University of Alberta

DESIGN AND DECODING LDPC CODES WITH LOW COMPLEXITY

by

Chao Zheng

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements for the degree of

Master of Science

in

Communications

Department of Electrical and Computer Engineering

Spring 2012
Edmonton, Alberta

To my family

Abstract

This thesis presents low complexity design and decoding schemes for low density parity check (LDPC) codes. First, we consider the iterative decoding of LDPC codes on multiple-input-multiple-output bit-interleaved coded modulation (MIMO-BICM) channels and two-way relay channels. More specifically, we study the log-likelihood ratio (LLR) calculation under MIMO-BICM channels when perfect channel information is known and LLR calculation for two-way relay channels when no channel information is known at the receiver. We propose the optimum piece-wise linear approximation in the sense of maximizing the achievable rate of the channel. Second, we introduce a novel "universal" LDPC code design method. We design universal LDPC codes based on our method and show that, compared to existing methods, a lager percentage of capacity is obtained. Then, we propose two conjectures about the extreme distributions under min-sum decoding based on numerical observations.

Acknowledgements

I would like to express my sincerest gratitude towards my supervisor, Dr. Masoud Ardakani, for his guidance and support. This work would not have been possible without his valuable advices, kindness and encourages. I am extremely grateful to him for his tireless editorial effort that largely improved the quality of this thesis. I cannot imagine a supervisor more caring than Masoud.

I wish to thank my friend Raman Yazdani in our research group. I deeply appreciate his insightful ideas and his many hours spent with me discussing my research.

I also wish to thank my friends in Edmonton who made my life enjoyable. To name a few of many: Jing Dong, Hao Chen, Shuangshuang Han, Madushanka Soysa, Jingyuan Zhang and Ting Zhao. I would also acknowledge my friends in China who supported me all the way.

Last but not least, the special thanks goes to my parents whose endless support from thousands miles away made me feel them besides me.

Table of Contents

List of Tables

List of Figures

List of Symbols

k	Dimension of a linear block code
n	Block length of a linear block code
R	Code rate
$P(E)$	Probability of event E
ε	Erasure probability in BEC and crossover probability in BSC
σ	Standard deviation of the additive white Gaussian noise
$\mathcal{N}(\mu, \sigma^2)$	Gaussian distribution with mean μ and variance σ^2
l, llr	Log likelihood ratio
$\delta_t(x)$	The Dirac delta function at $x = t$
$\mathcal{C}$	Capacity
I	Mutual Information
$h(\cdot)$	The binary entropy function
$\mathcal{P}_e(\cdot)$	Error probability
$\mathcal{B}(\cdot)$	The Bhattacharyya parameter
$\mathcal{F}_2$	The binary field
G	Generator matrix of a linear block code
H	Parity-check matrix of a linear block code
v	A variable node
c	A check node
E	Number of edges in the graph
d_v	Maximum degree of the variable nodes
d_c	Maximum degree of the check nodes
$\lambda(x)$	Generating polynomial of the variable degree distribution
$\rho(x)$	Generating polynomial of the check degree distribution
$\mathcal{C}^n(\lambda(x), \rho(x))$	Ensemble of LDPC codes of length n and degree distributions $\lambda(x)$ and $\rho(x)$
$\hat{R}$	A measure of achievable rate
erfc$(\cdot)$	Complementary error function

Acronyms

LDPC Low-density Parity-check

RA Repeat-Accumulate

LLR Log-likelihood Ratio

SISO Single Input Single Output

MIMO Multiple Input Multiple Output

BICM Bit-interleaved Coded Modulation

CSI Channel State Information

BMC Binary-input Memoryless Channel

BIMS Binary-input Memoryless Symmetric Channel

BEC Binary Eraser Channel

BSC Binary Symmetric Channel

BIAWGN Binary-input Additive White Gaussian Noise

BPSK Binary Phase-Shift Keying

pdf Probability Density Function

bpcu Bit Per Channel Use

AWGN Additive White Gaussian Noise

pmf Probability Mass Function

FFT Fast Fourier Transform

EXIT Extrinsic Information Transfer

SNR Signal to Noise Ratio

BER Bit Error Rate

MMSE Minimum Mean Square Error Approximation

TRC Two-way Relay Channel

AF Amplify-and-Forward

DF Decode-and-Forward

OFDM Orthogonal Frequency Division Multiplexing

Chapter 1

Introduction

The focus of this work is on designing and decoding an extremely powerful class of error-correcting codes called *low-density parity-check (LDPC) code* with low complexity. LDPC codes have been shown to perform close to the capacity of many channels.

In this chapter, we introduce the field of study, the interesting problems in this area and also discuss some interesting problems with LDPC code which are tackled in this thesis.

1.1 Overview

People have always sought fast, reliable and secure ways to exchange information. However, the communication channel usually introduces noise and interference to distort the transmission. To improve the quality of data communication, error-correcting codes are proposed.

Using error-correction coding, some redundancy is added to the information by the channel encoder. The reason for adding redundancy is to combat the channel noise. The *rate* of a code is defined as the ratio of the number of input bits and the output bits of the channel encoder, which is always less than one. For example, one powerful class of error-correction codes is called *block codes* where the input bit-stream is partitioned into several k-bit blocks and each block is mapped to an n-bit ($n > k$) word called *codeword*. Thus, the rate of block codes is $R = \frac{k}{n} < 1$.

In 1948, C.Shannon introduced the limits of reliable transmission over unre-

liable channels in one of his remarkable papers [1]. Given a channel, Shannon showed that there is a limit on the maximum code rate, called *channel capacity*, below which the reliable data transmission is possible. He proved that there exist a code that can be used to transmit data with arbitrarily small probability of error if the rate of the code is below the capacity. The Shannon limit, which is defined as the minimum transmission power required to transmit reliably for a given code rate, is proposed to measure the power efficiency of a coding scheme, Thus, the ultimate goal of error-correction coding is to find practical capacity-approaching codes.

Shannon used random codes to prove the channel capacity theorem. However, since the decoding complexity of random codes grows exponentially with the block length, random coding is not suitable for practical use.

1.2 Codes Defined on Graphs and Iterative Decoding

A coding scheme with its rate close to the capacity was not developed until the discovery of Turbo codes in 1993 [2]. By *iterative message-passing* decoding algorithms, Turbo codes can approach the capacity with practical decoding complexity. This class of decoding algorithms is applicable to the codes defined on graphs. Due to their reasonable decoding complexity, iterative decoding and graphical codes have drawn much attention in the past decades.

One of the most attractive properties of iterative decoding algorithms is that their complexity grows linearly with the length of the code which means that the complexity per information bit is independent of the code length. Thus, it allows us to use long codewords with reasonable decoding complexity to design codes that approach the channel capacity.

The graphical understanding of codes started with Tanner graphs for linear codes [3]. Later, Wiberg found that the turbo decoder can be represented graphically as well [4]. Because of the research on turbo code and graphical codes, a class of codes called LDPC code was rediscovered after been forgotten for several decades. LDPC codes were first proposed by Gallager in his PhD thesis [5], but were considered too complex at the time of their discovery.

LDPC codes are block codes which have a sparse structure. LDPC codes drew a lot of attention because they have extremely good performance with reasonable complexity under iterative message passing algorithms. Due to these properties of LDPC codes, they became one of the most active research topics in coding theory.

LDPC codes are already used in some communication standards such as ETSI EN 302 307 for digital video broadcasting [6] and IEEE 802.16 WiMAX standard [7]. Moreover, the discovery of many new classes of codes defined on graphs are influenced by the structure of LDPC codes, such as repeat-accumulate (RA) codes [8], Luby transform codes [9] and Raptor codes [10].

Although lots of researchers have been working in the area of graphical codes, there are still many open problems under study. This thesis has addressed some of these problems and raises new questions.

1.2.1 The Main Theme of This Thesis

This thesis studies efficient methods for design and decoding of LDPC codes with low complexity. These problems are studied from a practical point of view. It is worth mentioning that although we focus on LDPC codes, some of these problems are discussed beyond their application to LDPC codes. In this section, we will have an overview of the problems that are addressed.

Decoding LDPC codes starts with computing log-likelihood ratios (LLRs) from the channel observation. Computing channel LLRs can be a cumbersome task, thus approximate LLRs are suggested in the literature. However, current approximation methods, which all focus on single-input-single-output (SISO) channels [11–14], cannot apply to multiple-input-multiple-output (MIMO) channels and two-way relay channels directly. What is more, the computation of the channel LLRs is too complicated in these cases. Thus, this is a problem that needs to be solved.

In this thesis, we propose a piece-wise linear method to calculate LLR for LDPC codes used over multiple-input-multiple-output bit-interleaved coded modulation (MIMO-BICM) channels when the channel parameters is known at the receiver and the two-way relay channel when the channel information is unknown at the receiver.

By maximizing the achievable transmission rate on the channel, our method has low-complexity and performance close to the actual performance obtained by true LLR calculation. It is worth mentioning that this method can be also applied to other codes decoded by iterative message-passing decoders such as turbo codes.

Although carefully designed LDPC codes have very good performance on many channels, the code design process can be complex and time-consuming since if the channel condition changes, code has to be designed again. In this work, we address this problem by providing a one-for-all solution. We try to design "universal codes" which can perform almost the same on all the channels with the same capacity. Codes designed by our method can achieve a high percentages of the capacity.

1.3 Thesis Outline

The thesis is organized as follow: Chapter 2 reviews the necessary background about iterative decoding, LDPC codes, their decoding algorithms and the existing analysis methods for these decoding algorithms.

In Chapter 3, we investigate non-SISO channels where calculating true LLRs is difficult. We first assume perfect channel state information (CSI) is known at the receiver for MIMO-BICM channel and we find the optimum piece-wise linear approximation of LLRs in the sense of maximizing the achievable rate of the channel. Then, we extend this piece-wise linear method to the two-way relay channel when no CSI is known at the receiver. In both cases, the maximum achievable rates under the proposed method are quite close to the achievable rate when true LLRs are applied. The success of our method can be also observed based on the error rate.

In Chapter 4, we propose a new method to design codes with good universal properties based on the information combining bounds which are developed recently. We use the stability condition of density evolution to analysis our method, and by designing codes based on the proposed method, we show that compared to the exiting methods, a higher percentage of capacity can be achieved.

We conclude the thesis in Chapter 5 by summarizing the contributions made in this work and by suggesting possible directions for future work.

Chapter 2

Preliminaries and Background

In this chapter, we briefly review the necessary background on transmission model, LDPC codes and their structure, different decoding algorithms and exiting analysis methods.

2.1 Transmission Model and Channel Coding

In Fig. 2.1, the block diagram of a generic digital communication system is depicted, which consists of a transmitter, a channel and a receiver. The transmitter mainly consists of a binary source, a channel encoder and a modulator. Channel is the medium through which the information is transmitted, such as free air, optical fibers, a network link, etc.. The receiver mainly consists of a demodulator, and a channel decoder. The ultimate goal of this system is to transmit a data stream from the transmitter (source) to the receiver (sink) quickly and reliably. In this thesis, we mainly concern with the channel encoder, decoder and the channel itself.

Definition 2.1 [*Binary-Input Memoryless Channel (BMC)*]:
A binary-input channel is defined as a system comprising a finite input alphabet $X \in \mathcal{X}$ whose members are binary symbols, a finite output alphabet $Y \in \mathcal{Y}$, and a set of conditional probability assignments between them, $P_{Y|X}$. A channel is said to be memoryless *if each channel output depends only on the current input.*

Remark 2.1 [*Channel symmetry*]: *The channel is said to be* output-symmetric *if*

$$P_{Y|X}(y|x) = P_{Y|X}(-y| - x) \tag{2.1}$$

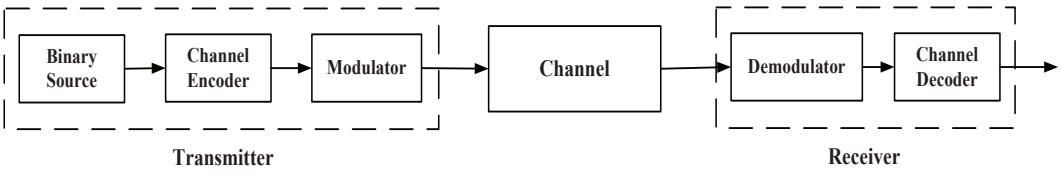

Figure 2.1: The block diagram of a generic digital communication system

The subset of BMC, which is output-symmetric, is also called binary-input memoryless symmetric channel (BIMS). The most famous channels in this class are the binary symmetric channel (BSC) and the binary erasure channel (BEC).

The BSC can be seen as a channel which flips every bit with probability ε, as shown in Fig. 2.2. It is denoted as BSC(ε). BSC(ε) is used in many studies because it is one of the simplest noisy channels to analyze. When hard decisions are made at the receiver, i.e., when the channel output values are quantized into two values, the channel can be seen as a BSC. Thus, many communication channels can be reduced to a BSC. Moreover, being able to transmit effectively over the BSC can give rise to solutions for more complicated channels.

The BEC is also an important channel which mostly occurs in data networks. The output of the BEC is either correct with probability $1 - \varepsilon$ or erased with probability ε, as depicted in Fig 2.2. In this case, BEC(ε) is error free since when the receiver gets one bit, it is certain that it is correct. Many real word channels can be reduced to BEC(ε), such as the packet transmission between two nodes in a data network, where a packet is either decodable (detected with no error) or undecodable (completely useless).

Another important channel model which will be used in this work is the binary-input additive white Gaussian noise (BIAWGN) channel which adds a random real number to the binary input, $x \in \{-1, 1\}$. The additive noise is drawn according to the Gaussian distribution with zero mean and variance σ^2, denoted as $\mathcal{N}(0, \sigma^2)$.

Although the class of BIMS channels is very important in theoretical and practical use, it is worth mentioning that the channels considered in this thesis are not necessarily output-symmetric.

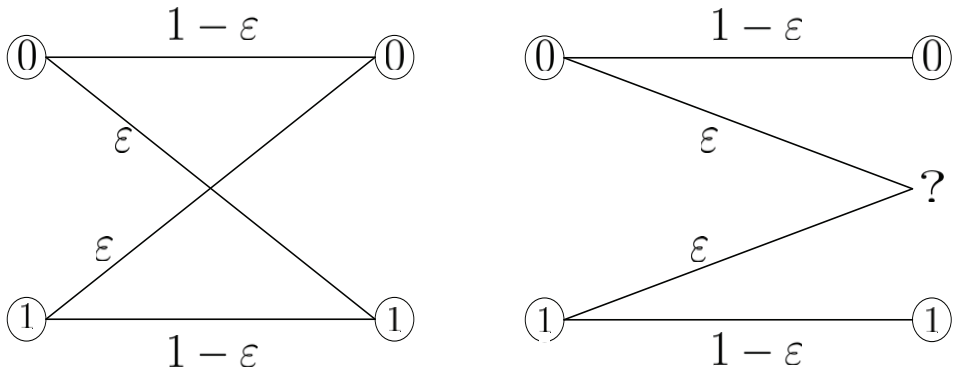

Figure 2.2: Left: The binary symmetric channel wit error probability ε. Right: The binary erasure channel with erasure probability ε

2.1.1 Log-Likelihood Ratios

Log-likelihood ratio (LLR), refereed as the soft information, has been widely used in the communication theory. It is defined as

$$\text{LLR} = l(y) = \frac{P_{X|Y}(0|y)}{P_{X|Y}(1|y)}. \tag{2.2}$$

where Y represents the channel output and X is the binary channel input. It is clear that LLR is a function of channel output Y, which can be denoted as $l(y)$. The input binary symbols, $\{0, 1\}$, are usually mapped to $\{+1,-1\}$ by the binary phase-shift keying (BPSK) modulation. Thus, $0 \leftrightarrow +1$ and $1 \leftrightarrow -1$. Using the channel symmetry condition, we have

$$\begin{aligned} l(y) &= \log \frac{P_{X|Y}(1|y)}{P_{X|Y}(-1|y)} \\ &= \log \frac{P_{Y|X}(y|1)}{P_{Y|X}(y|-1)} \\ &= \log \frac{P_{Y|X}(-y|-1)}{P_{Y|X}(-y|1)} \\ &= \log \frac{P_{X|Y}(-1|-y)}{P_{X|Y}(1|-y)} \\ &= -l(-y). \end{aligned}$$

