inherently assumes infinite block-length [14]. However, for a given $\Omega(x)$ and code-rate R_I, the curve of $\Lambda(x)$ remains almost the same irrespective of the block-length and has a mean of $\Omega'(1)/R_I$ as can be seen in Figure 4.4. Hence, regarding $\Lambda(x)$ for DDE implementation purpose, we can use a relatively larger block-length and consider only those first d Λ_i components such that $\sum_{i=0}^{d} \Lambda_i \approx 1$.

Let $\bar{v}$ be a quantized LLR message through a randomly chosen edge from a degree d_v VN to a CN and let $\bar{u}$ be that from a degree d_c CN to a VN. Under SP decoding, $\bar{v}$ is calculated

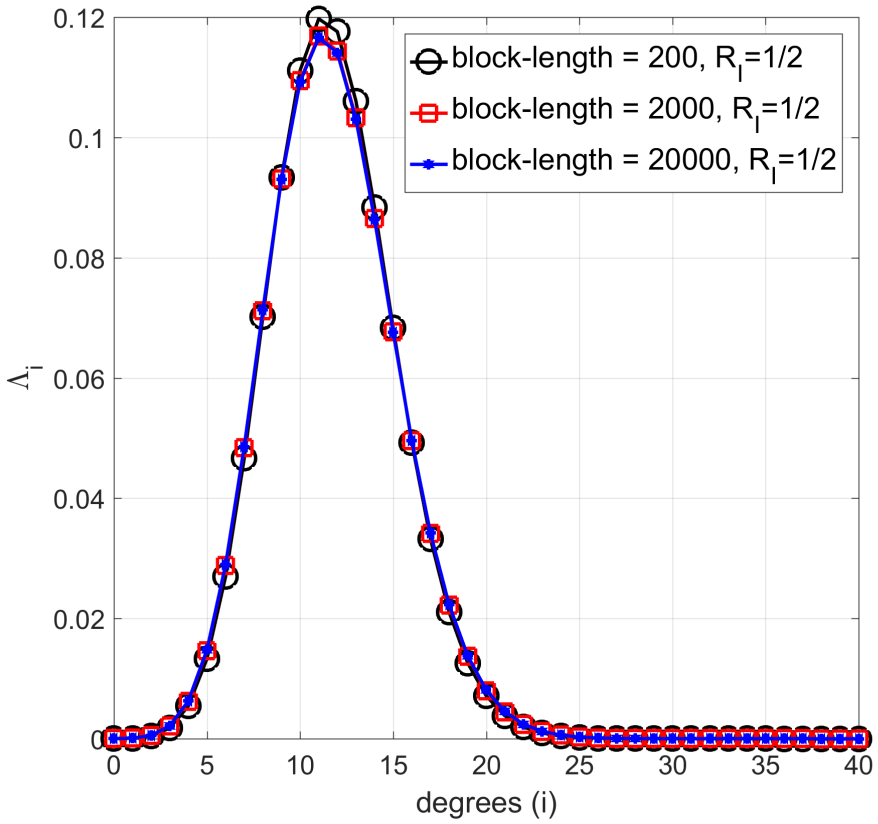

Figure 4.4. $\Lambda(x)$ **of LT codes**

as $\bar{v} = \sum_{i=0}^{d_v-1} \bar{u}_i$, where $\bar{u}_i,\ i = 1, \cdots, d_v - 1$ are the incoming quantized LLRs from the neighboring CNs of the VN except the CN that gets the message $\bar{v}$, and $\bar{u}_0$ is the observed LLR of the VN. As in [13], the PMF of $\bar{v}$ is calculated as $p_{\bar{v}} = p_{\bar{u}_0} \bigotimes \left(\bigotimes^{d_v-1} p_{\bar{u}} \right)$, where $\bigotimes$ is discrete convolution, the superscript represents the number of times convolution is operated, and $p_{\bar{u}}$ is the PMF of $\bar{u}$. Similarly, for the discretized decision rule $\bar{D} = \sum_{i=0}^{d_v} \bar{u}_i$, the associated PMF is $p_{\bar{D}} = p_{\bar{u}_0} \bigotimes \left(\bigotimes^{d_v} p_{\bar{u}} \right)$.

Since $p_{\bar{v}} = p_{\bar{v}_i}$ for any $1 \leq i \leq d_c$, the PMF $p_{\bar{u}}$ is calculated as $p_{\bar{u}} = R(p_{\bar{v}_0}, R(p_{\bar{v}}, R(p_{\bar{v}}, \cdots, R(p_{\bar{v}}, p_{\bar{v}})))) = R(p_{\bar{v}_0}, R^{d_c-1} p_{\bar{v}})$, where the PMF of $R(a, b)$ denoted as p_c is calculated as $p_c[k] = \sum_{(i,j):k\Delta = R(i\Delta, j\Delta)} p_a[i] p_b[j]$, where Δ is the quantization step.

The PMFs $p_{\bar{u}_0}$ and $p_{\bar{v}_0}$ need to be known initially. For inner LT codes, observed LLR for each VN is zero. Therefore, we can write, $p_{\bar{u}_0} = \sum_{j=-\infty}^{+\infty} \delta[j]$, where $\delta[j]$ represents the dirac

delta function. It is important to note that this $p_{\bar{u}_0}$ in the case of LDGM and LDPC codes is known to be Gaussian with mean $2/\sigma^2$ and variance $4/\sigma^2$ [14]. Furthermore, unlike LPDC codes, all the CNs of LT decoding graph actually have their observed LLRs calculated as $L(c_j) = 2y_j/\sigma^2, j = 1, 2, \cdots, n$. Hence, for each CN, the PMF of the observed LLR can be calculated as $p_{\bar{v}_0}$ = PMF of $L(c_j) = \mathcal{N}(2/\sigma^2, 4/\sigma^2)$.

By defining $f_\lambda(p) = \lambda_1 p_{\bar{u}_0} + \sum_{i=2}^{d_v} \lambda_i \left(p_{\bar{u}_0} \bigotimes \left(\bigotimes^{i-1} p \right) \right)$ and $f_\omega(p) = \omega_1 p_{\bar{v}_0} + \sum_{j=2}^{d_c} \omega_j \left(R(p_{\bar{v}_0}, R^{j-1} p) \right)$, the evolving PMF of $\bar{u}$ and $\bar{v}$ at each iteration can then be computed as $p_{\bar{v}}^{(l)} = f_\lambda \left(f_\omega \left(p_{\bar{v}}^{(l-1)} \right) \right)$ and $p_{\bar{u}}^{(l)} = f_\omega \left(f_\lambda \left(p_{\bar{u}}^{(l-1)} \right) \right)$ respectively, where the initial PMFs $p_{\bar{v}}^{(0)}$ and $p_{\bar{u}}^{(0)}$ have all mass at zero.