The distribution of the LLR plays an important role in iterative decoding [15]. We denote it as $f(l)$. It is clear that there is a one-to-one correspondence between

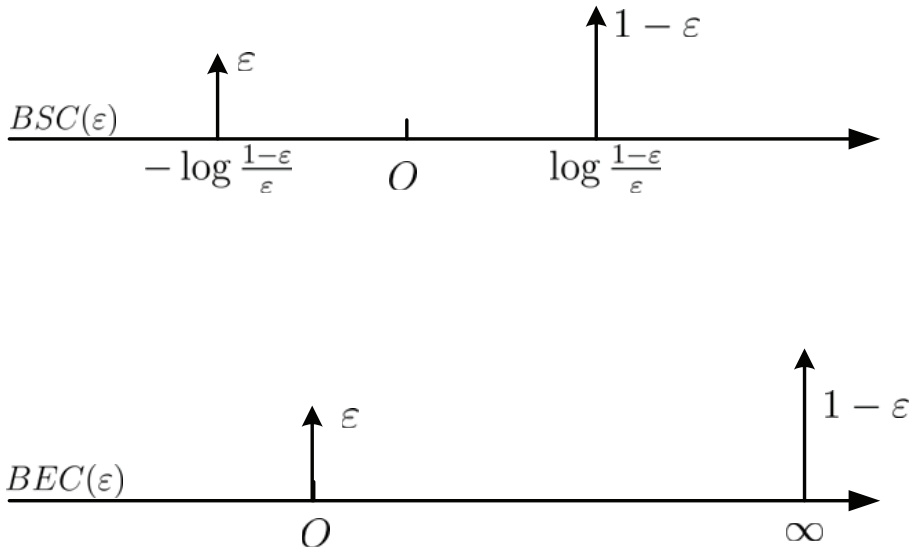

Figure 2.3: Upper: the pdf of LLRs for BSC(ε). Down: the pdf of LLRs for BEC(ε)

the probability density function (pdf) of LLR and the channel. For symmetric channels and assuming the all-zero codeword is transmitted, we have

$$\begin{aligned} f(-l) &= P_{Y|X}(y \in l^{-1}(-l)|1) \\ &= P_{Y|X}(-y \in l^{-1}(l)|1) \\ &= P_{Y|X}(y \in l^{-1}(l)|-1) \\ &= e^{-l}P_{Y|X}(y \in l^{-1}(l)|1) \\ &= e^{-l}f(l). \end{aligned}$$

which means that one side of the pdf can be obtained from the other side. Every such pdf satisfying this condition is called symmetric or consistent. For a Gaussian pdf $\mathcal{N}(\mu, \sigma^2)$, the symmetry condition can be simplified to $\sigma^2 = 2\mu$. Thus, by a symmetric Gaussian pdf, we mean a Gaussian pdf for which $\sigma^2 = 2\mu$.

Assuming all-zero codeword is transmitted, a channel can be fully specified by the pdf of its LLR. For example, the LLR pdf of the BSC(ε) is

$$f_{BSC}^{(\varepsilon)}(x) = (1-\varepsilon)\delta_{\log\frac{1-\varepsilon}{\varepsilon}}(x) + \varepsilon\delta_{-\log\frac{1-\varepsilon}{\varepsilon}}(x) \tag{2.3}$$

where $\delta_a(x)$ is the Dirac delta function at the location a. Also on the BEC(ε), the

LLR pdf is

$$f_{BEC}^{(\varepsilon)}(x) = \varepsilon\delta_0(x) + (1-\varepsilon)\delta_\infty(x). \tag{2.4}$$

These two pdfs are drawn in Fig .2.3.

One of the most important properties of a channel is its capacity. The definition of the capacity is provided in Definition 2.2.

Definition 2.2 [*Capacity*]: *For a channel, the information capacity is*

$$\mathcal{C} = \max_{p(x)} I(X;Y)$$

where $I(X;Y)$ is the mutual information between the input and output of the channel and the maximum is taken over all the input densities, $p(x)$.

For example, $\mathcal{C} = 1-\varepsilon$ for the BEC(ε), and $\mathcal{C} = 1 - h(\varepsilon)$ for the BSC(ε), where

$$h(\varepsilon) = -(1-\varepsilon)\log_2(1-\varepsilon) - \varepsilon\log_2(\varepsilon),$$

is called the binary entropy function [16].

Observed from (2.2), LLR is a sufficient statistic for X given Y, thus the capacity can also be represented from the LLR as

$$\mathcal{C} = \max_{p(x)} I(X;llr).$$

In this case, $\mathcal{C}$ can be interpreted as the capacity of the equivalent channel with input X, output llr and the channel transmission function $P(llr|X)$. Consider a BIMS, the density of LLR is also symmetric, denoted as $f(l)$. The capacity per channel use of this channel is given by [17]

$$\mathcal{C} \triangleq \mathcal{C}(f) = \int_{-\infty}^{+\infty} f(l)(1-\log_2(1+e^{-l}))dl. \tag{2.5}$$

The importance of the channel capacity is mainly due to the following theorem:

Theorem 2.1 [*Noisy-Channel Theorem* [1]]: *Given a channel with capacity $\mathcal{C}$, for any code rate $R < \mathcal{C}$, there exits encoding and decoding rules under which it is possible to have an arbitrary small probability of error.*
Conversely, for any rate $R > \mathcal{C}$, regardless of which encoding and decoding rules are used, the probability of error is bounded away from zero.

Thus, $\mathcal{C}$ is a fundamental limit for reliable data transmission on a channel.

2.1.2 Some Useful Parameters of A Symmetric Density

In this section, we introduce two important parameters of a symmetric density. For a BIMS channel with LLR pdf $f(l)$, since the channel is symmetric, the optimal input density is uniform over $\mathcal{X} \in \{-1, +1\}$ [18].

The error probability of $f(l)$ when $x = +1$ (all-zero codeword) is transmitted is given by $\mathcal{P}_e(\cdot)$ as

$$\begin{aligned}\mathcal{P}_e(f) &= P\{p(x=+1|y) < p(x=-1|y)\} + \frac{1}{2}P\{p(x=+1|y) = p(x=-1|y)\} \\ &= P\{l(y) < 0\} + \frac{1}{2}P(l(y)=0) \\ &= \int_{-\infty}^{0^-} f(l)dl + \int_{0^-}^{0^+} f(l)dl.\end{aligned}$$

Thus the error probability can be seen as the area of the negative tail of the density. For a symmetric density, $\mathcal{P}_e(f)$ can be written as

$$\mathcal{P}_e(f) = \frac{1}{2}\int_{-\infty}^{+\infty} f(l)e^{-(\frac{l}{2}+|\frac{l}{2}|)}dl.$$

Another important function for a symmetric LLR density associate is called the Bhattacharyya parameter [17], which will be used in Chapter 4. Suppose $f(l)$ is a symmetric LLR density and $x = +1$ is transmitted, the parameter is defined as

Definition 2.3 [*The Bhattacharyya Parameter*]:

The Bhattacharyya parameter associated with the symmetric density, $f(l)$, denoted as $\mathcal{B}(f)$ is

$$\mathcal{B}(f) = \int_{-\infty}^{+\infty} f(l)e^{-l/2}dl. \tag{2.6}$$

Remark 2.2 [*Extremes of the Bhattcharyya Parameter [17]*]: *For a arbitrary symmetric LLR density $f(l)$, we have*

$$2\mathcal{P}_e(f) \leq \mathcal{B}(f) \leq 2\sqrt{\mathcal{P}_e(f)(1-\mathcal{P}_e(f))}, \tag{2.7}$$

where $\mathcal{P}_e(f)$ is the error probability associated with the symmetric density f. Note that lower and upper bounds on the $\mathcal{B}(f)$ given in (2.7) are satisfied with equality for a BEC and BSC, respectively.

2.2 LDPC Codes: Graphical Representation

In this section, we introduce basic concepts of linear block codes and the structure of LDPC codes. In this thesis, all the codewords and information sources are defined in the binary field, denoted as $\mathcal{F}_2$.

2.2.1 Linear Block Codes

Linear block codes are one of the richest classes of codes which have practical encoding and decoding complexity. In block coding, the encoder splits the information sequence into blocks of fixed length k, called message blocks, $\mathbf{u}$, representing k information bits. Thus, there are a total of 2^k codewords corresponding to the 2^k possible message blocks. Therefore, a linear block codes is defined as:

Definition 2.4 [*Linear Block Codes [17]*]: *An (n,k) block code is a transformation of message blocks of length k according to a pre-defined rule into blocks of length n $(n > k)$, called codewords. A block code is linear if and only if any linear combination of codewords is also a codeword.*

Let

$$\mathbf{u} = [\mathbf{u_1}, \mathbf{u_2}, \ldots, \mathbf{u_k}]^{\mathbf{T}} \in \mathcal{F}_{\mathbf{2}}^{\mathbf{k}}$$

be the vector of information bits. We define the *generator matrix* $\mathbf{G}$ *of an* (n, k) linear block code as

$$\mathbf{G} = [\mathbf{g_1}, \mathbf{g_2}, \ldots, \mathbf{g_k}]^{\mathbf{T}} \in \mathcal{F}_{\mathbf{2}}^{\mathbf{k}\times\mathbf{n}}$$

which generates all the codewords. Let $\mathbf{v}$ denote the codeword, then

$$\begin{aligned} \mathbf{v} &= \mathbf{u} \cdot \mathbf{G} \\ &= \sum_{i=1}^{k} u_i g_i. \end{aligned} \tag{2.8}$$

Thus, linear block codes are fully specified by the rows of their generator matrix $\mathbf{G}$.

Another matrix which is useful in the decoding process of linear block codes is called the *parity-check matrix*, denoted as $\mathbf{H}$. A vector $\mathbf{v}$ of length n is a codeword

Information sequences	Codewords
(000)	(000000)
(001)	(001011)
(010)	(010101)
(011)	(011110)
(100)	(100110)
(101)	(101101)
(110)	(110011)
(111)	(111000)

Table 2.1: The information sequences and their corresponding codewords for a (6,3) linear block code

if and only if $\mathbf{v} \cdot \mathbf{H^T} = \mathbf{0}$ (parity check condition), where

$$\mathbf{H} = [\mathbf{h_1}, \mathbf{h_2}, \ldots, \mathbf{h_{n-k}}]^{\mathbf{T}} \in \mathcal{F}_{\mathbf{2}}^{(\mathbf{n-k})\times\mathbf{n}}$$

In fact, the rows of $\mathbf{H}$ generate the null space of $\mathbf{G}$, i.e., $\mathbf{G} \cdot \mathbf{H^T} = \mathbf{0}$. Thus, in the decoder, if the received codeword does not satisfy the parity check condition, it means that errors have occurred during the transmission.

Example 2.1 *Consider a (6,3) linear block codes, which has 3-bit messages and 6-bit codewords. The generator matrix G and parity-check matrix H of this code are given by*

$$\mathbf{G} = \begin{pmatrix} 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}$$

and

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

Thus, we can obtain all the possible codewords according to (2.8), which are listed in the Table 2.1

The parity-check equations are given as

$$c_0 \oplus c_1 \oplus c_3 = 0 \tag{2.9}$$

$$c_0 \oplus c_2 \oplus c_4 = 0 \tag{2.10}$$

$$c_1 \oplus c_2 \oplus c_5 = 0 \tag{2.11}$$

where $\oplus$ denotes the addition in $\mathcal{F}_2$.

2.2.2 LDPC Codes: Structure

An LDPC code is a special kind of linear block codes which has a sparse parity-check matrix. By sparse parity-check matrix, we mean that the number of nonzero entries in the parity-check matrix $\mathbf{H}$ is much smaller that the total number of entries. Although LDPC codes can be represented by their generator and party-check matrices like other linear block codes, using graphical representation gives us a more efficient way to analyze their iterative decoding algorithms.

The graphical representation of linear block codes started with Tanner graphs [3]. Later on, people focused on factor graphs due to their more general nature [19]. A factor graph is a bipartite graph whose nodes are partitioned to two groups, variable nodes and function (check) nodes. The *degree* of a particular node is the number of edges connected to that node.

The variable nodes v_i, which are binary variables $\{0,1\}$, represent codeword bits and the check nodes c_j represent the even parity constraints on their neighboring variable nodes, i.e.,

$$\bigoplus_{i:v_i \in n(c_j)} v_i = 0 \tag{2.12}$$

where $n(c_j)$ represents the set of all variable nodes connected to c_j and $\oplus$ shows the modulo-two sum.

Considering n variable nodes and r check nodes gives rise to a binary linear code of block length n, dimension $k \geq n - r$ and an $r \times n$ parity-check matrix $\mathbf{H}$. In other words, the (j, i) entry of H, h_{ji}, is 1 if and only if the $j$the check node c_j is connected to the $i$the variable node v_i. The dimension of the code is equal to $n - r$ if and only if all the parity constraints are linearly independent which is equivalent to $\mathbf{H}$ being full rank.

Any linear block code can be represented by a factor graph. In the case of LDPC codes, the factor graph is a sparse graph whose number of edges, E, grows linearly with the number of variable nodes n. LDPC codes can be simply extended to $GF(q)$, however, in this work, we focus on the binary LDPC codes.

Now, let us look at how a binary linear code or a specifical LDPC code can be

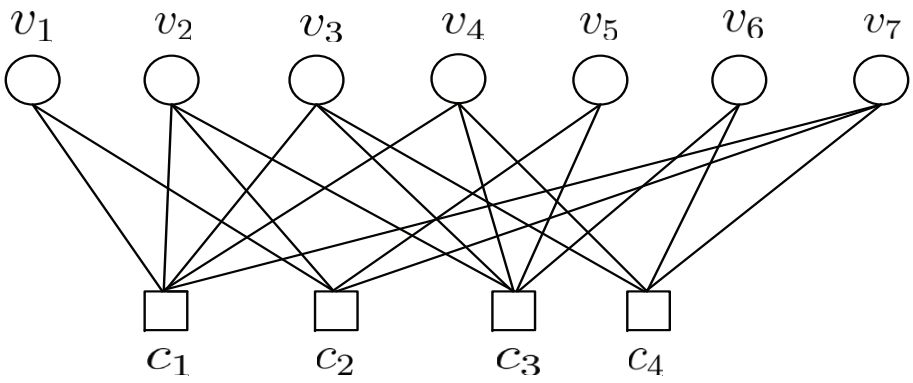

Figure 2.4: A factor graph representing an LDPC code. The graph has 7 variable nodes and 4 check nodes

represented by factor graphs. Consider a bipartite graph $\mathcal{G}$ with n variable nodes and r check nodes and E edges. We show variable nodes with circles and check nodes with squares, a simple graph having seven variable nodes and four check nodes is depicted in Fig 2.4. In this case, the four parity check equations given by four check nodes can be written as:

$$c_1 : v_1 \oplus v_2 \oplus v_3 \oplus v_4 \oplus v_7 = 0 \tag{2.13}$$

$$c_2 : v_1 \oplus v_2 \oplus v_5 \oplus v_7 = 0 \tag{2.14}$$

$$c_3 : v_2 \oplus v_3 \oplus v_4 \oplus v_5 \oplus v_6 = 0 \tag{2.15}$$

$$c_4 : v_3 \oplus v_4 \oplus v_6 \oplus v_7 = 0. \tag{2.16}$$

LDCP codes can be classified as *regular* or *irregular* according to their different structures. An LDPC code is called regular if all the variable nodes and check nodes have the fixed equal degree d_v and d_c, respectively. It means that the numbers of 1's in each row and column of $\mathbf{H}$ are constant. For a regular code, it follows that

$$E = d_v \cdot n = d_c \cdot r. \tag{2.17}$$

Therefore, assuming H is full rank, the code rate R can be computed as

$$R = 1 - \frac{d_v}{d_c} \tag{2.18}$$

Otherwise, $R > 1 - \frac{d_v}{d_c}$ [17].

Usually, an ensemble of LDPC codes having variable nodes of degree d_v and check nodes of degree d_c is represented as (d_v, d_c)-regular LDPC codes. For example, a (3,5) regular-LDPC code refers to a code with variable nodes of degree 3 and check nodes of degree 5. According to the rate equation (2.18), the rate of this code is $2/5$ bit per channel use (bpcu).

Irregular LDPC codes were first considered in [20] and it was shown that using irregular graphs can largely improve the performance of LDPC codes. In an irregular LDPC code, the numbers of 1's in each row or column of $\mathbf{H}$ are not constant. By careful design, irregular codes can be found which perform quite close to the channel capacity. For example, on the additive Gaussian noise channel (AWGN) channel, irregular LDPC codes can be designed to perform a few hundredths of a dB way from the Shannon limit [21].

For irregular LDPC codes, the variable and check nodes are usually defined by two edge degree distributions, $\{\lambda_2, \lambda_3, \ldots, \lambda_{d_v}\}$ and $\{\rho_1, \rho_2, \ldots, \rho_{d_c}\}$. In this notation, λ_i denotes the fraction of edges incident on variable nodes of degree i, ρ_j denotes the fraction of edges incident on check nodes of degree j and d_v, d_c denote the maximum degree of variable and check nodes, respectively. Thus, the edge degree distributions obey the constrains $\sum_i \lambda_i = 1$ and $\sum_i \rho_i = 1$. In polynomial form, they can be denoted as $\lambda(x) = \sum_{i=2}^{d_v} \lambda_i x^{i-1}$ and $\rho(x) = \sum_{j=2}^{d_c} \rho_j x^{j-1}$. For example, the degree distribution of the LDPC code shown in Fig. 2.4 can be represented as $\lambda(x) = \frac{1}{3}x + \frac{2}{3}x^2$ and $\rho(x) = \frac{4}{9}x^3 + \frac{5}{9}x^4$. Usually, an irregular LDPC code can be represented as $\mathcal{C}^n(\lambda(x), \rho(x))$, where n is the code length. In this thesis, we mainly use the polynomial notation which is more convenient.

Given $\lambda(x)$ and $\rho(x)$ of an irregular code and its number of edges E, we can have the number of variable nodes n

$$n = E \sum_{i=2}^{d_v} \frac{\lambda_i}{i} = E \int_0^1 \lambda(x) dx, \tag{2.19}$$

and the number of the check nodes r

$$r = E \sum_{j=2}^{d_c} \frac{\rho_j}{j} = E \int_0^1 \rho(x) dx. \tag{2.20}$$

Thus, ignoring the possibility of linearly dependent rows in H, the rate of the irregular code is given by

$$R = 1 - \frac{\sum_{j=2}^{d_c} \frac{\rho_j}{j}}{\sum_{i=2}^{d_v} \frac{\lambda_i}{i}} = 1 - \frac{\int_0^1 \rho(x)dx}{\int_0^1 \lambda(x)dx}. \tag{2.21}$$

Example 2.2 *consider an irregular LDPC code $\mathcal{C}^{1100}(\lambda(x), \rho(x))$ with variable node degree distribution $\lambda(x) = 0.3x + 0.3x^3 + 0.4x^7$ and check node degree distribution $\rho(x) = x^7$. Using (2.19) and (2.20), there are 4000 edges. According to (2.21), the rate of this code is $5/11$ bpcu.*

2.3 LDPC Codes: Decoding

LDPC codes are usually decoded by a class of iterative algorithms called *message-passing algorithms*. Since the *messages* passed in these algorithms are probabilities or beliefs, these algorithms are also called belief propagation algorithms. A message passing algorithm is an iterative decoding algorithm where there are two sources of information about the transmitted codewords available at each iteration: information from the channel and information from previous iteration. At each iteration, the decoder combines these two sources of information following some predefined rules to gain better knowledge about the the transmitted codewords.

To further illustrate the idea of iterative decoding algorithms for LDPC codes, consider the updated message from a variable node v of degree d_v to a check node in the decoder. In each iteration, this message is computed from $d_v - 1$ incoming message and the channel message to v. In fact, these $d_v - 1$ incoming messages are the outgoing messages of some check nodes which are updated previously. Consider one of these check node c with degree d_c, then the outgoing message is calculated based the the incoming $d_c - 1$ messages. This process can be seen as a decoding tree of depth one. Decoding trees of any depths can be obtained by continuing in the same fashion. Fig.2.5 shows an example of a depth-two decoding tree for an irregular LDPC code, which represents two message passing iterations.

When a factor graph is cycle-free (there is at most one path between every pair of nodes in the graph), the messages in the decoding tree of any depth are indepen-

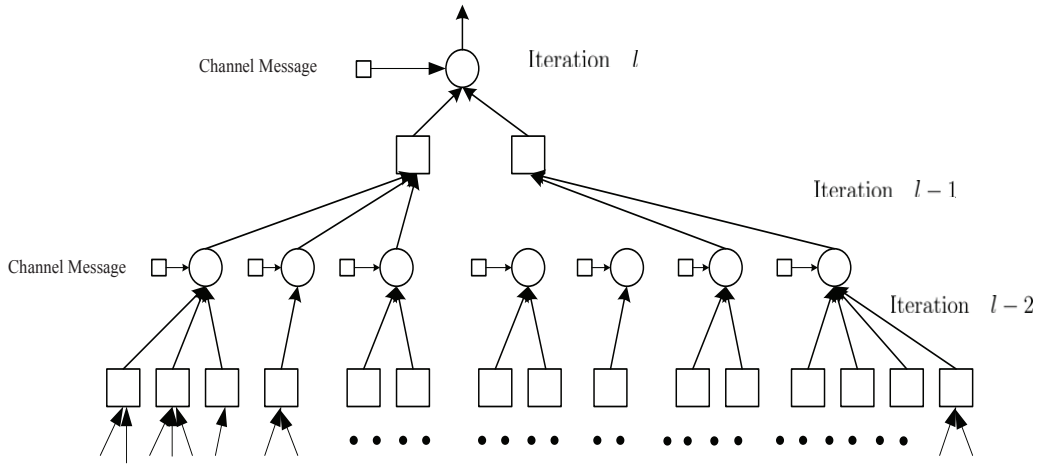

Figure 2.5: The decoding tree of depth two of an irregular LDPC code.

dent. If the factor graph has cycles with the smallest length (called girth) l, then up to depth $\lfloor \frac{l}{2} \rfloor$ the messages in the decoding tree are independent. Thus the independence assumption is valid only up to $\lfloor \frac{l}{2} \rfloor$ iteration and cycles make the message dependent for further iteration.

There are many different message-passing algorithms for LDPC codes. In this section, we will introduce some of them. We start with *Sum-Product* algorithm [22], which is the most powerful iterative message-passing decoding algorithm. Then, we present the basic idea of other algorithms.

2.3.1 The Sum-Product Algorithm

In this section, we describe the sum-product algorithm and also its message passing and updating rule to further interpretation.

When the variable nodes only have binary values, the probability messages passed along the edges have only two values, $P(0)$ or $P(1)$ with $P(0) + P(1) = 1$. Here, $P(x), x \in \{0, 1\}$, denote the probability of x. Thus, passing only one of $P(0)$ or $P(1)$ is enough for passing the message. However, it is usually more advantageous to use LLR instead of probabilities. For binary-valued random variable, LLR is defined as

$$\text{LLR} = \log \frac{P(0)}{P(1)}.$$

In fact, the sign of LLR shows the hard estimation while its magnitude reflect the reliability of decision. The reason for using LLR is that in computer implementations, the probability values that are very close to zero or very close to one can be represented without causing a precision error.

For the lth iteration of the sum-product decoding, all incoming LLR are processed by check nodes and then the resulting messages, denoted by $m^l_{c\to v}$, are sent back to the variable nodes. These messages are processed by variable nodes and then sent back to check nodes, these messages are denoted by $m^l_{v\to c}$. As there is no message from the check nodes at iteration $l = 0$, the variable nodes are initialized by the message calculated based on the channel output values, m_{0v}, which can be found by

$$m_{0v} = \log \frac{P(x = 0|y)}{P(x = 1|y)}, \tag{2.22}$$

where $x \in \{0, 1\}$ is the channel input bit and y is the channel output. And the message at the check node, $m^{(0)}_{c_j\to v}$ is initialized as 0.

The iterative process of sum-product algorithm can be described by two iterative updating rules. The updating rule at a parity-check node c is

$$m^l_{c\to v} = 2\tanh^{-1}\left(\prod_{v_i\in n(c)-v} \tanh(\frac{m^{(l-1)}_{c\to v_i}}{2})\right), \tag{2.23}$$

where $m_{a\to b}$ shows the message sent from node a to node b and $n(a)$ is the set of neighboring nodes connected to node a. In fact, to simplify the notation, we usually use CHK to denote the check nodes update rule, where $\text{CHK}(m_1, m_2) = 2\tanh^{-1}\left(\tanh(\frac{m_1}{2})\tanh(\frac{m_2}{2})\right)$. Thus, for a check node of degree d_c, the update rule can be also shown as

$$m^l_{c\to v} = \text{CHK}(m^{l-1}_{c\to v_1}, m^{l-1}_{c\to v_2}\cdots m^{l-1}_{c\to v_{d_c-1}}), v_i \in n(c) - v, i = 1, 2\cdots, d_c - 1.$$

The update rule at a variable node v is

$$m^l_{v\to c} = m_{0v} + \sum_{c_j\in n(v)-c} m^{(l)}_{c_j\to v}, \tag{2.24}$$

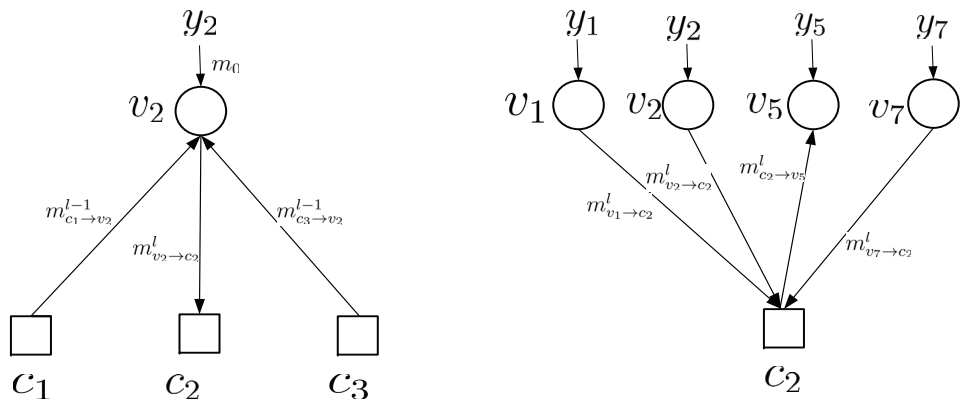

Figure 2.6: An example of message passing over factor graph of the code in Fig.2.4

Also, let us assume VAR$(m_1, m_2) = m_1 + m_2$, for a variable node of degree d_v the variable update rule can be simplified as

$$m^l_{v\to c} = \text{Var}(m_{0v}, m^l_{c_1\to v}, m^l_{c_2\to v}, \cdots, m^l_{c_{d_v-1}\to v}), c_i \in n(v) - c, i = 1, \cdots, d_v - 1.$$

To clarify these updating rules, a simple example is shown in Fig.2.6. At the lth iteration, the variable v_2 passes a message to c_2, denoted as $m^l_{v_2\to c_2}$, and then c_2 passes the calculated message to v_5, denoted as $m^l_{c_2\to v_5}$. This process can be done based on the following procedure. At the first half iteration, v_2 calculates its outgoing message based on the message it received from c_1 and c_3 according to (2.24) and send it to c_2. In the next iteration, c_2 calculates its outgoing message based on the messages received from v_1, v_2, v_7 according to (2.23) and send it to v_5.

At the $l$the iteration, a decision for a variable node v can be made based on the following decision rule:

$$V = \begin{cases} 0 & \text{if } m_{0v} + \sum_{c_j\in n(v)-c} m^{(l)}_{c_j\to v} > 0 \\ 1 & \text{if } m_{0v} + \sum_{c_j\in n(v)-c} m^{(l)}_{c_j\to v} < 0 \end{cases} \tag{2.25}$$

If $m_{0v} + \sum_{c_j\in n(v)-c} m^{(l)}_{c_j\to v} = 0$, the decision can be randomly chosen between 0 and 1 with equal probability.

One key advantage of the sum-product algorithm is that its decoding complexity grows linearly with the block length n because the number of edges is linearly increased with n. Hence increasing the block length for reducing the gap to capacity does not increase complexity per information bit.

When the code's factor graph is a tree, i.e. cycle free, the sum-product algorithm is optimal [4, 19, 23]. However, when the factor graph has cycles, sum-product becomes sub-optimal [4, 19, 23]. However, since with large block length the girth of the cycles grows and the graph becomes tree-like, the sum-product algorithm is still used in the decoding of LDPC codes, and it is also observed that even for moderate block length (a few hundred bits), the performance of the sum-product algorithm is fairly good [19].

2.3.2 The Min-Sum Algorithm

Min-sum decoding is the second most attractive decoding algorithm for LDPC codes due to its less complexity and easier implementation. Since min-sum is a simplified version of the sum-product algorithm, it is not as effective and powerful as the sum-product algorithm.

In the min-sum algorithm, the update rule at the variable nodes is the same as sum-product algorithm (2.24), but the update rule at the check node is an approximation of (2.23). As we observe that for $x \gg 1$, $\ln(\cosh(x)) \approx |x| - \ln 2$. An approximation for CHK can be obtained as

$$\begin{aligned} \text{CHK}(m_1, m_2) &\approx |(m_1 + m_2)/2| + |(m_1 - m_2)/2| \\ &= \text{sign}(m_1)\text{sign}(m_2)\min(|m_1|, |m_2|). \end{aligned} \tag{2.26}$$

Applying this approximation in (2.23), the check node update rule for min-sum algorithm is presented as

$$m^l_{c\to v} = \min_{v_i \in n(c)-\{v\}} (|m^{(l-1)}_{v_i \to c}|) \prod_{v_i \in n(c)-\{v\}} \text{sign}(m^{(l-1)}_{v_i \to c}) \tag{2.27}$$

This approximation becomes more accurate when the magnitude of the message is increased. Therefore, in the later iterations, when the magnitude of the messages usually have become large, the performance of this algorithm is almost the same as that of sum-product algorithm. Also, there are other message-passing algorithms such as Gallager's algorithm A and Gallager's algorithm B [5, 24] which are not the focus of this thesis.

2.4 LDPC Codes: Analysis

In order to analyze the performance of an ensemble of LDPC codes, we need to statistically analyze the message passing decoder. In this section, we introduce some analysis methods which are used in the future chapters. Let us first present some basic background.

In this part, we consider an ensemble of LDPC codes whose rate is less than the capacity of the channel. As mentioned in the previous section, we have two sources of information available in the decoder in iterative decoding. One is called intrinsic information which is observed from the channel and the other one is called extrinsic information which is from the previous iterations. Based on these two sources of information, extrinsic information for the next iteration is calculated. In a successful decoding, the reliability of extrinsic messages gets better and better as the decoding continues iteration by iteration. Thus, for analyzing of iterative decoders, the statistics of the extrinsic messages are studied at each iteration.

Check Node Symmetry

Consider a check node c of degree d_c, with input LLR messages $m_1, m_2, \cdots, m_{d_c-1}$, the check node update rule is symmetric if

$$\text{CHK}(b_1 m_1, \cdots, b_{d_c-1} m_{d_c-1}) = \text{CHK}(m_1, m_2, \cdots, m_{d_c-1})(\prod_{i=1}^{d_c-1} b_i),$$

where $b_i \in \pm 1, i = 1, 2, \cdots, d_c - 1$ and CHK denotes the check node update rule, which generates an output message based on the input d_c messages.

Variable node symmetry

Consider a degree-d_v variable node v, with input LLR messages $m_0, m_1, \cdots, m_{d_v-1}$, the variable node update rule is symmetric if

$$\text{VAR}(-m_0, -m_1, \cdots, -m_{d_c-1}) = -\text{VAR}(m_0, m_1, \cdots, m_{d_c-1}),$$

where VAR denotes the variable node update rule, which generates one output message based on the input d_v messages which includes the channel message m_0.

It is shown in [25] that over a BIMS, if the update rules at the check nodes and variable nodes are symmetric, then the performance of this decoder is independent of the transmitted codeword. Therefore, in this work, we assume that all-zero codeword is transmitted. Under this assumption, the error rate can be defined based on the percentage of messages carrying a belief for '1'.