Probability of Decoding Error for inner code

For the inner code with all VNs having same degree, the associated PMF of the decision rule is $p_{\bar{D}} = p_{\bar{u}_0} \bigotimes \left(\bigotimes^{d_v} p_{\bar{u}} \right)$. However, for irregular inner code, we can calculate $p_{\bar{D}}$ at the lth iteration as

$$p_{\bar{D}}^{(l)} = \sum_{i=0}^{d_v} \Lambda_i \left(p_{\bar{u}_0} \bigotimes \left(\bigotimes^{i} p_{\bar{u}}^{(l)} \right) \right). \tag{4.7}$$

It is important to note that during LT encoding process, some of the intermediate symbols are left unchosen with Λ_0 probability. These unchosen symbols impact the decoding performance. Hence, Λ_0 should be included in (4.7). Let $[-L_a, L_a]$ be the range of LLRs used. Thus, quantization step is $\Delta = 2L_a/2^{n_b}$. Then, the decoding error probability of the inner decoder can be calculated at each iteration as

$$E_{in}^{(l)} = \sum_{\bar{d}} p_{\bar{D}}^{(l)} (\bar{d}) \quad \text{for} \quad \bar{d} \in [-L_a, -L_a + \Delta, \cdots, 0]. \tag{4.8}$$

4.3.2 DDE for Outer Code

Let $q_{\bar{u}_0}$ be the PMF of the initial LLRs for the outer decoder. Then, we can write $q_{\bar{u}_0} = p_{\bar{D}}^{(l)}$. Let $\Omega^{(o)}(x)$ and $\Lambda^{(o)}(x)$ be the node degree distributions, and $\omega^{(o)}(x)$ and $\lambda^{(o)}(x)$ be the edge degree distributions, respectively for the outer code. Let $q_{\bar{u}}^{(l)}$ be the PMF of

the message through a randomly chosen edge from a CN to a VN at the lth outer decoding iteration. Let $d_{max}^{(o)}$ be the maximum VN degree of the outer code. Then, the PMF of the decision variable associated with the outer LDPC code at the lth iteration is $q_{\bar{D}}^{(l)} = \sum_{i=1}^{d_{max}^{(o)}} \Lambda_i^{(o)} \left(q_{\bar{u}_0} \bigotimes \left(\bigotimes^i q_{\bar{u}}^{(l)} \right) \right)$. Finally, the overall decoding error probability of the Raptor codes at each outer iteration is calculated as

$$E^{(l)} = \sum_{\bar{d}} q_{\bar{D}}^{(l)} \left(\bar{d} \right) \quad \text{for} \quad \bar{d} \in [-L_a, -L_a + \Delta, \cdots, 0]. \tag{4.9}$$

4.3.3 Asymptotic Curves

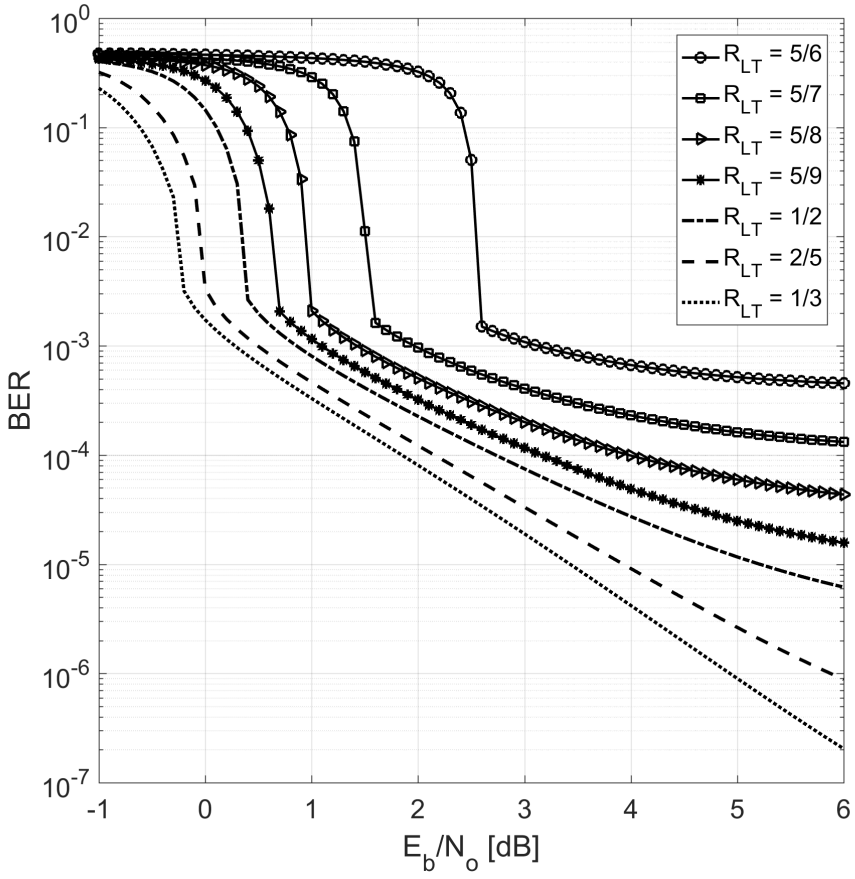

Figure 4.5. Asymptotic performance of LT codes. The output degree distribution used is $\Omega_{\mathrm{T}}(x)$.

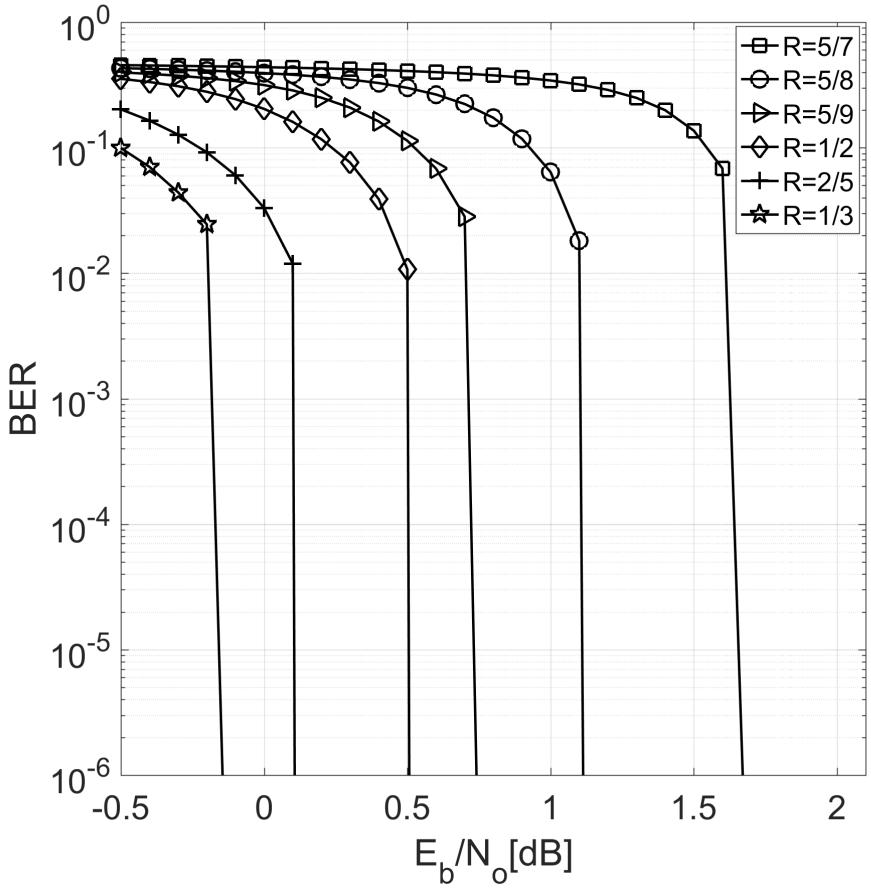

Figure 4.6. Asymptotic performance of Raptor codes. The output degree distribution used is $\Omega_{\mathrm{T}}(x)$.