It is worth mentioning that when the block length is large, the effect of cycles in the decoding performance is small and vanish asymptotically. Therefore, the random messages at the input of variable nodes and check nodes become independent. In this work, we always assume the code length is large enough such that the factor graph is a tree and all the messages are independent.

2.4.1 Density Evolution

Density evolution, first presented by Richardson and Urbanke in [25], is a general asymptotic analysis method for the message passing decoders. It can also be used for other codes defined on graphs associated with iterative decoding [26, 27]. It tracks the evolution of the pdf of the extrinsic message at each iteration.

The analytical formulation of this technique can be found in [25]. Density evolution is computationally complex thus not suitable for direct use. For practical use, Chung proposed a quantized version of density evolution called *discrete density evolution* which quantizes the message alphabet and uses probability mass functions (pmf) instead of pdfs in order to make a computer implementation possible. Discrete density evolution is a powerful tool for code design and performance analysis [21]. In the rest of this work, when we refer to density evolution, we mean discrete density evolution. Next, we present some details of this technique for sum-product decoding.

Before we go to the details, let us first introduce the quantizing function used in the density evolution, denoted as $\mathcal{Q}(x)$. This function can be represented as

$$\mathcal{Q}(x) = \begin{cases} \lfloor \frac{x}{\triangle} + \frac{1}{2} \rfloor \cdot \triangle & \text{if } x \geq \frac{\triangle}{2}, \\ \lfloor \frac{x}{\triangle} - \frac{1}{2} \rfloor \cdot \triangle & \text{if } x \leq -\frac{\triangle}{2}, \\ 0 \quad , & \text{otherwise} \end{cases} \tag{2.28}$$

where $\triangle$ is the quantization interval.

At the first iteration, the variable nodes are initialized with the pmf of the channel messages. Then, these messages are sent to the check nodes. And the check nodes calculate the output pmf based on the incoming pmfs following the check nodes update rule (2.23). For a check node with two input pmfs p_a and p_b, the output pmf is given by

$$p_{out}[k] = \sum_{(i,j):k\triangle=R(i\triangle,j\triangle)} p_a[i]p_b[j], \tag{2.29}$$

where

$$R(a,b) = \mathcal{Q}(\text{CHK}(a,b)).$$

These equations can be implemented using a look-up table. For a variable node with two incoming pmfs p_a and p_b, the output pmf is given by

$$p_{out}[k] = p_a[k] \bigotimes p_b[k], \tag{2.30}$$

where $\bigotimes$ denotes the discrete convolution which can be easily done by using fast-Fourier transform (FFT) techniques. To simplify the notation, let us denote the pmf update operations for a check node and a variable node on two incoming pmfs p_a and p_b as $\mathcal{CHK}(p_a, p_b)$ and $\mathcal{VAR}(p_a, p_b)$, respectively.

Also, notice that the update rules discussed in Section 2.3.1 can be written as

$$\begin{aligned} &\text{CHK}(m_1, m_2, \cdots, m_{d_c-1}) = \text{CHK}(m_1, \text{CHK}(m_2, \cdots, m_{d_c-1})) \\ &\text{VAR}(m_0, m_1, \cdots, m_{d_v-1}) = \text{VAR}(m_0, \text{VAR}(m_1, \cdots, m_{d_v-1})). \end{aligned} \tag{2.31}$$

This implies that the check nodes and variable nodes operations can be done pairwise. Assuming the variable node message pmf is p_v (which is the same for all the variables) and the check nodes message pmf is p_c, thus p_c and p_v can be calculated by

$$\begin{aligned} p_c &= \mathcal{CHK}(p_v, \mathcal{CHK}(p_v, \cdots, \mathcal{CHK}(p_v, p_v)). \\ p_v &= \mathcal{VAR}(p_0, \mathcal{VAR}(p_c, \cdots, \mathcal{VAR}(p_c, p_c)). \end{aligned} \tag{2.32}$$

where p_o is the intrinsic message pmf. At the lth iteration, we also use

$$p_v^l = p_0 \bigotimes \lambda(\rho(p_v^{l-1}))$$

as the shorthand for the variable node update rule.

If the all-zeros codeword is transmitted and BPSK is used, in each iteration the probability of error of the messages can be represented as the negative tail of

the pmf. Thus, the decoding is successful if this negative tail vanishes after some iterations, i.e.,

$$\lim_{l\to\infty} p_v^l = \delta_\infty.$$

Due to high complexity of the exact density evolution, there have been some approximations in the literature. The most important ones are Gaussian approximations proposed by Chung *et al.* [28] and semi-Gaussian approximation proposed by Ardakani *et al.* [29]. In the Gaussian approximation, Chung *et al.* assumed all the extrinsic messages have Gaussian distributions. In [29], Ardakani *et al.* consider that only variable node messages are Gaussian. These methods have considerably lower complexity than the density evolution with some penalty in the accuracy.

Density evolution is a powerful tool for analyzing the iterative decoding under the sum-product algorithm. It is worth mentioning that other iterative decoding algorithms (e.g., the min-sum algorithm) and other codes defined on graphs which use iterative decoding can also use density evolution as a powerful analysis tool.

2.4.2 Decoding Threshold of An LDPC Code

The decoding threshold of an LDPC code, first introduced by Richardson and Urbanke [25], is defined as the worst channel condition for which the message error probability converge to zero as the number of iteration goes to infinity. In other words, when the channel is better than the decoding threshold, density evolution can converge to an arbitrary small message error rate. When the channel is worse than the decoding threshold, the error probability remains larger than a constant even if the number of iterations goes to infinity. If the threshold of a code is equal to the Shannon limit, this code is said to be a capacity-achieving code.

The decoding threshold is one of the most important properties of an LDPC code. It depends on many characters, such as degree distributions, the decoding algorithm used, etc. For example, the threshold of the (3,6)-regular code on AWGN channel under sum-product decoding algorithm is 1.1015 dB, while under min-sum decoding algorithm it is 1.6990 dB. This means that for successful convergence under the sum-product decoding, the channel signal-to-noise ration (SNR) must be

better than 1.1015 dB and for the min-sum decoding better than 1.6990 dB. It is worth mentioning that we usually use finite-length codes in practice which would demonstrate a gap to the asymptotic performance and this gap increases as the code length decreases.

2.4.3 Extrinsic Information Transfer Chart Analysis

A fast and efficient approach for analyzing iterative decoders is to use an extrinsic information transfer (EXIT) chart. The idea of this method, which was first introduced by Ten Brink [30], is to track the evolution of a single parameter iteration by iteration. In other words, this parameter can be seen as a measure of the decoder' success. In fact, many parameters can be chosen to present the decoder's evolution. For example, one can track the SNR of the extrinsic messages [31, 32], the mutual information between the transmitted bits and the extrinsic messages in each iteration [30], or the extrinsic message error probability [29, 33], etc. To help the understanding, we consider here an EXIT chart based on tracking the message error rate.

In the EXIT chart method, we express the message error probability at the output of one iteration $p_{e,out}$ in terms of the message error probability at the input of the iteration $p_{e,in}$ and the intrinsic messages p_0,i.e.,

$$p_{e,out} = f(p_{e,in}, p_0). \tag{2.33}$$

For a fixed p_0, the EXIT char can be plotted using $(p_{e,in}, p_{e,out})$ coordinates. Usually, EXIT charts are presented by plotting both f and its inverse f^{-1} to better visualize the behavior of the decoder in each iteration. A sample EXIT chart is plotted in Fig.2.7. Each arrow in this figure represents on iteration of decoding. If the tunnel of an EXIT chart is wide, only a few number of iterations is needed for convergence. When the tunnel is open, we say the EXIT chart is open. Otherwise, if the EXIT chart gets closed such as for some $p_{e,in}$, it happens that $p_{e,out} > p_{e,in}$, the error probability cannot get smaller than a certain value and the convergence cannot be achieved. An open EXIT chart requires that the curve be always below

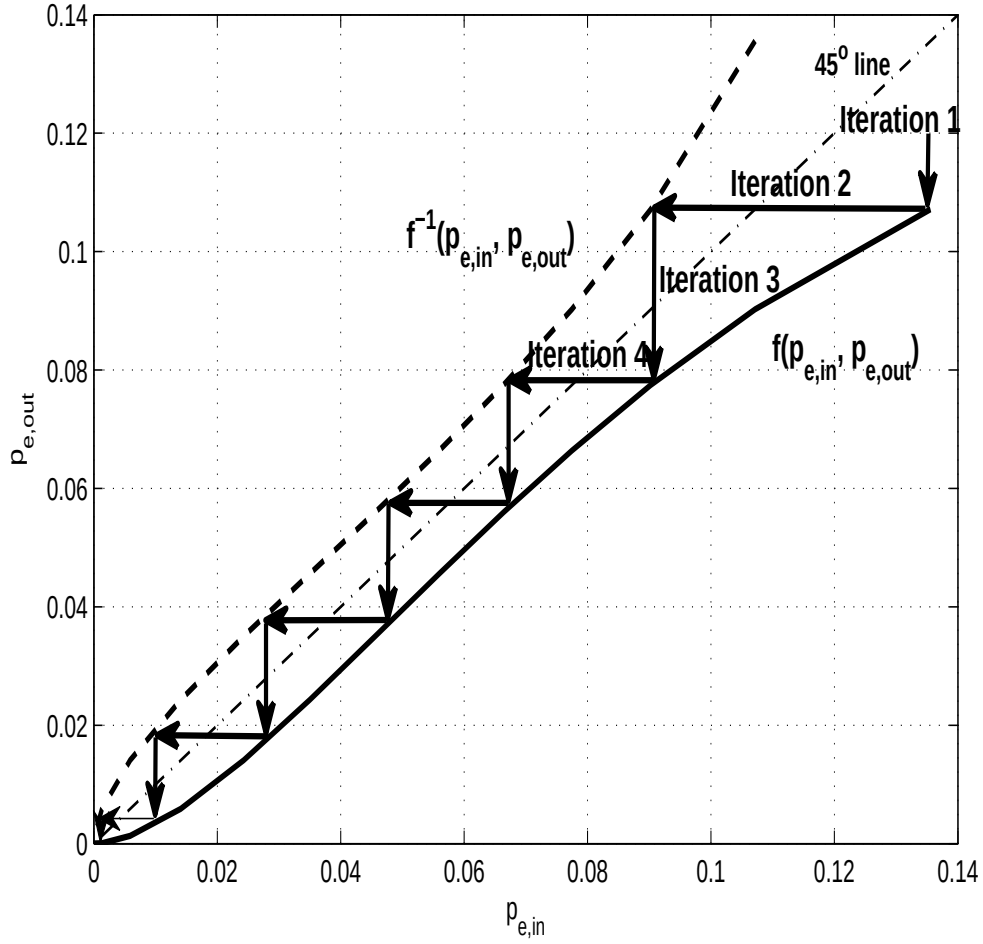

Figure 2.7: An EXIT chart of a (3,5)-regular LDPC codes on the BIAWGN under the sum-product algorithm. This EXIT chart is based on message error rate.

the 45-degree line. The decoding threshold $p_0^\star$, which is the worst channel condition for an open EXIT chart, can be written as

$$p_0^\star = \arg\sup_{p_0} f(p_{e,in}, p_0) < p_{e,in}, \quad \text{for all} \quad 0 < p_{e,in} \leq p_0. \tag{2.34}$$

Since an EXIT chart tracks just the evolution of a single parameter, it is not as accurate as density evolution in general. However, when the pdf of the extrinsic messages can be fully represented by a single parameter, such as in the case of decoding on the BEC or for hard decoding algorithms, EXIT charts are exact and equivalent to density evolution. Due to their simplicity, EXIT charts are commonly used for LDPC code design [34]. In the next section, we will briefly discuss how EXIT charts can help us to find good LDPC codes.

2.5 LDPC Codes: Design Methods

In this thesis, an ensemble of LDPC codes is represented by its check node and variable node degree distributions, $\rho(x) = \sum_{i=2}^{d_c} \rho_i x^{i-1}$ and $\lambda(x) = \sum_{i=2}^{d_v} \lambda_i x^{i-1}$, respectively. Thus, designing a good code means finding degree distributions which have a desired performance under some constrains. The LDPC code design is usually done by numerical optimization method.

There are mainly two methods to design an LDPC code. One is for a given minimum rate, we seek a code which has the highest decoding threshold. Generally, this approach is complex and is usually done by search based methods. Details can be found in [15]. The other method is based on seeking the code who has the highest rate for a given threshold.

When we are designing an LDPC code, an analysis method is also needed to measure the performance. We use EXIT chart to design the code because the optimization problem using this approach can be formulated as a linear program which can be solved efficiently. In the design method of this work, we fix the check degree distribution and optimize the variable node degree distribution due to the result reported in [15, 21].

In the EXIT chart method, we replace the output pdf by a pdf from a family

of pdfs which has the same entropy at each iteration of density evolution. Given a channel with capacity $\mathcal{C}$, the channel entropy is $H_0 = 1 - \mathcal{C}$. Now denoting the output entropy of the previous iteration by h, we can define the output entropy of a degree-i variable node as $f_i(H_0, h)$, which is also called the *elementary EXIT chart* for degree-i variable nodes. Thus, the condition for successful decoding is that

$$\sum_i \lambda_i f_i(H_0, h) < h, \quad \text{for all} \quad 0 \leq h \leq H_0. \tag{2.35}$$

This is a linear constraint on the design parameters $\{\lambda_i, i \geq 2\}$. Given $\rho(x)$ is fixed, according to (2.21), it is sufficient to maximize $\sum_i \frac{\lambda_i}{i}$ to obtain the maximum code rate. Thus, the process of finding the highest rate for a given BMC can be presented as the following linear programming problem:

$$\begin{aligned} &\text{maximize} \sum_i \frac{\lambda_i}{i} \\ &\text{subject to} \lambda_i \geq 0 \\ &\sum_i \lambda_i = 1 \\ &\sum_i \lambda_i f_i(H_0, h) < h, \forall h \in [0, H_0], \end{aligned} \tag{2.36}$$

The elementary EXIT chart can be found by density evolution and the EXIT chart of a code is a linear combination of the elementary EXIT charts.

Although the performance of LDPC codes is not too sensitive to $\rho(x)$, further optimization on $\rho(x)$ will get better results. Suggestions and guidelines can be found in [15]. During this work, we use this method to obtain close-to-capacity LDPC ensembles.

Chapter 3

Low Complexity LLR Calculation

3.1 Introduction

There have been many advances for the iterative decoding algorithms of graphical codes. Error-correcting codes, such as LDPC codes [5] and turbo codes [2], associated with iterative decoding are known to approach the Shannon limit on many channels [21,35,36]. These codes have been also proposed for MIMO channels and relay channels [37].

For soft iterative decoding, the messages calculated in each iteration are usually LLR. LLR are shown to be efficient metrics in the decoding and analyzing the performance of binary graphical codes [15, 19]. The channel LLR usually depends on the channel output, noise variance, fading characteristics, and also the availability of the channel state information (CSI) at the receiver. The capacity of the channel is also affected with the availability of channel parameters at the receiver. Under a realistic wireless channel, the relationship between LLR and the channel output can be quite complex. For example, under MIMO channels, LLRs are usually non-linear complicated functions of the channel output even when CSI is perfectly known. As a result, it may be time and energy consuming for the decoder to calculate exact LLRs. Consequently, accurate approximate LLRs should be used to implement efficient decoders.

Calculation of approximate LLRs has been considered in the recent literature, e.g, see [11–14]. For example, the expected value of channel gain was used as

an estimation parameter in [20, 36], but this choice cannot guarantee the optimum performance in the decoder. Another common approximation method is the max-log approximation. The max-log approximation is usually accurate at high SNRs, but modern error-correcting codes such as LDPC codes and turbo codes usually operate at low SNRs [11]. Moreover, when there is no CSI at the receiver, the max-log approximation is still quite complex. Thus, alternative methods, which are also accurate at low SNRs, are much needed.

Piecewise linear LLR calculation has been considered in [11] for soft Viterbi decoding of convolutional codes in the HIPERLAN/2 standard. In [13], approximating LLRs by linear functions has been proposed for BPSK over symmetric channels and a measure of LLR accuracy has also been introduced. Using this measure, linear LLRs have been designed with almost the same performance as true LLRs. Later, this measure has been generalized to binary asymmetric channels in [14] and applied to non-binary modulations under BICM [38]. Numerical results have also shown that the performance loss of their optimized piecewise linear LLRs based on the generalized measure is quite small.

The linear approximate methods considered above are all focused on SISO channels. MIMO technology plays an important role in modern wireless communication due to its higher spectral efficiency (more bits per second per hertz of bandwidth) and link reliability [17]. The problem of finding a good approximation method for MIMO channels when CSI is known at the receiver is studied in this chapter. We find a piecewise linear approximation which allows for the maximum achievable rate on the MIMO channels. We also show that the approximation method closely approaches the capacity under true LLR calculation.

The two-way relay communication channel has gained significant interest recently due to its potential application in modern cellular network. One advantage of two-way relay channel is that it has higher spectral efficiency than the one-way relay channel [39, 40]. In this chapter, when no CSI is available at the receiver, we find a piecewise linear approximation for the two-way relay channel. Numerical results confirm that the proposed method outperforms the max-log approximation

and performs closely to the true LLRs.

The rest of this chapter is organized as follows. Section 3.2 reviews some preliminaries and studies the proposed approaches. Section 3.3 studies the approximation method for the MIMO-BICM channel. Approximation method for two-way relay channel when there is no CSI is studied in Section 3.4. Section 3.5 concludes the chapter.

3.2 Preliminaries and Approaches

3.2.1 Important Approximation Methods

In this section, we review some approximation methods and background knowledge which will be used in this chapter.

Max-log Approximation

Consider a sequence of real number $z_1, z_2, \cdots, z_n$, the method called max-log approximation is described in (3.1) and is accurate when there is a dominant term in the sequence.

$$\log \sum_k z_k \approx \max_k \log z_k. \tag{3.1}$$

Minimum Mean Square Error Approximation (MMSE)

Consider a random variable $r \geq 0$ with arbitrary pdf and its estimator $\hat{r}$, its MMSE is given by

$$\hat{r}_r = arg \min_{\hat{r}} E(|r - \hat{r}|^2). \tag{3.2}$$

Where $E(\cdot)$ denotes the expectation. This estimation has a simple solution which is $\hat{r} = E(r)$. This approximation has been widely used in the literature [20, 36].

3.2.2 Capacity

As we discussed in Chapter 1, the capacity of a BIMS channel can be written via the pdf $f(l)$ of its LLRs. That is

$$\mathcal{C} = 1 - \int_{-\infty}^{+\infty} (1 - \log_2(1 + \exp^{-l}))dx. \tag{3.3}$$

This equation is only valid for the BIMS where its LLR is symmetric.

When the channel is asymmetric, we cannot use (3.3). In such cases, if the input is equally likely 0 or 1, the capacity of the channel is given by [41]:

$$\mathcal{C} = 1 - \frac{1}{2}\int_{-\infty}^{+\infty} \log_2(1+e^{-l})p^0(l)dl - \frac{1}{2}\int_{-\infty}^{+\infty} \log_2(1+e^{l})p^1(l)dl \qquad (3.4)$$

where $p^b(l)$ denote the LLR pdf conditioned on the $b \in \{0,1\}$.

3.3 Low Complexity Linear LLR Calculation for MIMO-BICM Channel

MIMO technology has attracted much attention in the past few decades since it can offer high spectral efficiency and reliable wireless communication over a multi-path environment [42]. BICM is a pragmatic technique which can achieve large diversity orders in fading wireless channels [38]. The combination of these two technologies, MIMO-BICM, has also received attention recently, because of its lower detection complexity and near-optimal performance [43–45]. In this section, we study the piece-wise linear approximation for MIMO-BICM channels.

3.3.1 Background

Multiple Antenna Channel Model

Wireless communication experiences multi-path propagation as the signal is reflected by the nearby surfaces on the way to the receiver. As multi-path propagation causes dispersions in delay, frequency and the spatial domains, each antenna receives transmitted signals with different attenuation, phase or propagation delay.

If the separations of antennas are sufficiently large, we can assume the received signals of different antennas are independent. Each equivalent path between each transmit and receive antenna experiences a complex Gaussian noise with zero mean and unit variance, denoted as $\mathcal{CN}(0,1)$. This channel model is depicted in Fig. 3.1.

For the fading channel, there are different classifications based on different rules. For example, based on speed of change in the magnitude and phase of the channel gain, a fading channel can be classified as slow fading or fast fading

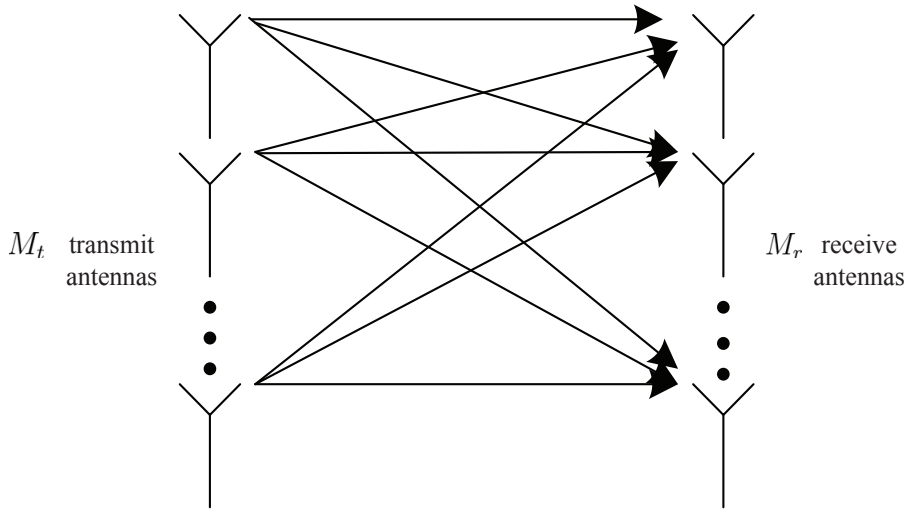

Figure 3.1: MIMO channel model

channel. From the perspective of frequency, a fading channel is classified as flat fading or frequency-selective fading. In this work, we only consider the slow, flat fading channel which means that the amplitude and phase of the channel gain can be seen roughly constant over the period of use and all frequency components of the signal will experience the same magnitude of fading [42].

MIMO-BICM System Model

The block diagram of a MIMO-BICM system is depicted in Fig.3.2. Here, a flat slow-fading MIMO channel with M_{t} transmitting antennas and M_{r} receiving antennas is considered.

During the transmission, a sequence of binary information bits generated by the binary source is first encoded by an error-correcting encoder and then interleaved to get the sequence $\{c_i\}$. After being de-multiplexed into M_{t} antenna sequences, each group of m bits are mapped to data symbols, $x_i \in \mathcal{X}$, $i = 1, \ldots, M_{\mathrm{t}}$. Here $\mathcal{X}$ denotes the symbol alphabet of size $|\mathcal{X}| = 2^m$ and unit average power. As a result, at time index t the transmit vector, which carries $R_0 = m \times M_{\mathrm{t}}$ coded bits, is given by $\mathbf{x} = (x_1, x_2, \ldots, x_{M_{\mathrm{t}}})^T$ [12, 46].

At the receiver, the received vector is given by

$$\mathbf{y} = \sqrt{\gamma}\mathbf{H}\mathbf{x} + \mathbf{n}.$$

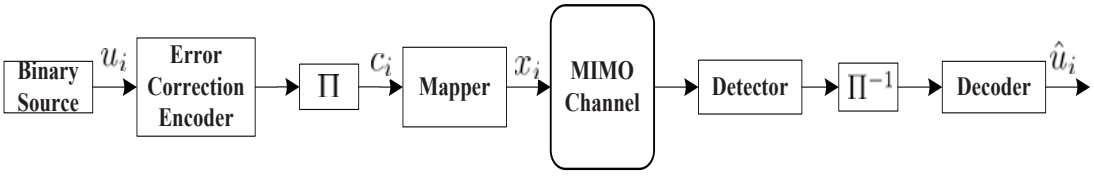

Figure 3.2: MIMO-BICM transmission model.

where $\mathbf{H}$ is an $M_\mathrm{r} \times M_\mathrm{t}$ channel matrix, the entries of which are all independent, and $\mathbf{n}$ is the additive white Gaussian noise vector, $(n_1, n_2 \ldots, n_{M_\mathrm{r}})^T$, each component of which is modeled as i.i.d. $\mathcal{CN}(0,1)$. Assuming $\mathbf{H}$ with normalized entries and unit average power, γ is the average SNR per transmit antenna.

At the receiver, the received signal $\mathbf{y}$ and the channel matrix $\mathbf{H}$ are used to calculate the LLR value for each bit c_i. Then, the sequence of LLR values is de-interleaved and passed to the decoder to obtain an estimation of the information bits $\hat{u}_i$.

In the case of ideal interleaving, the system can be equivalently seen as R_0 parallel independent and memoryless binary-input sub-channels [38]. As a result, the capacity of the MIMO-BICM is given by $\mathcal{C} = \sum_{i=1}^{R_0} \mathcal{C}_i$, where $\mathcal{C}_i$ is the capacity of each sub-channels.

LLR Calculation

Let $\mathcal{X}_i^b$ denote the sets of transmit vectors for which the i-th coded bit $c_i = b \in \{0,1\}$ ($i = 1, \ldots, R_0$). Assuming a uniform input distribution and that the channel matrix is known at the receiver, the true LLR for the i-th sub-channel is given by

$$l_i = \log \frac{P(c_i = 0|\mathbf{y}, \mathbf{H})}{P(c_i = 1|\mathbf{y}, \mathbf{H})} = \log \frac{\sum_{\mathbf{x} \in \mathcal{X}_i^0} p(\mathbf{y}|\mathbf{x}, \mathbf{H})}{\sum_{\mathbf{x} \in \mathcal{X}_i^1} p(\mathbf{y}|\mathbf{x}, \mathbf{H})}, \tag{3.5}$$

where the conditional pdf, $p(\mathbf{y}|\mathbf{x}, \mathbf{H})$, is given by

$$p(\mathbf{y}|\mathbf{x}, \mathbf{H}) = \frac{1}{(2\pi)^{M_\mathrm{r}}} \exp\left(-\frac{\|\mathbf{y} - \sqrt{\gamma}\mathbf{H}\mathbf{x}\|^2}{2}. \right) \tag{3.6}$$

According to (3.5), l_i is a function of $\mathbf{y}$ denoted as $f_i\ (\mathbf{y})$. Also, it is clear that the complexity of computation of true LLR values grows exponentially with the

number of transmit antennas. As we mentioned before, for practical purposes, an approximation method is desired.

Applying max-log approximation (3.1) to (3.5) results in

$$l_i \approx \frac{1}{2}\left(\min_{\mathbf{x}\in\mathcal{X}_i^1} \|\mathbf{y}-\sqrt{\gamma}\mathbf{H}\mathbf{x}\|^2 - \min_{\mathbf{x}\in\mathcal{X}_i^0} \|\mathbf{y}-\sqrt{\gamma}\mathbf{H}\mathbf{x}\|^2\right). \tag{3.7}$$

Obviously, this method reduces the complexity, but the search space still grows exponentially with M_{t}. Also, the max-log approximation is usually accurate when there is a dominant term in the sum, which usually happens at high SNRs. At low SNRs, therefore, other methods should be considered.

3.3.2 Problem Description and Proposed Method

In this section, we are looking for an efficient LLR approximation method over MIMO channels with perfect CSI at the receiver. In other words, we seek low-complexity approximate functions $\hat{l}_i = \hat{f}_i(\mathbf{y})$ which provide good performance. To optimize the parameters of the proposed approximate LLR calculating functions, we need an LLR accuracy measure.

LLR Accuracy Measure

Since the capacity of sub-channel i is given by (3.4), for a single-input single-output channel, it is shown in [14] that good approximate LLRs can be found by maximizing

$$\hat{R}_i = 1 - \frac{1}{2}\int_{-\infty}^{+\infty} \log_2(1+e^{-\hat{l}})\hat{p}_i^0(\hat{l})d\hat{l} - \frac{1}{2}\int_{-\infty}^{+\infty} \log_2(1+e^{\hat{l}})\hat{p}_i^1(\hat{l})d\hat{l},$$

where $\hat{p}_i^b(\hat{l})$ is the conditional pdf of approximate LLRs given $c_i = b$. In other words, approximate LLRs are found in the sense of maximizing the achievable rate on the SISO channel.

Similarly, for the MIMO case, we propose the following LLR accuracy mea-

sure:

$$\hat{R} = \sum_{i=1}^{R_0} \hat{R}_i = \sum_{i=1}^{R_0} \left(1 - \frac{1}{2} \int_{-\infty}^{+\infty} \log_2(1 + e^{-\hat{l}}) \hat{p}_i^0(\hat{l}) d\hat{l} - \frac{1}{2} \int_{-\infty}^{+\infty} \log_2(1 + e^{\hat{l}}) \hat{p}_i^1(\hat{l}) d\hat{l} \right). \tag{3.8}$$

Maximizing this measure is equivalent to maximizing each term of the sum, and thus maximizing the overall achievable rate on the MIMO channel. This measure can be seen as the generalization of the method proposed in [14] from the SISO case to the MIMO case.

Given the LLR accuracy measure of (3.8), the procedure of finding good LLR approximating functions is as follows. First, for each sub-channel i, a general LLR approximating function is defined as $\hat{l}_i = \hat{f}_i^{\mathcal{O}_i}(\mathbf{y})$, where $\mathcal{O}_i$ denotes the set of parameters. Next, at any SNR, the optimized parameters $\mathcal{O}_i$ of each bit-channel are found by solving the following optimization problem:

$$\mathcal{O}_i^{\text{opt}} = \arg \max_{\mathcal{O}_i} \hat{R}_i, \quad \text{subject to } \Psi_i(\mathcal{O}_i)=0. \tag{3.9}$$

Here, $\Psi_i(\mathcal{O}_i) = 0$ represents the constraints that may be imposed on $\mathcal{O}_i$, such as continuity conditions.

LLR Approximation Functions

To find the desired approximation functions, the first step is to choose the right class of approximation function. Clearly, many approximation functions can be applied to solve the optimization problem in (3.9). Here, piecewise linear functions are chosen because of their low implementation complexity.

For the MIMO channel of our case, $\mathbf{y} \in \mathbb{C}^{M_\text{r}}$ where $\mathbb{C}$ denotes the set of complex numbers. First, this M_r-dimension space is partitioned into N_i regions $\Omega_1, \ldots, \Omega_{N_i}$. The number of regions and their shapes are chosen based on the shape of true LLRs and the affordable computational complexity. Next, for each region, a linear function is proposed for the approximation. Thus, the proposed general

approximation function for the MIMO-BICM system is as follows:

$$\hat{l}_i = \hat{f}_i^{\mathcal{O}_i}(\mathbf{y}) = \sum_{k=1}^{N_i} \left(\boldsymbol{\alpha}_i^k \mathbf{y} + \beta_i^k\right) \mathbf{1}_{(\mathbf{y}\in\Omega_i)}. \tag{3.10}$$

Here, $\boldsymbol{\alpha}_i^k = (\alpha_i^{k,1}, \ldots, \alpha_i^{k,M_\mathrm{r}})$ are row vectors and β_i^k are real scalars. Also, $\mathbf{1}_{(\cdot)}$ denotes the indicator function.

Finally, $\mathcal{O}_i = \{\boldsymbol{\alpha}_i^1, \ldots, \boldsymbol{\alpha}_i^{N_i}, \beta_i^1, \ldots, \beta_i^{N_i}\}$ are the set of parameters needed to be optimized, and Ω_i's are chosen by search. It is worth mentioning that the symmetry of the true LLRs usually reduce the number of parameters. To better illustrate our method, two examples are provided in the following.