The asymptotic performance of LT codes obtained using the DDE method is presented in Figure 4.5 for various realized code-rates. If we do not consider the outer code, i.e., considering $\mathrm{R_O} = 1$, the two-step DDE method discussed in Section 4.3 actually becomes the DDE method for LT codes only. Using this method, we can completely analyse the performance of LT codes in BIAWGN channels. We see that these codes have error floors and are asymptotically bad.

Using the two-step DDE method discussed in Section 4.3, we have obtained the asymptotic performance of Raptor codes for various realized code-rates as shown in Figure 4.6. We observed no error floors for the Raptor codes at any realized code-rates presented. We

see that the use of high-rate outer LDPC codes not only helps eliminate the error floors but also makes the Raptor codes capacity approaching codes. The ODD used for each realized code-rates is the same $\Omega_{\mathrm{T}}(x)$. It is very interesting that the same ODD provides impressive decoding performance for many different realized code-rates. Throughout this chapter, unless otherwise mentioned, the number of quantization bits used is 10, the range of LLRs used is $[-25, 25]$, and the maximum number of iterations (N_T) used is 200 for all the asymptotic results presented.

4.3.4 Necessary Condition for Successful Decoding

Considering non-discretized messages, i.e., under density evolution, (4.8) can be rewritten as $E_{in}^{(l)} = \int_{-\infty}^{0} p_D^{(l)} dD$. Let $M_D^{(l)}$ be the mean associated with $p_D^{(l)}$. Under the symmetry condition as proved in [15], we have $p_D^{(l)} \sim \mathcal{N}\left(M_D^{(l)}, 2M_D^{(l)}\right)$. Hence, we can write $E_{in}^{(l)}$ as

$$E_{in}^{(l)} = \int_{-\infty}^{0} p_D^{(l)} dD = 1 - \mathrm{Q}\left(\frac{0 - M_D^{(l)}}{\sqrt{2M_D^{(l)}}}\right) = \mathrm{Q}\left(\sqrt{\frac{M_D^{(l)}}{2}}\right). \tag{4.10}$$

In the two-stage decoding, we can consider the encoder-channel-decoder chain of the inner code as a super-channel [24] as shown in Figure 3.3. The LLR values obtained from the inner decoder serve as the channel estimates for the outer decoder and have the PDF of $\mathcal{N}\left(M_D^{(l)}, 2M_D^{(l)}\right)$. Therefore, the equivalent noise parameter, i.e., the noise standard deviation, associated with the super-channel assuming the unit signal power, can be calculated as $\sqrt{2/M_D^{(l)}}$. Hence, the necessary condition for the successful decoding of the outer decoder becomes $\sqrt{2/M_D^{(l)}} \leq \sigma_{th}^{(0)}$, where $\sigma_{th}^{(o)}$ is the decoding threshold of the high-rate LDPC code used as the outer code. Using the Q-function, i.e., the upper tail function of the standard Gaussian distribution, this necessary condition can be rewritten as

$$\mathrm{Q}\left(\sqrt{\frac{M_D^{(l)}}{2}}\right) \leq \mathrm{Q}\left(\frac{1}{\sigma_{th}^{(o)}}\right). \tag{4.11}$$

From (4.10) and (4.11), we get

$$E_{in}^{(l)} \leq \mathrm{Q}\left(\frac{1}{\sigma_{th}^{(o)}}\right). \tag{4.12}$$

In summary, (4.12) gives the necessary condition that a Raptor code must satisfy in order to be successfully decoded in a BIAWGN channel using the SPA. We call $\mathrm{Q}\left(1/\sigma_{th}^{(o)}\right)$ the critical BER. Once the inner decoder achieves the critical BER, the outer decoder drives the overall BER down to an infinitesimally small value. In the Raptor codes, the outer code used is a high-rate LDPC code. The decoding threshold of 50/51-rate regular (d_v, d_c)-LDPC code calculated using the DDE method and the corresponding critical BER is presented in Table 3.3.

4.3.5 Verification of the Necessary Condition

The asymptotic performance in Figure 4.7 clearly depicts that until the inner decoder achieves the critical BER, the outer decoder cannot enhance the overall performance. However, when the inner decoder achieves the critical BER, the outer decoder drives the overall BER to an infinitesimally small value. The error floors of the inner LT code are completely eradicated with the use of outer LDPC code. It is very important to note that the Raptor codes of various realized code-rates using the same ODD $\Omega_{\mathrm{T}}(x)$ are performing within 0.4 dB to the respective Shannon limits as reported in Table 4.2. The $\Omega_{\mathrm{T}}(x)$ used, as given by (4.4), is from [4] and has $\Omega_T'(1) = 5.87$.

Table 4.2. DDE thresholds for Raptor codes. The output degree distribution used is $\Omega_{\mathrm{T}}(x)$.

Code-rate	Shannon limit (dB)	Threshold (dB)	Gap (dB)
5/7	1.37	1.70	0.33
5/8	0.82	1.12	0.30
5/9	0.47	0.76	0.29
1/2	0.19	0.51	0.32
2/5	-0.24	0.11	0.35
1/3	-0.50	-0.12	0.38

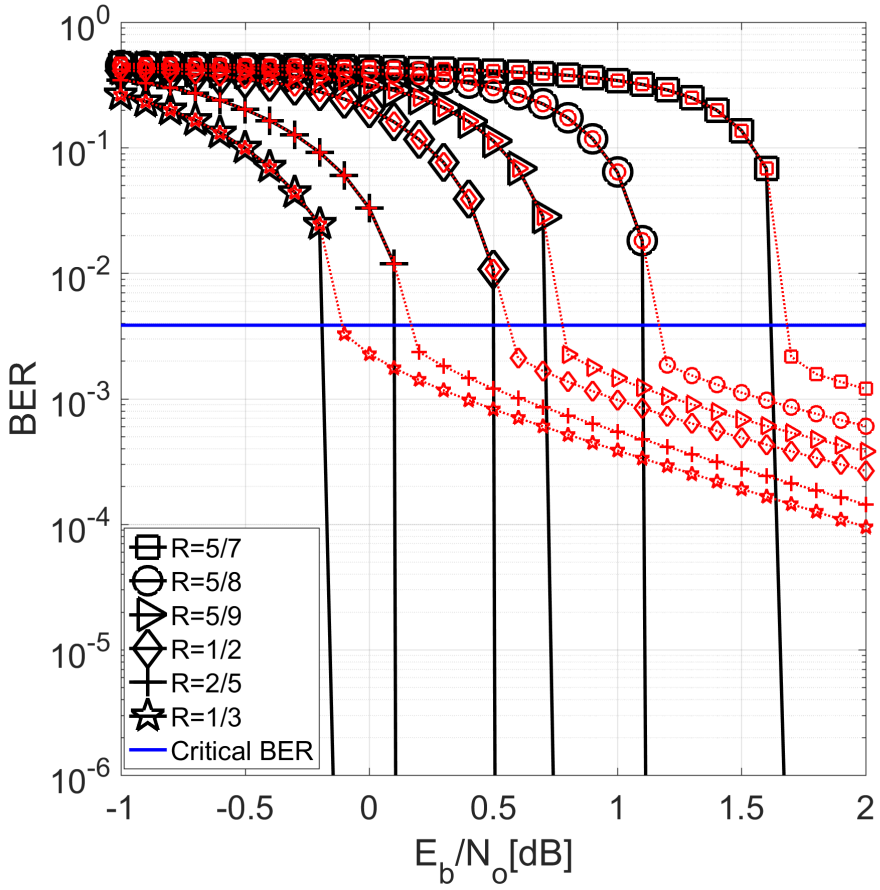

Figure 4.7. Asymptotic performance of Raptor codes. The output degree distribution used is $\Omega_{\mathrm{T}}(x)$. The dotted lines are the BERs obtained after inner decoding and the solid lines are those after outer decoding.