Example 3.1 (2×1 **MIMO Channels)** *In the case of* 2×1 *MIMO channels, the received signal is just a complex scalar* $y \in \mathbb{C}$*. Thus,* $\boldsymbol{\alpha}_i$ *is also a complex scalar. Then, (3.31) becomes:*

$$\hat{f}_i^{\mathcal{O}_i}(y) = \sum_{k=1}^{N_i} \left(\alpha_i^k y + \beta_i^k\right) \boldsymbol{1}_{(y\in\Omega_i)}, \tag{3.11}$$

where Ω_i*'s partition* $\mathbb{C}$ *into* N_i *regions by one-dimensional boundaries. Thus, the parameters needed to be optimized in this case are* $\mathcal{O}_i = \{\alpha_i^1, \ldots, \alpha_i^{N_i}, \beta_i^1, \ldots, \beta_i^{N_i}\}$ *and are optimized by (3.9).*

Example 3.2 (2×2 **MIMO Channels)** *In the case of multiple receive antennas, the true LLRs are functions of the multidimensional vector* $\mathbf{y}$*. For example, in the case of* 2×2 *MIMO channels, the received signal* $\mathbf{y} = (y_1, y_2)^T$ *and* $f_i^{\mathcal{O}_i}$ *are functions of a two-dimensional complex vector. Let* $\boldsymbol{\alpha}_i^k = [\alpha_i^{k,1}, \alpha_i^{k,2}]$*, then*

$$\hat{f}_i^{\mathcal{O}_i}(\mathbf{y}) = \sum_{k=1}^{N_i} \left([\alpha_i^{k,1}, \alpha_i^{k,2}] \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} + \beta_i^k \right) \boldsymbol{1}_{(\mathbf{y}\in\Omega_i)}, \tag{3.12}$$

where Ω_i *are segments of the* $\mathbb{C}^2$ *domain. Thus, the parameters here are* $\mathcal{O}_i = \{(\alpha_i^{1,1}, \alpha_i^{1,2}), \ldots, (\alpha_i^{N_i,1}, \alpha_i^{N_i,2}), \beta_i^1, \ldots, \beta_i^{N_i}\}$.

As mentioned earlier, after observing the shape of true LLRs, we decide N_i and the corresponding regions. For example at low SNR, where BPSK is commonly used,

observing the shape of true LLRs suggests that $N_i = 1$ is enough. In other words, we only have Ω_1 and in fact $\Omega_1 = \mathbb{C}$. This leads to linear approximation functions, thus minimum complexity. Increasing N_i provides slightly better performance at the cost of extra complexity. Thus, we have

$$\hat{f}_i^{\mathcal{O}_i}(\mathbf{y}) = [\alpha_i^1, \alpha_i^2] \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} + \beta_i. \tag{3.13}$$

The same process in (3.9) can be applied here to find these approximation functions.

3.3.3 Experimental Results

In this section, the performance of the proposed method is illustrated through examples and numerical results. Here BPSK is used as the modulation scheme.

According to Section 3.3.2, the first step to find a good approximating functions is to observe the shape of the true LLRs and then propose the approximating functions. Next, using the optimization process (3.9), we optimize their parameters.

Example 3.3 *Consider a* 2×1 *MIMO-BICM system with perfect CSI in the receiver. Assuming the channel matrix* $\mathbf{H} = [2, -1]$ *and SNR*$= 5$ *dB, according to the shapes of true LLRs, we propose the following approximation functions*

$$\hat{l}_1 = \hat{f}_1^{\mathcal{O}_1}(y) = \alpha_1^1 \cdot y, \tag{3.14}$$

$$\begin{aligned} \hat{l}_2 = \hat{f}_2^{\mathcal{O}_2}(y) =& (\alpha_2^1 \cdot y + \beta_2^1)\mathbf{1}_{(y \le -r)} + (\alpha_2^2 \cdot y + \beta_2^2)\mathbf{1}_{(-r \le y \le r)} \\ &+ (\alpha_2^3 \cdot y + \beta_2^3)\mathbf{1}_{(y \ge r)}. \end{aligned} \tag{3.15}$$

Because of the symmetry properties of true LLRs, $\alpha_2^1 = \alpha_2^3$ and $\beta_2^1 = -\beta_2^3$ and $\beta_2^2 = 0$. Thus, the parameters needed to be optimized are $\mathcal{O}_1 = \{\alpha_1^1\}$, and $\mathcal{O}_2 = \{\alpha_2^1, \alpha_2^2, \beta_2^1\}$. For the given SNR, we optimize $\mathcal{O}_1$ and $\mathcal{O}_2$ by solving (3.9) and we find r via search. The optimized parameters are given by $\mathcal{O}_1 = \{-3.5532\}$ and $\mathcal{O}_2 = \{2.5758, 2.1682, -1.3784, 0.55\}$. And the corresponding approximate LLRs are depicted in Fig.3.3. Moreover, for the same SNR, approximate LLRs obtained by max-log approximation and the true LLR values are calculated and plotted in the

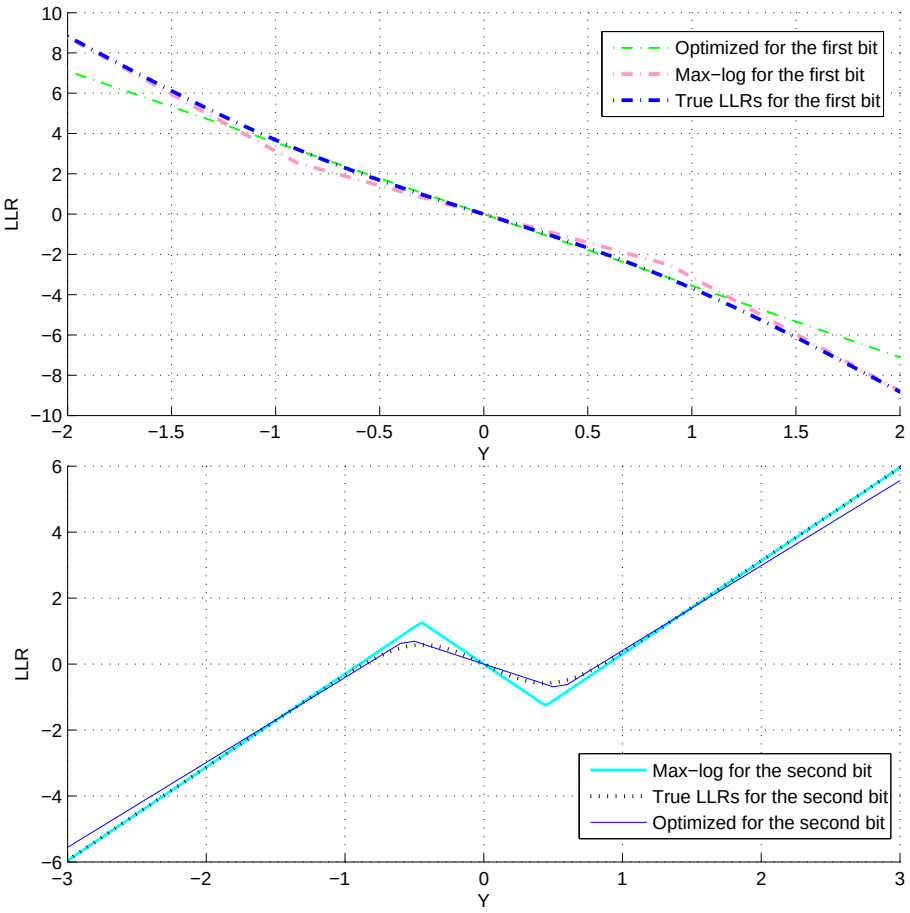

Figure 3.3: Comparing the true bit LLR values l with the optimized piecewise linear LLRs and max-log approximation for BPSK at $\gamma = 5$ dB.

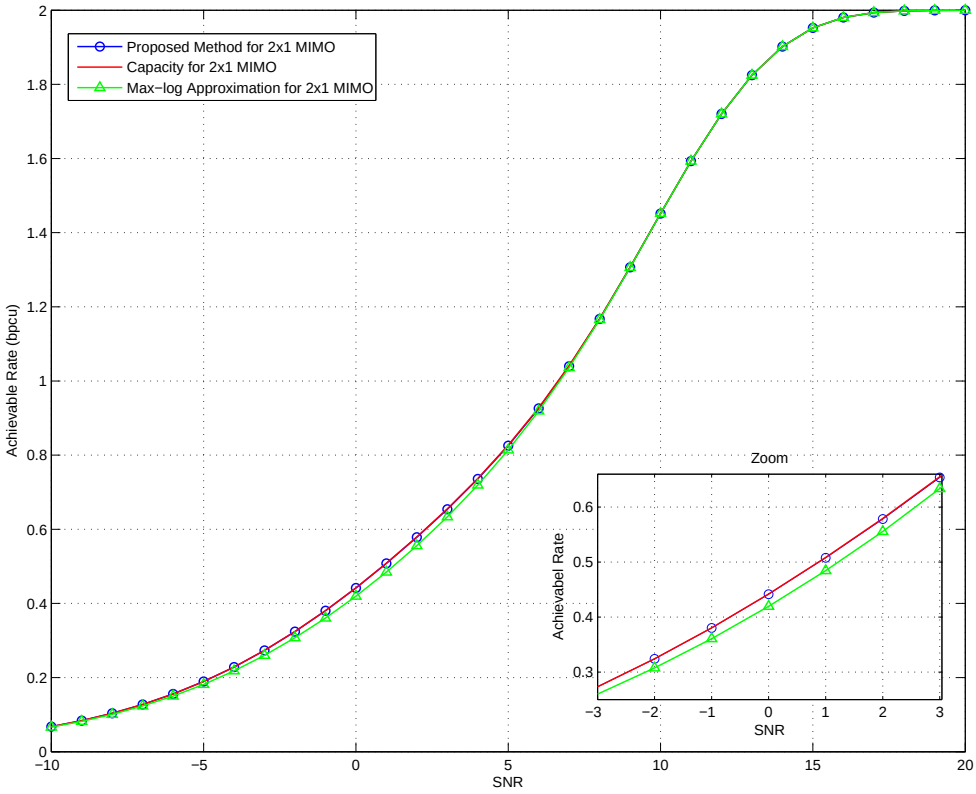

Figure 3.4: Comparing the achievable rate of MIMO-BICM channels with the achievable transmission rate under proposed approximate LLRs and max-log approximation.

same figure. It can be seen that the proposed optimized piecewise linear LLRs are more accurate than max-log in the small LLR regions, which affects the reliability of the received bits significantly.

To depict the performance of the piecewise linear LLR functions, in Fig. 3.4, we plot the maximum achievable transmission rate of our method, max-log approximation and the true LLR versus SNR. It can be seen that the achievable rate of piecewise linear approximation is always close to the capacity of MIMO-BICM system and better than that of max-log approximation. For example, at SNR$=2$ dB the gap of achievable rates between piecewise linear approximation and max-log approximation is about 0.5 dB.

Example 3.4 *Now consider 2×2 MIMO-BICM system without space-time coding. Assuming the channel matrix $\mathbf{H} = \begin{bmatrix} 1 & -2 \\ -3 & 4 \end{bmatrix}$ and the received signal $\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$. As discussed earlier, for BPSK, we can use linear approximation functions ($N_i = 1$). Thus,*

$$\hat{l}_1 = \hat{f}_1^{\mathcal{O}_1}(\mathbf{y}) = \alpha_1^1 y_1 + \alpha_1^2 y_2 + \beta_1^1, \tag{3.16}$$

$$\hat{l}_2 = \hat{f}_2^{\mathcal{O}_2}(\mathbf{y}) = \alpha_2^1 y_1 + \alpha_2^2 y_2 + \beta_2^1, \tag{3.17}$$

Thus the parameters are $\mathcal{O}_1 = \{\alpha_1^1, \alpha_1^2, \beta_1^1\}$, and $\mathcal{O}_2 = \{\alpha_2^1, \alpha_2^2, \beta_2^1\}$. Given SNR$=1$ dB, the optimized parameters are obtained as $\mathcal{O}_1 = \{-0.1581, 0.7797, 0\}$ and $\mathcal{O}_2 = \{0.7257, -1.2574, 0\}$. The achievable rate of the proposed method is 0.4821 bpcu while the capacity is 0.4859 bpcu and achievable rate of max-log approximation is 0.4574 bpcu.

3.4 Low Complexity Linear LLR Calculation for Two-way Relay Channel

In this section, we study the piecewise linear approximation for LLR calculation in two-way relay channels.

3.4.1 System Model

Relays have been found many applications in wireless network to enhance reliability and coverage [39, 40, 47]. Although one-way relaying has been widely considered in the literature, two-way relaying is more suitable when data flows in both directions. The two-way relay channel (TRC), which was first studied by Shannon [47], can be used to model many practical communication scenarios such as two separate mobile terminals communicating with each other by using a base station or a satellite. Recently, TRC has gained renewed interest form both academic and industry due to its numerous potential applications on cellular and peer-to-peer networks [48–50].

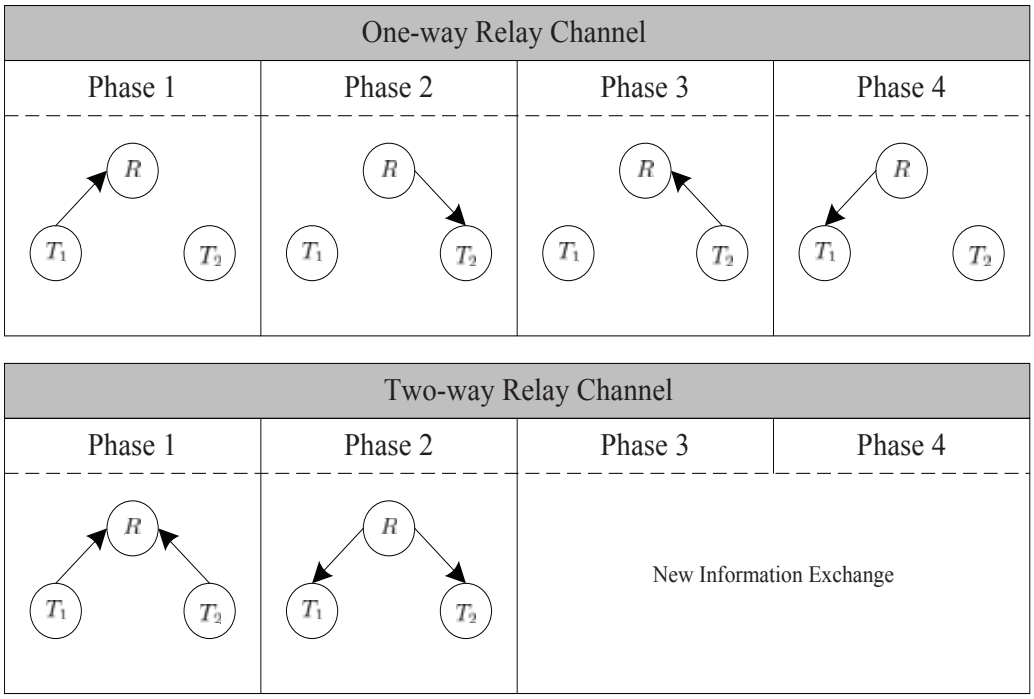

Figure 3.5: Comparison of information exchange under one-way and two-way relaying

One attractive feature of this two-way relay channel model is that it can improve the spectral efficiency of the one-way relay under half-duplex constrains [39, 40]. With a half-duplex relay node, one-way relaying has to use four phases to exchange information between two terminals, i.e., it takes two phases to send information

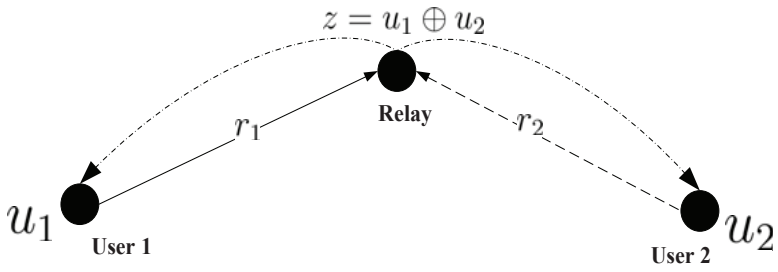

Figure 3.6: Two-way relay channel

from one terminal to the other terminal and another two phases for the reverse direction (see Fig. 3.5). However, with two-way relaying, we can improve the spectral efficiency by using only two phases to exchange information between the two terminals (see Fig. 3.5).

There are mainly two strategies at the relay: amplify-and-forward (AF) and decode-and-forward (DF) [39, 50].

Amplify-and-forward: Amplify-and-forward is the simplest form of relaying. As the name suggests, the relay simply amplifies the received signal before forwarding it to the destination. However, the main problem of AF is that the relay amplifies the noise as well, which may degrade the performance.