4.4 Critical BER-based Optimization Design

The main objective of optimization design is to obtain an optimized ODD such that $E^{(l)}$ of (4.9) tends to zero at the lowest possible E_b/N_o. Since achieving an extremely small BER by a Raptor code is equivalent to achieving the critical BER by the inner code as proved in Section 4.3.4, the optimization objective boils down to obtaining an optimized ODD such that $E_{in} \leq \mathrm{Q}\left(1/\sigma_{th}^{(o)}\right)$ is achieved by the inner decoder at the lowest possible E_b/N_o. This lowest E_b/N_o becomes the decoding threshold of the Raptor code.

4.4.1 Optimization Process

We first fix the outer code as 50/51-rate regular (4,204)-LDPC code. Hence, the critical BER is known to be 3.831×10^{-3} from Table 3.3. We then optimize the inner code, i.e., search optimized ODD for the inner LT code. The constraints to be satisfied are: (a) $\Omega(1) = 1$, (b) $0 \leq \Omega_i \leq 1,\ i = 1, 2, \cdots k$, (c) $E_{in}^{(l)} \leq \mathrm{Q}\left(1/\sigma_{th}^{(o)}\right)$, and (d) $\omega(1-\lambda(1-z)) > z,\ \forall z \in [0, 1)$. $\Lambda(x)$ is known to be binomial as given by (4.6). The constraints (a) and (b) guarantee a valid degree distribution. The constraint (c) is the necessary condition for the successful decoding of the Raptor codes. The design constraint (d) discussed and developed in [31, 32] is used for designing such rateless codes. This constraint ensures the successful completion of SP decoding process.

The optimization process is better understood by considering a particular realized code-rate. Let us consider a half rate Raptor code. We know from Table 4.2 that the half-rate Raptor code using $\Omega_{\mathrm{T}}(x)$ has a decoding threshold of 0.51 dB while the Shannon limit is 0.19 dB. Therefore, regarding the optimization, the range we need to consider is $[0.19, 0.51)$ dB. At equally spaced points in that range, we minimize $E_{in}^{(l)}$ of (4.8) subject to our constraints. We then consider the lowest E_b/N_o in that range at which our constraints are satisfied as the decoding threshold and the corresponding distribution as the optimized ODD. Due to the monotonicity of the threshold [28], in practice, we speeded up the search process by using the bisection search at a desired level of precision. The same process can be used to obtain optimized ODD for the codes of any realized code-rates.

4.4.2 Results

The optimized ODD and decoding thresholds obtained using our approach for the Raptor codes of various realized code-rates are presented in Table 4.3. Note that optimized ODDs are obtained by considering the R-rate Raptor code with 50/51 rate $(4, 204)$-LDPC code as outer code and $\mathrm{R_I}$-rate inner LT code. Figure 4.8 and Table 4.3 clearly depict that the gap to the Shannon limit in each case is further reduced. Furthermore, the simulation results

Table 4.3. Optimized output degree distributions, and their corresponding decoding thresholds and gap to the Shannon limit.

Code-rate	5/7	5/8	5/9	1/2	2/5	1/3
Ω_1	0.00959	0.00962	0.00955	0.00967	0.00942	0.00896
Ω_2	0.47527	0.46539	0.45963	0.45025	0.44283	0.44293
Ω_3	0.19096	0.19711	0.18665	0.20937	0.18359	0.13185
Ω_4	0.04689	0.04662	0.07334	0.02332	0.08961	0.21253
Ω_5	0.12159	0.11969	0.10349	0.14735	0.09978	0.00602
Ω_{10}	0.02441	0.05638	0.04876	-	0.04890	0.14513
Ω_{11}	0.08605	0.05655	0.07067	0.11249	0.07609	-
Ω_{30}	-	0.00902	-	-	-	-
Ω_{40}	0.04523	0.03961	0.04667	0.04755	0.04978	0.05257
Ω_{50}	-	-	0.00122	-	-	-
$\Omega'(1)$	5.328	5.358	5.493	5.508	5.621	5.725
Threshold	1.65 dB	1.05 dB	0.66 dB	0.38 dB	-0.06 dB	-0.33 dB
Gap	0.28 dB	0.23 dB	0.19 dB	0.19 dB	0.18 dB	0.17 dB

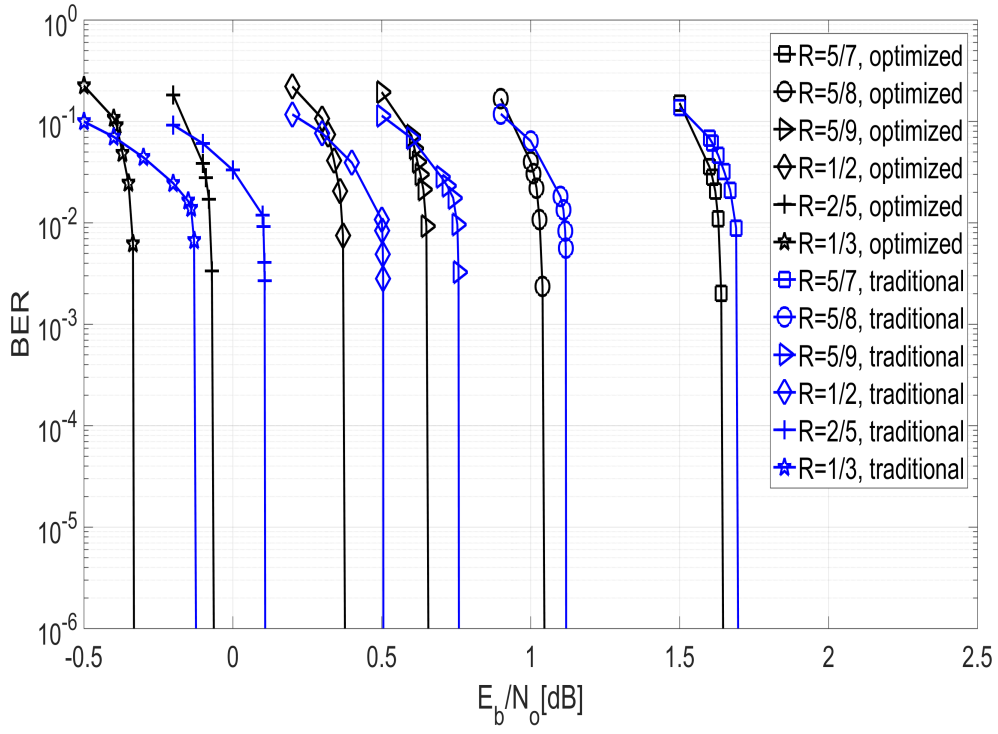

Figure 4.8. Asymptotic performance of our optimized Raptor codes vs. traditional Raptor codes using $\Omega_{\mathrm{T}}(x)$.

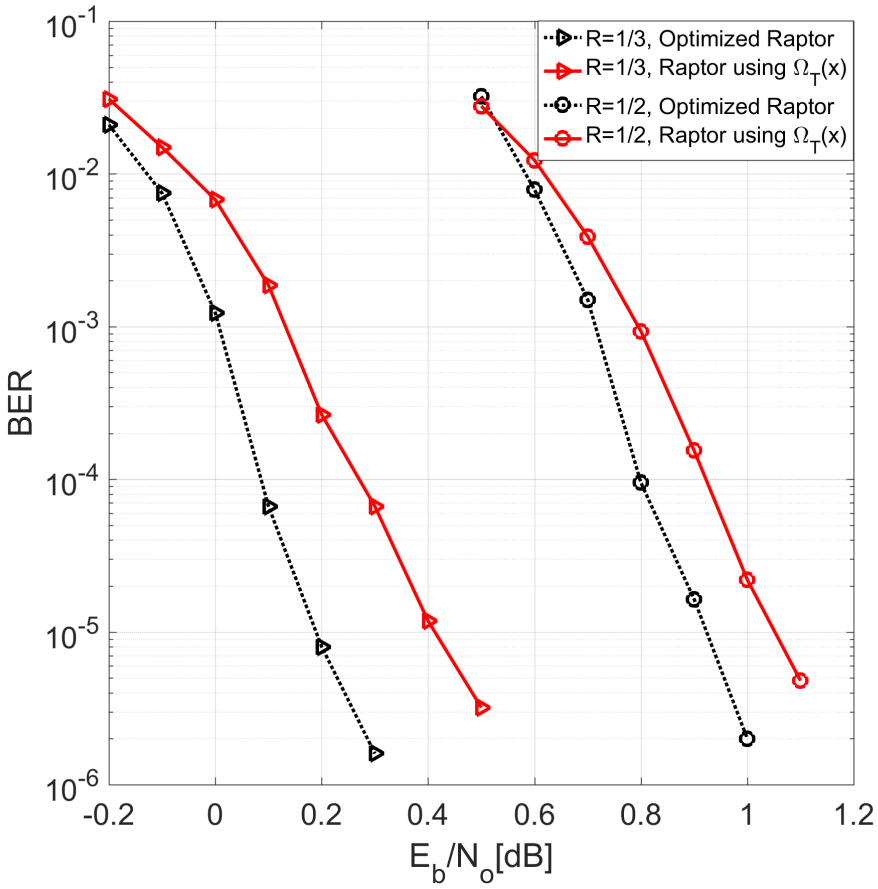

Figure 4.9. BER performance of our optimized Raptor codes vs. traditional Raptor codes using $\Omega_{\mathrm{T}}(x)$. k=10000.

presented in Figure 4.9 show that our optimized Raptor codes clearly outperform those using $\Omega_{\mathrm{T}}(x)$. These results verify the effectiveness of our critical BER-based optimization approach in obtaining optimized ODD. We also want to emphasize that for the optimization design part, we have considered $N_T = 200$. Considering higher N_T as in LDPC design can provide even better decoding thresholds.

4.5 Systematic LT and Systematic Raptor Codes

The systematic versions of LT and Raptor codes have also received attention since they are efficient in practical applications and also provide lower encoding and decoding complexities. Different ways of constructing systematic LT and systematic Raptor codes for

BEC are proposed in [33, 34]. However, for noisy channels, SLT codes are constructed in a straightforward manner, i.e., the encoded output bits consist of both the original source bits and LT encoded bits. This was first proposed in [35] and used RSD as the output degree distribution. A modified version of RSD named as truncated RSD is proposed in [36] for SLT codes with better performance. When a SLT code and a high-rate LDPC code are serially concatenated, the resulting code is called the SR code.

4.5.1 Encoding and Decoding of SLT Codes

The only difference between the LT encoding and SLT encoding is that the output bits generated by the SLT encoding also includes the original source bits. The LT encoding process used to generate the LT encoded bits are exactly the same in both the cases. Hence, the output bits are represented as $\boldsymbol{o} = [\boldsymbol{s}\ \boldsymbol{e}] = [s_1, s_2, \cdots, s_k, e_1, e_2, \cdots, e_m]$, where $\boldsymbol{e} = [e_1, e_2, \cdots, e_m]$ are the LT encoded bits generated. Again, due to the random LT encoding process involved, any number of LT encoded bits can be generated as required. Thus, any required code-rate $k/(k+m)$ can be realized by controlling m.

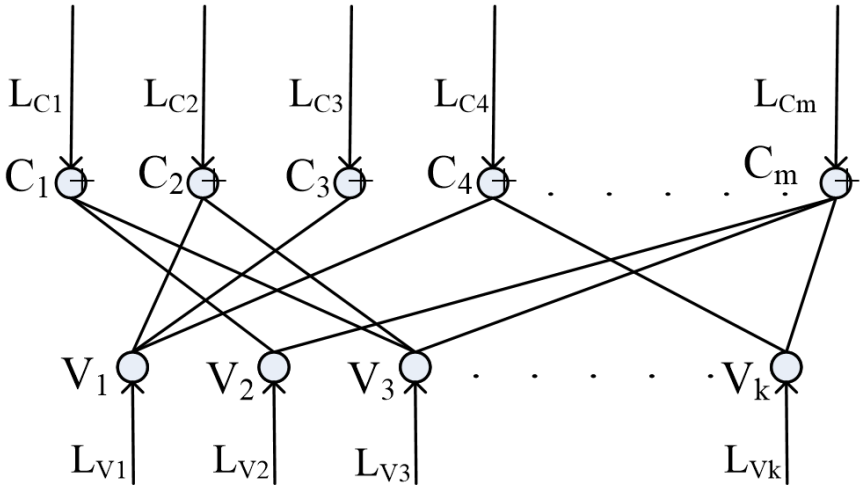

Figure 4.10. Bipartite graph used for the decoding of SLT codes using SPA.

For the decoding, the SP decoding is run over the decoding graph as shown in Figure 4.10,

where the CNs $C_1, C_2, \cdots, C_{n-k}$ represent the LT encoded bits while the VNs $V_1, V_2, \cdots, V_k$ represent the original source bits. This decoding graph is the irregular version of LDGM's decoding graph. Moreover, SLT codes are the irregular rateless counterpart of LDGM codes. Therefore, the product rule, the sum rule, and the decision rule used for the decoding of SLT codes are exactly the same as those of the LDGM codes presented in Chapter 3.

4.5.2 Encoding and Decoding of SR Codes

Since SR codes are the serial concatenation of SLT codes and a high-rate LDPC code, the encoding process involves two-stages as shown in Figure 4.11. The k source bits are initially encoded using a high-rate LDPC precode to generate k' intermediate bits. Let $\boldsymbol{m} = [m_1, m_2, \cdots, m_{k'}]$ represent k' LDPC encoded intermediate bits. Since the LDPC precode being used is also systematic, the first k intermediate bits are in fact the original source bits. These intermediate bits are further encoded using LT encoding to generate LT encoded bits. During the transmission, all the intermediate bits are transmitted followed by the LT encoded bits.

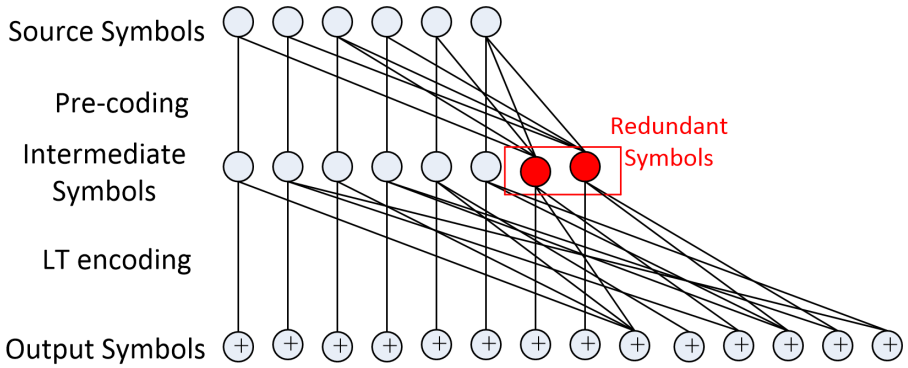

Figure 4.11. Graphical representation of systematic Raptor Codes.