Decode-and-forward: A decode-and-forward relay will sample, demodulate and decode the received signal. Then, the regenerated and decoded signal will be transmitted to the destination. DF relays do not amplify the noise, thus when detection quality at the relay is good, the performance is usually better than that of AF relay. Moreover, when the relay decodes and forwards the binary sum ($\oplus$) of the input messages, comparing to the AF the sum-rate of the two terminals is increased by 50% and the decoding complexity is also reduced by 50% [50].

In this work, we consider a memoryless two-way relay fading channel with two communication nodes and one relay node shown in Fig.3.6. These two communication nodes want to transmit data to each other but without a direct path between them. Flat slow fading environment is assumed between terminals and the relay. Let $\mathbf{z} = \mathbf{c}_1 \oplus \mathbf{c}_2$ denotes the decoded symbol at the relay.

The system model is depicted in Fig.3.7. Let $U_i = [u_{i,1}, u_{i,2} \dots u_{i,K}], u_{i,k} \in$

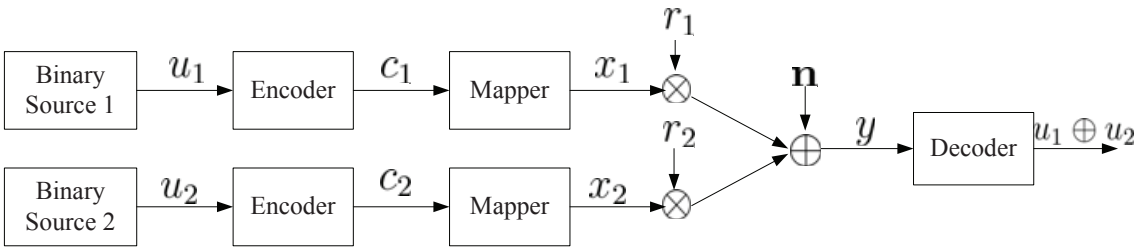

Figure 3.7: System Model

$\{0,1\}$ for $k = 1, 2, \ldots, K$ denote the information bits at node i and $C_i = [c_{i,1}, c_{i,2} \ldots c_{i,K}], c_{i,n} \in \{0,1\}$ for $n = 1, 2, \ldots, N$ denote the coded bits based on U_i. Assuming both nodes using the same channel coding schemes Γ, where Γ is a reversible mapping function, the relation between U_i and C_i is shown as

$$\Gamma(U_i) = C_i, \quad \Gamma^{-1}(C_i) = U_i \tag{3.18}$$

where Γ^{-1} represents the decoding process. If Γ is a linear scheme then we have

$$\Gamma(U_1 \oplus U_2) = \Gamma(U_1) \oplus \Gamma(U_2). \tag{3.19}$$

Thus, the same decoding scheme can be used at the relay [51].

At the relay, the received signal is given by

$$y = r_1 x_1 + r_2 x_2 + n. \tag{3.20}$$

where x_i represents transmitted symbol at terminal i, $i = 1, 2$, and n is the Gaussian noise with zero mean and variance σ^2. Also, $r_i > 0$ are independent channels gains with arbitrary pdfs, f_i, $i = 1, 2$.

For simplicity, BPSK modulation is assumed for the proposed scheme, which can be extended to other typical modulation as the future work. Thus, the transmitted symbol is given by

$$x_i = 2c_i - 1. \tag{3.21}$$

3.4.2 LLR Calculation and Problem Definition

Assuming that σ is known at the relay and uniform input distribution, three scenarios can be considered depending on the availability of r_i at the relay.

Known CSI

In this case, both channel gains, r_1 and r_2, are known at the relay for each bit. This scenario usually happens when two channels are both good and there are accurate channel estimations. Thus, the conditional pdf is given by

$$p(\mathbf{y}|x_1, x_2, r_1, r_2) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(\mathbf{y} - r_1 x_1 - r_2 x_2)^2}{2\sigma^2}\right)$$

Let z be the message which the relay would forward to both communication nodes. The relationship between z and x_1, x_2 is shown in Table 2.1.

x_1	x_2	z
-1	-1	0
-1	1	1
1	-1	1
1	1	0

Table 3.1: The relation between the output information z and input information, x_1, x_2.

Therefore, the LLR is given by

$$\begin{aligned} l(y) &= \log \frac{P(\mathbf{z} = 0|\mathbf{y}, r_1, r_2)}{P(\mathbf{z} = 1|\mathbf{y}, r_1, r_2)} \\ &= \log \frac{e^{-\frac{(\mathbf{y} - r_1 - r_2)^2}{2\sigma^2}} + e^{-\frac{(\mathbf{y} + r_1 + r_2)^2}{2\sigma^2}}}{e^{-\frac{(\mathbf{y} - r_1 + r_2)^2}{2\sigma^2}} + e^{-\frac{(\mathbf{y} + r_1 - r_2)^2}{2\sigma^2}}}. \end{aligned} \tag{3.22}$$

Partial CSI

In this case, only one channel gain, r_1 or r_2, is known at the relay for each bit. This usually happens when only one channel is good. Without loss of generality, we assume that r_1 is known. Thus the conditional pdf is

$$p(\mathbf{y}|x_1, x_2, r_1) = \int_{-\infty}^{+\infty} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(\mathbf{y} - r_1 x_1 - r_2 x_2)^2}{2\sigma^2}\right) f(r_2) dr_2 \tag{3.23}$$

Then, the LLR is given by

$$\begin{aligned} l(y) &= \log \frac{P(\mathbf{z}=0|\mathbf{y},r_1)}{P(\mathbf{z}=1|\mathbf{y},r_1)} \\ &= \log \frac{p(\mathbf{y}|x_1=-1,x_2=-1,r_1)+p(\mathbf{y}|x_1=+1,x_2=+1,r_1)}{p(\mathbf{y}|x_1=-1,x_2=+1,r_1)+p(\mathbf{y}|x_1=+1,x_2=-1,r_1)}. \end{aligned} \tag{3.24}$$

Thus, finding $l(y)$ requires calculating four integrals which is quite complex for the practical implementation.

Unknown CSI

In this case, both channel gains, r_1 and r_2, are unavailable at the relay. This scenario usually happens when both channels are poor or no accurate channel estimation applied, which is much more realistic especially for the wireless channels. Thus, the pdf of the received signal conditioned on x_1, x_2 being transmitted is given by

$$p(\mathbf{y}|x_1,x_2) = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} p(\mathbf{y}|x_1,x_2,r_1,r_2)f(r_1)f(r_2)dr_1dr_2 \tag{3.25}$$

Therefore, the LLR is

$$\begin{aligned} l(y) &= \log \frac{P(\mathbf{z}=0|\mathbf{y})}{P(\mathbf{z}=1|\mathbf{y})} \\ &= \log \frac{p(\mathbf{y}|x_1=-1,x_2=-1)+p(\mathbf{y}|x_1=+1,x_2=+1)}{p(\mathbf{y}|x_1=-1,x_2=+1)+p(\mathbf{y}|x_1=+1,x_2=-1)}. \end{aligned} \tag{3.26}$$

Thus, finding $l(y)$ requires calculation of four double integrals which is much more complex than the former two cases.

Problem Definition

It is clear that Case 3 is the most general case, i.e., if r_i is known, we must replace its pdf with a delta function which would reduce (3.26) to (3.24), and similar will reduce (3.24) to (3.22). Thus, we will focus on Case 3 in this work, where no CSI is available at the relay. As can be seen from (3.26), the relationship between the LLR and channel output, which includes four double integrations, is rather complex. This means that the calculation of LLR at the relay is highly time and energy

consuming, which is not practical especially when there are limited power available at the relays. Thus, efficient LLRs approximation method is of practical interest.

One simple method is to use expected value of r ($E(r)$) as the approximate channel coefficients. Let us call this method mean square approximation. This method simply removes the two integrations involved in the LLR calculation, thus reduces the complexity significantly. However, the expected value, $E(r)$, is only the minimum mean square error estimation of r, which cannot guarantee the optimum performance in the decoder. Besides, the computation is still very large especially when high-order modulations are used.

Another practical approximation is to use the max-log approximation (3.1), leading to piece-wise linear approximate LLR functions when perfect CSI is known at the relay [11]. However, when CSI is not available at the relay, the max-log approximation becomes

$$
\begin{aligned}
l_{max}(y) &= \log \frac{\max_{(x_1,x_2)\in\mathcal{X}^0} p(\mathbf{y}|x_1,x_2)]}{\max_{(x_1,x_2)\in\mathcal{X}^1} p(\mathbf{y}|x_1,x_2)]}. \qquad (3.27)\\
&= \log \left(\frac{\max\limits_{(x_1,x_2)\in\mathcal{X}^0} \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} e^{-\frac{(\mathbf{y}-r_1x_1-r_2x_2)^2}{2\sigma^2}} f(r_1)f(r_2)dr_1dr_2}{\max\limits_{(x_1,x_2)\in\mathcal{X}^1} \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} e^{-\frac{(\mathbf{y}-r_1x_1-r_2x_2)^2}{2\sigma^2}} f(r_1)f(r_2)dr_1dr_2} \right)
\end{aligned}
$$

where $\mathcal{X}^0$ and $\mathcal{X}^1$ denote subsets of transmit vectors when $z = 0$ and $z = 1$ respectively.

It is obvious that it is no longer piece-wise linear and much complicated for practical implementation as it involves four double integrations. Moreover, max-log approximation usually performs good at high SNR region, when there is a dominant term in the sum sequence. However, low SNR region is more common especially for the wireless channel. Therefore, a better approximation is desired.

3.4.3 Piece-wise Linear Approximation

In this work, we are seeking approximate LLR as piece-wise linear functions of the output $\mathbf{y}$ when there is no CSI available at the relay. Let $\hat{l} = \hat{f}(\mathbf{y})$ denotes

the approximate LLR functions. To optimize the LLR approximating function, an accuracy measure is required.

LLR Accuracy Measure

Notice that from the point of view of the relay, we can model the channel as a channel whose input is $c_1 \oplus c_2$ and whose output is $y = r_1 x_1 + r_2 x_2 + n$. The capacity of this channel can be given by (3.4).

Instead of true LLR, let us replace them with the approximate LLR. As a result, (3.4) becomes

$$\hat{R} = 1 - \frac{1}{2}\int_{-\infty}^{+\infty} \log_2(1 + e^{-\hat{l}})\hat{p}^0(\hat{l})d\hat{l} - \frac{1}{2}\int_{-\infty}^{+\infty} \log_2(1 + e^{\hat{l}})\hat{p}^1(\hat{l})d\hat{l}, \qquad (3.28)$$

where $\hat{p}^b(\hat{l})$ is the conditional pdf of approximate LLR given $c = b$ transmitted. We can see that (3.28) gives an achievable rate of the system operating under approximate LLRs, thus $\hat{R} \leq \mathcal{C}$, where the equality holds when $\hat{p}^b(\hat{l}) = p^b(l)$ [14]. As a result, by maximizing (3.28) good approximating functions can be found in the sense of maximizing the achievable rates on the channel.

LLR Approximation

In this work, we use the piece-wise linear functions as our approximate function because of its simplicity and ease of implementation. Hence, it can be efficiently solved by different numerical optimization techniques.

From the definition of LLR in (3.26), it is easy to see that $l(0) = l_{max}(0)$ under BPSK modulation. Let us assume the piece-wise linear approximation also crosses this point and define

$$k = l(0) = \log \frac{\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} \exp\left(-\frac{(r_1+r_2)^2}{2\sigma^2}\right) f(r_1)f(r_2)dr_1dr_2}{\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} \exp\left(-\frac{(r_1-r_2)^2}{2\sigma^2}\right) f(r_1)f(r_2)dr_1dr_2}. \qquad (3.29)$$

For case 1 (i.e., perfect CSI), using this condition k can be found in close form as $k = -\frac{2r_1r_2}{\sigma^2}$. For the case 2, when r_2 is available and under normalized Rayleigh channel (i.e.,$f(r_1) = 2re^{-r_1^2}$) [52].

$$k = \log \frac{\Phi(-r_2/\sqrt{2\sigma^2(1+2\sigma^2)})}{\Phi(r_2/\sqrt{2\sigma^2(1+2\sigma^2)})}. \tag{3.30}$$

where $\Phi(t) = 1 + \sqrt{\pi} t e^{t^2} \operatorname{erfc}(-t)$ and $\operatorname{erfc}(\cdot)$ is the complementary error function. However, for more general case there is no close form function, and k is obtained by Monte Carlo simulation.

To summarize, the LLR approximate function can be found by following steps:

Step1) The real number domain $\mathbb{R}$ is partitioned into a finite number, N, regions $\mathbb{R}_1, \ldots, \mathbb{R}_N$. The regions and N is usually chosen based on the shape of true LLRs and the accuracy requirements.

Step2) For each region $\mathbb{R}_k$, a linear function, $\hat{f}_k = \alpha_k y + \beta_k$, is proposed. Thus,

$$\begin{aligned} \hat{l} = \hat{f}(\mathbf{y}) &= \sum_{k=1}^{N} (\boldsymbol{\alpha}_k \mathbf{y} + \boldsymbol{\beta}_k) \mathbf{1}_{(\mathbf{y} \in \mathbb{R}_i)}. \\ \text{with } \hat{l}(y=0) &= k. \end{aligned} \tag{3.31}$$

where $\mathbf{1}_{(\cdot)}$ denotes the indicator function.

Step3) Let $\mathcal{O} = \{\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_N, \beta_1, \ldots, \beta_N\}$ denotes the set of parameters, which can be found by solving the following optimization problem:

$$\mathcal{O}^{\text{opt}} = \arg \max_{\mathcal{O}} \hat{R}, \tag{3.32}$$
$$\text{subject to } \Psi(\mathcal{O})=0.$$

Here, $\Psi(\mathcal{O}) = 0$ represents the constraints that may be imposed on $\mathcal{O}$, such as continuity conditions.

Step4) The regions $\mathbb{R}_1, \ldots, \mathbb{R}_N$ can be optimized by search.

3.4.4 Numerical Results and Discussions

In this part, we present examples and numerical results based on the proposed piece-wise linear approximation for BPSK modulation on TRC model.