Since SR codes are serially concatenated codes, we use the same two-stage decoding as in SCLDGM codes. We first run the SP decoding over the decoding graph associated with the inner SLT codes. The inner decoding graph used is constructed by using LT encoded

bits as CNs and intermediate bits as VNs. After inner decoding, the SP decoding of outer LDPC code is conducted. The outer decoding graph is constructed based on the associated (H)-matrix of the used LDPC code, where again the intermediate bits represent the VNs while the CNs are determined by the check equations obtained from the H-matrix.

4.6 Asymptotic Analysis of SLT and SR Codes

Both the systematic and non-systematic Raptor codes use the same outer LDPC code. The only difference is that the SR codes have inner SLT codes due to which the intermediate bits are also transmitted over a communication channel. Hence, the channel estimates associated with the intermediate bits that form the VNs are available during the decoding which are known to be Gaussian distributed with $2/\sigma^2$ mean and $4/\sigma^2$ variance, i.e., $p_{\bar{u}_0} \sim \mathcal{N}\left(2/\sigma^2, 4/\sigma^2\right)$. However, for non-systematic version, these channel estimates are absent and hence, $p_{\bar{u}_0} = \sum_{j=-\infty}^{+\infty} \delta[j]$ was used. Due to the systematic nature, the inner decoding graph of SR code has $k/\mathrm{R} - k'$ CNs which for the non-systematic case is k/R. Hence, by easily incorporating these two modifications in the two-step DDE method discussed in Section 4.3, we can obtain the asymptotic curves for SR codes easily.

The exact asymptotic curves for SLT codes of various realized code-rates are presented in Figure 4.12 while that of SR codes are presented in Figure 4.13 respectively. The ODD used is $\Omega_{\mathrm{T}}(x)$ from [4] as given by (4.4). The precode used is a regular LDPC code of code-rate 50/51 with degree 4 VNs. The asymptotic results show that the SR codes have huge performance improvements over the SLT codes for all the realized code-rates considered. We have observed no error floors for the SR codes. The decoding thresholds of the SR codes using $\Omega_{\mathrm{T}}(x)$ for the realized code-rates of 2/3, 2/4, 2/5, and 2/6 are found to be 4.42 dB, 3.10 dB, 2.07 dB, and 1.42 dB respectively.

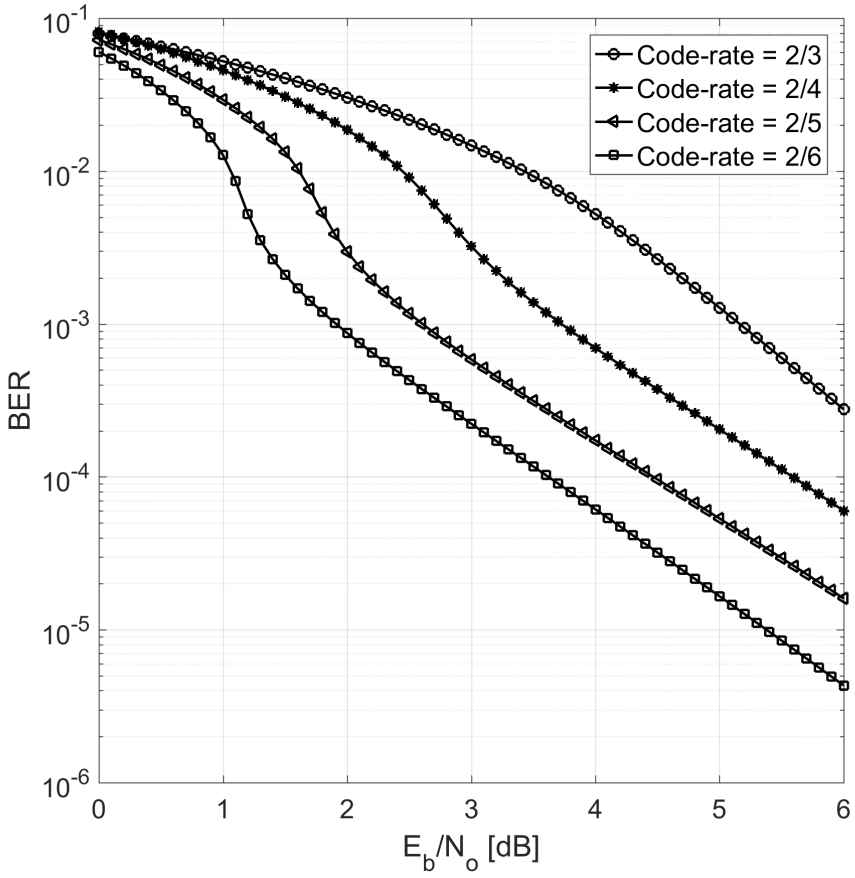

Figure 4.12. Asymptotic performance of systematic LT codes over BIAWGN channels. The output degree distribution used is $\Omega_{\mathrm{T}}(x)$.

4.7 Optimization Design of SR Codes

Since the outer code we have used for the SR codes is the same 50/51 rate LDPC code as in non-systematic Raptor codes, the necessary condition for the successful decoding remains the same. It means that the critical BER that the inner decoder of the SR codes must at least produce is as presented in Table 3.3. Hence, we use the same critical-BER based optimization approach used for Raptor codes as described in Section 4.4 and use it to obtain optimized ODD for SR codes by considering the two-step DDE of the SR codes.

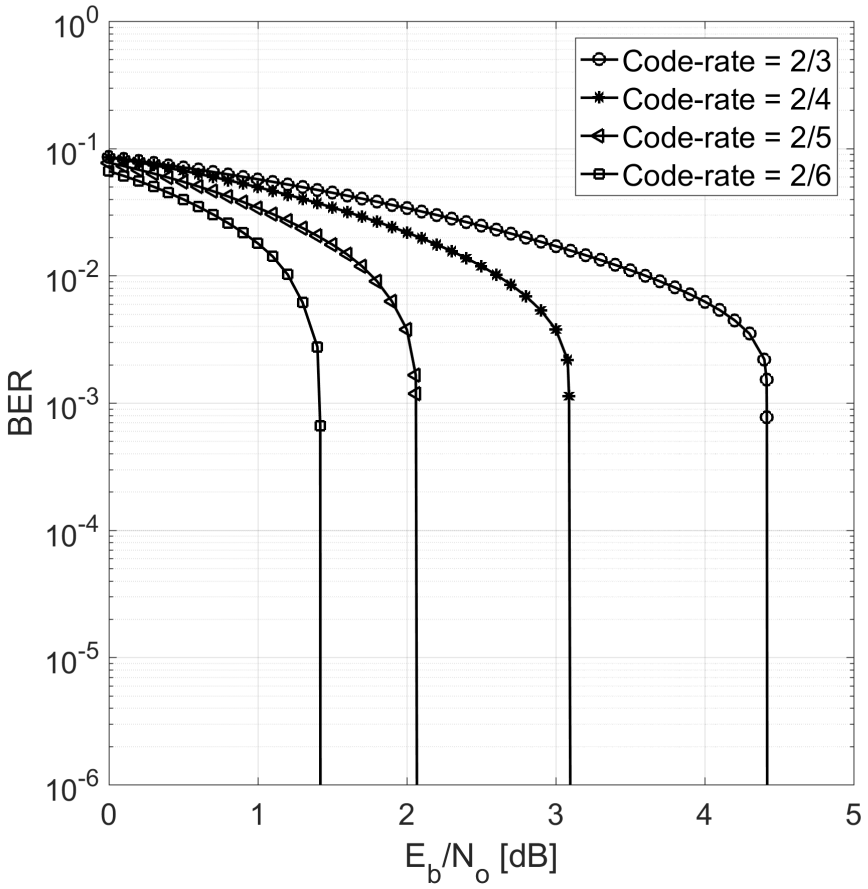

Figure 4.13. Asymptotic performance of systematic Raptor codes over BIAWGN channels. The output degree distribution used is $\Omega_{\mathrm{T}}(x)$.

4.7.1 Optimized Output Degree Distribution

The optimized output degree distributions we have obtained for the SR codes of various realized code-rates are presented in Table 4.3. The decoding threshold, the gap to the Shannon limit, and the improvement of our optimized ODD over $\Omega_{\mathrm{T}}(x)$ for various realized code-rates are also presented. We see that our optimized SR codes largely outperform the traditional SR codes. It is also observed that as the code-rate decreases, the gap to the Shannon limit further improves.

Table 4.4. Optimized output degree distributions and their corresponding decoding threshold, gap to the Shannon limit, and improvement over $\Omega_{\mathrm{T}}(x)$.

Code-rate (R)	2/3	2/4	2/5	2/6
Ω_1	0.00635	0.00477	0.00381	0.01527
Ω_2	0.35367	0.26101	0.22579	0.14492
Ω_3	0.01867	0.09240	-	-
Ω_4	0.20177	-	-	-
Ω_5	-	-	-	0.59406
Ω_6	-	-	0.48017	0.14058
Ω_7	-	-	0.20647	-
Ω_8	-	0.06913	-	-
Ω_9	-	0.51223	-	-
Ω_{20}	0.38134	-	-	-
Ω_{30}	0.03820	-	-	0.07743
Ω_{40}	-	-	0.05607	-
Ω_{50}	-	-	0.02769	0.02775
Ω_{60}	-	0.06046	-	-
Threshold	2.39 dB	0.88 dB	0.20 dB	-0.15 dB
Gap	1.33 dB	0.69 dB	0.44 dB	0.35 dB
Improvement	2.03 dB	2.22 dB	1.87 dB	1.57 dB

4.7.2 Simulation Results

The simulation results of our optimized SR codes of realized code-rates 1/2 and 1/3 are compared with those of traditional SR codes for different values of k in Figure 4.14 and Figure 4.15 respectively. Our optimized SR codes used the optimized ODDs presented in Table 4.4, while traditional SR codes used the $\Omega_{\mathrm{T}}(x)$. The maximum number of iterations allowed for the inner and outer codes during the SP decoding are 200 and 50 respectively. We see from the graphs that our optimized SR codes largely outperform the traditional SR codes across a wide range of E_b/N_o values in both the cases. The improvements are observed for all the values of k considered. These simulation results further validate our asymptotic results.

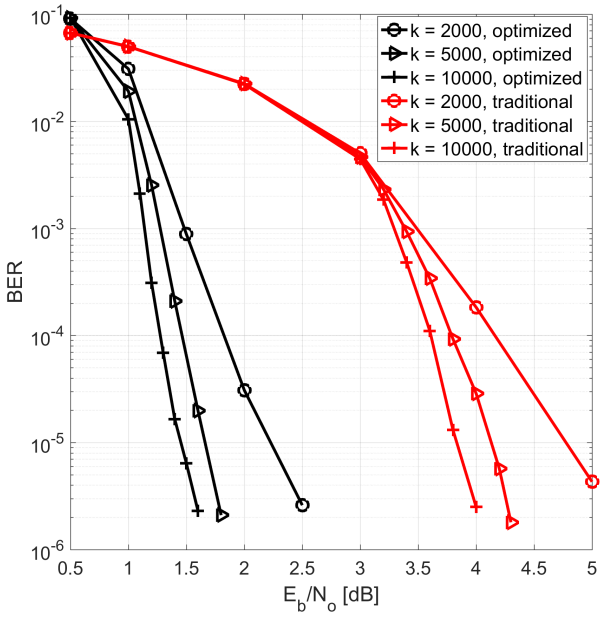

Figure 4.14. BER performance of optimized SR codes vs. traditional SR codes. Code-rate = 1/2.

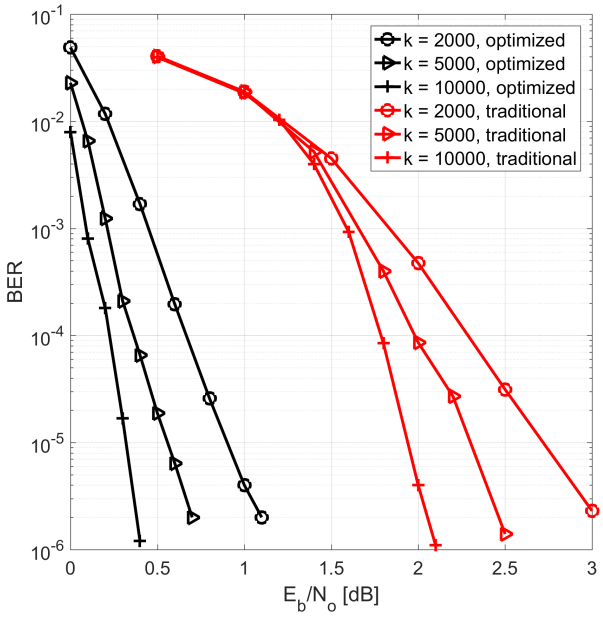

Figure 4.15. BER performance of optimized SR codes vs. traditional SR codes. Code-rate = 1/3.

CHAPTER 5

CONCLUSION

5.1 Conclusion

We accomplished the following tasks:

- For serially concatenated error correcting codes in which the SP decoding of the inner code is followed by the SP decoding of the outer code, we showed how the DDE method can be applied in two-steps to analyse their asymptotic behaviour.

- We developed and used the two-step DDE method to provide the exact asymptotic performance of SCLDGM codes, Raptor codes, and systematic Raptor codes.

- We presented the detailed error-floor analysis of both the LDGM and the SCLDGM codes.

- We proved a necessary condition for the successful decoding of such concatenated codes using two-step density evolution. While doing so, we showed that the inner decoder must at least produce a certain BER called the critical BER so that the outer decoder can provide an extremely small overall BER. Hence, handing decoding to the outer decoder without inner decoder achieving the critical BER is just the waste of decoding. This necessary condition was verified for both the SCLDGM and Raptor codes.

- We proposed a critical BER-based optimization technique to design capacity approaching SCLDGM codes.

- We used the critical BER-based optimization approach to design Raptor codes for BIAWGN channels that perform within 0.2 dB to the Shannon limit for most of the realized codes-rates. We showed through asymptotic curves and simulation results the superiority of our optimized Raptor codes over the traditional Raptor codes.

- We also designed optimized systematic Raptor codes that hugely outperformed the traditional systematic Raptor codes in BIAWGN channels.

BIBLIOGRAPHY

[1] John W. Byers, Michael Luby, Michael Mitzenmacher, and Ashutosh Rege. A digital fountain approach to reliable distribution of bulk data. In *Proc. ACM SIGCOMM '98 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, pages 56–67, 1998.

[2] J. W. Byers, M. Luby, and M. Mitzenmacher. A digital fountain approach to asynchronous reliable multicast. *IEEE Journal on Selected Areas in Communications*, 20(8):1528–1540, Oct 2002.

[3] M. Luby. LT codes. In *Proc. 43rd Symposium on Foundations of Computer Science*, FOCS '02, pages 271–280, Washington, DC, USA, 2002. IEEE Computer Society.

[4] A. Shokrollahi. Raptor codes. *IEEE Trans. Inf. Theory*, 52(6):2551–2567, 2006.

[5] R. Palanki and Jonathan S. Yedidia. Rateless codes on noisy channels. In *International Symposium on Information Theory*, 2004.

[6] O. Etesami and A. Shokrollahi. Raptor codes on binary memoryless symmetric channels. *IEEE Trans. Inf. Theory*, 52(5):2033–2051, 2006.

[7] I Hussain, Ming Xiao, and L.K. Rasmussen. Error floor analysis of LT codes over the additive white Gaussian noise channel. In *IEEE Global Telecommunications Conference (GLOBECOM)*, pages 1–5, Dec 2011.

[8] Shuang Tian, Yonghui Li, M. Shirvanimoghaddam, and B. Vucetic. A physical-layer rateless code for wireless channels. *IEEE Trans. Commun.*, 61(6):2117–2127, June 2013.

[9] M. Shirvanimoghaddam and S. Johnson. Raptor codes in the low SNR regime. *IEEE Trans. Commun.*, 64(11):4449–4460, Nov 2016.

[10] Z. Cheng, J. Castura, and Y. Mao. On the design of Raptor codes for binary-input Gaussian channels. *IEEE Trans. Commun.*, 57(11):3269–3277, Nov 2009.

[11] A. Venkiah, C. Poulliat, and D. Declercq. Analysis and design of Raptor codes for joint decoding using information content evolution. In *IEEE International Symposium on Information Theory*, pages 421–425, June 2007.

[12] R. Gallager. Low-density parity-check codes. *IEEE Trans. Inf. Theory*, pages 21–28, Jan. 1962.

[13] Sae-Young Chung, G. D. Forney, T. J. Richardson, and R. Urbanke. On the design of low-density parity-check codes within 0.0045 dB of the Shannon limit. *IEEE Commun. Lett.*, 5(2):58–60, Feb 2001.

[14] S.-Y. Chung. On the construction of some capacity-approaching coding schemes. *Ph.D. dissertation, MIT, Cambridge, MA, USA*, 2000.

[15] T. J. Richardson and R. L. Urbanke. The capacity of low-density parity-check codes under message-passing decoding. *IEEE Trans. Inf. Theory*, 47(2):599–618, Feb 2001.

[16] T. J. Richardson, M. A. Shokrollahi, and R. L. Urbanke. Design of capacity-approaching irregular low-density parity-check codes. *IEEE Trans. Inf. Theory*, 47(2):619–637, Feb 2001.

[17] J. Garcia-Frias and Wei Zhong. Approaching Shannon performance by iterative decoding of linear codes with low-density generator matrix. *IEEE Commun. Lett.*, 7(6):266–268, June 2003.

[18] K. Liu and J. Garcia-Frias. Error floor analysis in LDGM codes. In *IEEE International Symposium on Information Theory*, pages 734–738, June 2010.

[19] Wei Zhong, Huiqiong Chai, and J. Garcia-Frias. Approaching the Shannon limit through parallel concatenation of regular LDGM codes. In *Proc. International Symposium on Information Theory*, pages 1753–1757, Sept 2005.

[20] H. Jenkac, T. Mayer, T. Stockhammer, and W. Xu. Soft decoding of LT-codes for wireless broadcast. *Proc. IST Mobile Summit*, pages 1–5, 2005.

[21] D. J. C. MacKay. Good error-correcting codes based on very sparse matrices. *IEEE Trans. Inf. Theory*, 45(2):399–431, Mar 1999.

[22] N. Alon and M. Luby. A linear time erasure-resilient code with nearly optimal recovery. *IEEE Transactions on Information Theory*, 42(6):1732–1736, 1996.

[23] R.M Tanner. A recursive approach to low complexity codes,. *IEEE Trans. Inf. Theory*, pages 533–547, Sept. 1981.

[24] G. David Forney. Concatenated codes. *MIT Press, Cambridge*, 1966.

[25] William E. Ryan and Shu Lin. *Channel Codes*. Cambridge University Press, 2009.

[26] M. Gonzalez-Lopez, F. J. Vazquez-Araujo, L. Castedo, and J. Garcia-Frias. Serially-concatenated low-density generator matrix (SCLDGM) codes for transmission over AWGN and Rayleigh fading channels. *IEEE Trans. Wireless Commun.*, 6(8):2753–2758, August 2007.

[27] S. ten Brink. Convergence behavior of iteratively decoded parallel concatenated codes. *IEEE Trans. Commun.*, 49(10):1727–1737, Oct 2001.

[28] T. J. Richardson and R. L. Urbanke. *Modern Coding Theory*. Cambridge University Press, Mar 2008.

[29] Rainer Storn and Kenneth Price. Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 11:341–359, 1997.

[30] A. Shokrollahi and R. Storn. Design of efficient erasure codes with differential evolution. In *IEEE International Symposium on Information Theory*, 2000.

[31] Michael G. Luby, Michael Mitzenmacher, and M. Amin Shokrollahi. Analysis of random processes via And-Or tree evaluation. In *Proc. 9th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 364–373, 1998.

[32] M. Luby, M. Mitzenmacher, A. Shokrollahi, D. Spielman, and V. Stemann. Practical loss-resilient codes. In *Proc. 29th Symp. on Theory of Computing*, pages 150–159, 1997.

[33] X. Yuan and L. Ping. On systematic LT codes. *IEEE Commun. Lett.*, 12(9):681–683, September 2008.

[34] M. A. Shokrollahi and M. G. Luby. Systematic encoding and decoding of chain reaction codes. *United States Patent Serial Number 6909383*, June 2005.

[35] T. D. Nguyen, L. L. Yang, and L. Hanzo. Systematic Luby transform codes and their soft decoding. In *IEEE Workshop on Signal Processing Systems*, pages 67–72, Oct 2007.

[36] T. D. Nguyen, L. L. Yang, S. X. Ng, and L. Hanzo. An optimal degree distribution design and a conditional random integer generator for the systematic Luby transform coded wireless Internet. In *IEEE Wireless Communications and Networking Conference (WCNC)*, pages 243–248, March 2008.

VITA

Amrit Kharel obtained his bachelor of engineering degree in electronics and communication engineering from Tribhuvan University, Nepal in 2009. He joined the University of Mississippi, USA in January 2011 for his graduate studies. He received M.S. in engineering science with the emphasis in telecommunications in August 2013 and Ph.D. in engineering science with the emphasis in electrical engineering in May 2018. During his graduate studies, he worked as a teaching assistant, research assistant, and lab instructor at the department of electrical engineering. His research interest includes error-correcting codes, rateless codes, and wireless communications.