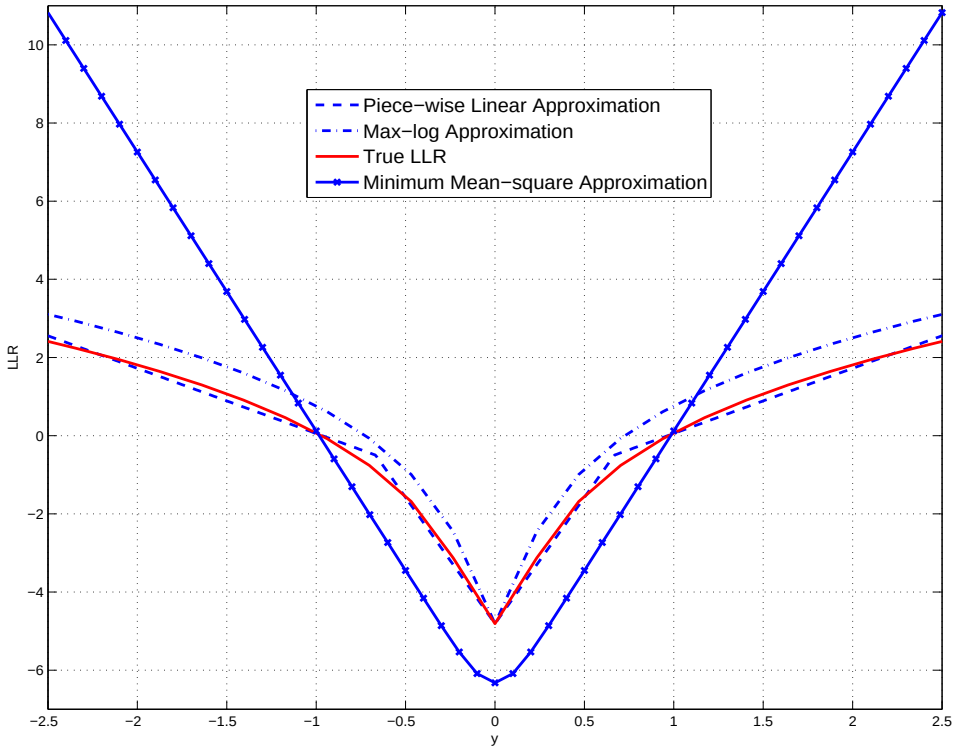

Figure 3.8: Comparison of the shape of true LLR at 5dB with that of piece-wise linear approximation, max-log approximation and minimum mean-square approximation

Example 3.5 *Consider a two-way relay fading channel with DF at the relay and the channel between the terminals and the relay are normalized Rayleigh channel (i.e.,*$f(r) = 2re^{-r^2}, E(r) = \frac{\sqrt{\pi}}{2}$*). According to the curves of true LLR, we propose the following approximation functions:*

$$\begin{aligned}\hat{l} = \hat{f}^{\mathcal{O}}(y) =&(\alpha_1 \cdot y + \beta_1)\mathbf{1}_{(y\leq -r)} + (\alpha_2 \cdot y + k)\mathbf{1}_{(-r\leq y\leq 0)} \\ &+ (\alpha_3 \cdot y + k)\mathbf{1}_{(0\leq y\leq r)} + (\alpha_4 \cdot y + \beta_4)\mathbf{1}_{(y\geq r)}.\end{aligned} \tag{3.33}$$

Due to the symmetry of the LLR, it can be assumed that $\alpha_1 = -\alpha_4$, $\beta_1 = \beta_4$ and $\alpha_2 = -\alpha_3$. Also, because of the continuity condition r can be obtained by

$$r = \frac{k - \beta_1}{\alpha_2 - \alpha_1}.$$

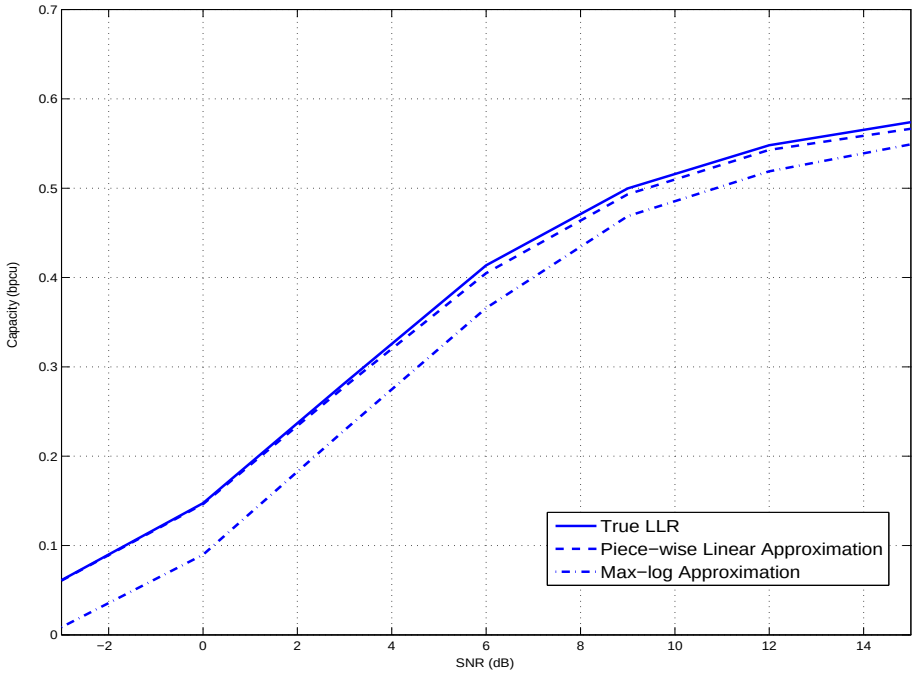

Figure 3.9: Comparing the achievable rate under the piece-wise linear approximation, max-log approximation and true LLR

As a result, $\mathcal{O} = \{\alpha_1, \alpha_2, \beta_1\}$. For the given SNR, optimized piece-wise linear approximation function can be found by the steps described in Section 3.4.3.

For example, when SNR$= 5$ dB the optimized parameters can be found as $\mathcal{O} = \{6.4606, 1.6667, -1.6113\}$, and $k = -4.811, r = 0.6675$. The corresponding approximate LLRs and true LLRs are depicted in Fig. 3.8. It can be seen that the piece-wise linear LLR is much closer to the true shape than the max-log approximation and the minimum mean-square approximation especially around LLR $= 0$.

In Fig. 3.9, we depict the achievable transmission rate for the designed piece-wise linear approximation, max-log approximation and the true LLRs case. We can see that the achievable rate of our method is always closer to the true capacity than the max-log approximation especially when SNR is low. For example, at SNR$=$ $5\ dB$, the achievable rate of the proposed method is 0.3676 bpcu and the capacity is 0.3740 bpcu, while the achievable rate of max-log approximation is 0.3271 bpcu. On average, it can be seen from the curves that the gap between the piece-wise

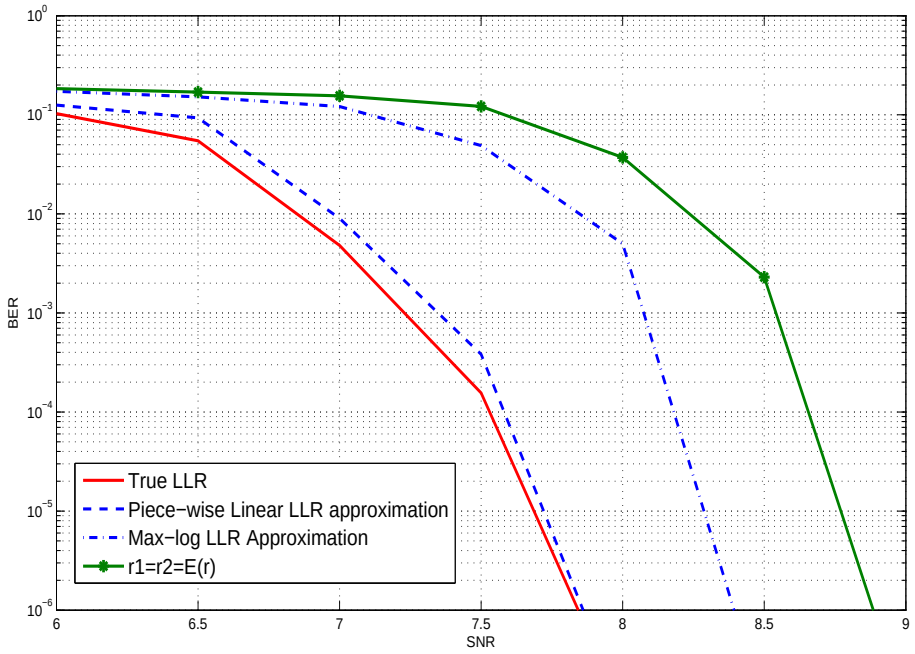

Figure 3.10: Comparison among the BER of a randomly constructed (3,4)-regular LDPC code of length 15000 decoded by true and approximate LLRs on two way relay fading channel.

linear approximation and max-log approximation is around 1 dB. What is more, max-log approximation is much more complex than the proposed method.

To evaluate the decoding performance of the optimized piece-wise linear approximations, we compare the BER under the proposed method, the max-log approximation, minimum mean-square approximation and the true LLRs. As an example, the performance of a (3,4)-regular LDPC code of length 15000 is depicted for these four cases in Fig. 3.10. It is clear that the performance of the optimized approximate LLR is quite close to that of the true LLRs, about 1 dB better than that of the max-log approximation and about 1.5 dB better than the minimum mean-square approximation.

3.5 Conclusion

LLR calculation can be very complicated in a real wireless communication environment. This motivates us to find efficient and accurate approximation methods. In this chapter, we deal with this problem for two different cases, MIMO channels and two-way relay channels.

In the Section 3.3, an efficient method using piecewise linear functions for approximating LLRs over flat slow-fading MIMO channels is studied. The system considered in this work is MIMO-BICM which allows us to compute LLRs at bit levels. To this end, an accuracy measure and approximation functions for MIMO-BICM systems are proposed. We use this measure to optimize the parameters in the approximation functions. The performance of the proposed method is investigated in terms of the achievable rate. Our results show that the performance of the optimized piecewise linear approximation function is superior to that of the max-log approximation. In fact, the achieved performance is very close to that of true LLRs. Compared to max-log, our method is more complex because it needs the parameters to be optimized. Thus, our method is more attractive when channel gains are not subject to rapid change.

For two way relay fading channel with DF strategy, the relay has to calculate channel LLR for the decoding. When no CSI is available at the relay, the LLR computation is much more complicated and computationally expensive. Thus, approximate LLR are desired. In Section 3.4, we used an accuracy measure based on the achievable rate and under this measure, a piece-wise linear approximation is proposed. For the normalized fading channels, we investigated the performance of the proposed method in terms of achievable rate and BER. Our experimental results showed that the proposed piece-wise linear LLR approximation outperforms the max-log approximation and minimum mean-square approximation and its performance was very close to that of true LLRs. The proposed approximation is less complex and easier to implement than the max-log approximation or the true LLRs.

Chapter 4

Design of LDPC Codes with Strong Universal Properties

4.1 Introduction and Background Knowledge

As we discussed in Chapter 1, LDPC codes can approach the Shannon limit on many channels. However, design of LDPC codes can be quite complex. For example, LDPC codes should be chosen carefully based on different channel types and channel parameters. If the parameters change, the code design process should be done again, otherwise, the performance would degrade.

In this chapter, we are dealing with these problems by designing the so called "universal" LDPC codes. Before we go to further details, let us first introduce some background knowledge.

4.1.1 Universal Codes

Designing a code that can successfully be used over a multitude of channels is of great theoretical and practical interest. Such robust codes are called universal codes. Universal codes have various advantages over channel codes that are designed for a specific channel.

Usually, the term "universal codes" have two meanings in the context of channel coding. First, a coding scheme that does not have a specific rate and works well on a family of channels of different capacity, but of the same type, e.g., BEC with different erasure rates. Raptor codes [10], for example, are universal in this sense.

Second, to have a channel coding scheme which can be employed over all channels of equal capacity. The latter case is what we mean by an universal code in this chapter.

There has been some recent efforts to design LDPC codes which perform sufficiently well on a variety of BIMSs with identical capacity. For example, the idea of designing a code for a properly chosen surrogate channel, among a limited set of given channels, and then using it over other members of the set has been discussed in [53]. This approach is a practical solution, but unfortunately can handle a limited number of channels. Universal codes can also be found analytically, but those codes can only achieve a very low percentage of the capacity [54].

4.1.2 Stability Condition for Density Evolution

Fixed point of density evolution is often used to analyze the convergence of LDPC codes. When density evolution has only a zero-error fixed point, i.e., δ_∞, the given code can be successfully decoded. It is desirable that this fixed point be *stable*. Thus, the stability property can be seen as a condition which ensures that once the density has been evolved to something "close" to perfect decoding, it will converge to the perfect decoding fixed point and the decoding will be successful [17].

For the sum-product decoder, the parameter that characterizes the channel with respect to the stability of the system is the Bhattacharyya parameter $\mathcal{B}(\cdot)$ introduced in Section 2.3. Thus, we have the following stability condition theorem:

Theorem 4.1 [*Stability Condition for Sum-product Decoder* [17]]: *Assuming we have a degree distribution pair $(\lambda(x), \rho(x))$ and a symmetric channel with LLR pdf f_{ch}. For $l \geq 0$ define*

$$f_l = f_0 \otimes \lambda(\rho(f_{l-1})).$$

with an arbitrary f_0, we have:

[Sufficiency**]** *If $\lambda'(0)\rho'(0)\mathcal{B}(f_{ch}) < 1$[1], then there does exist a strictly positive constant $\xi = \xi(\lambda, \rho, f_{ch})$ such that if, for some $l \in \mathbb{N}$, $\mathcal{P}_e(f_l) \leq \xi$, then f_l converge to δ_∞.*

[1] $\lambda'(x)$ and $\rho'(x)$ are the derivatives of $\lambda(x)$ and $\rho(x)$, respectively.

[Necessity**]** *If $\lambda'(0)\rho'(0)\mathcal{B}(f_{ch}) > 1$, then there exists a strictly positive constant $\xi = \xi(\lambda, \rho, f_{ch})$ such that for all $f_0 \neq \delta_\infty$,*

$$\lim_{l\to\infty} \mathcal{P}_e(f_l) > \xi.$$

Example 4.1 *For a BEC(ε) with the LLR pdf given by (2.4), we have*

$$\mathcal{B}(f_{BEC(\varepsilon)}) = \int_{\infty}^{\infty} (\varepsilon\delta_0(x) + (1-\varepsilon)\delta_\infty(x))e^{-x/2}dx = \varepsilon$$

which results in the stability condition for BEC channel

$$\lambda'(0)\rho'(0) < \frac{1}{\varepsilon}$$

However, it is worth mentioning that stability condition does not guarantee the convergence of the code. For example, for LDPC codes with $\lambda'(0) = 0$, the zero-error fixed point is always stable while the codes may not converge.

4.1.3 Information Combining Bounds

In the context of LDPC codes and under sum-product decoding, the following bounds are established in [55, 56]:

Theorem 4.2 [*Information Combining Bounds*]:

Bound 1**:** *Consider a variable node of degree d and assume that all except one of the input messages have known LLR pdfs. The other input message has a fixed mutual information I_{in} with its actual value, but its LLR distribution is unknown. The mutual information between the output message of the variable node and its actual value is minimized if this unknown LLR pdf is $f_{\text{BSC}(h^{-1}(1-I_{\text{in}}))}(x)$, where $h(\cdot)$ is the binary entropy function.*

Bound 2**:** *Consider a check node of degree d and the same assumptions with Bound 1. Thus, the mutual information between the output message of the check node and its actual value is minimized if this unknown LLR pdf is $f_{\text{BEC}(1-I_{\text{in}})}(x)$.*

4.2 Universal Codes Design

In this section, without considering a specific set of channels, our goal is to find LDPC codes with good universal properties over all BIMSs with a given capacity. The simplest way to tackle the problem is to use information combining bounds in order to obtain codes with guaranteed convergence over all channels with the given capacity. Unfortunately, using these bounds results in very conservative (thus inefficient) codes. To improve the efficiency, a refined Gaussian approximation together with the information combining bounds is suggested to find codes with good universal properties. We argue why these codes should have strong universal behavior and the designed codes is also proved to satisfy the universal stability condition (i.e., stability condition on all channels).

4.2.1 Problem Description

Clearly, information combining bounds can be used to find codes that are guaranteed to converge over every channel with a given capacity $\mathcal{C}$. To guarantee universal convergence, it is sufficient to observe convergence when worst case pdfs at both variable and check nodes are assumed. EXIT curves can be plotted for the worst case pdfs, and a sufficient condition on universal convergence is to have an open tunnel between these two EXIT curves. Therefore, the procedure for finding these codes is not different from designing codes for specific channels, which are studied in the literature in various forms [15, 33].

Unfortunately, codes obtained by applying both information combining bounds have poor performance because of the stringent constraints forced by these bounds. For example, when $f_{v\to c}(x)$ (the pdf of input messages to check nodes) is $f_{\mathrm{BEC}(\epsilon)}(x)$, the pdf of the output of check nodes ($f_{c\to v}(x)$) is $f_{\mathrm{BEC}(1-\rho(1-\epsilon))}(x)$. That is to say, $f_{c\to v}(x)$ is of BEC form and not BSC (which is assumed for the worst case scenario). Interestingly, BEC maximizes the output mutual information at the variable nodes [55, 56]. In other words, in this example, to pass the sufficient condition of convergence at a variable node, the best pdf (i.e., BEC) in reality, is replaced with the worst pdf (i.e., BSC).

Thus, using both bounds, one at the variable nodes and one at the check nodes, results in too stringent constraints. To improve the efficiency, a new method is desired. In the rest of this chapter, we denote the method that applies both bounds as the sufficient condition of the codes' convergence.

4.2.2 Refined Gaussian Approximation

A symmetric Gaussian pdf for the messages at the output of the variable nodes is a common assumption, regardless of the channel type [28]. Notice that under sum-product decoding, the output messages of variable nodes are obtained by adding all the input LLR values, so a Gaussian distribution is a reasonable assumption because of the central limit theorem.

Based on this common practice and the information combining bounds, a new approach for designing universal LDPC codes is proposed. We suggest replacing the information combining bound for $f_{v\to c}(x)$ with a Gaussian approximation at the output of variable nodes. But for $f_{c\to v}(x)$, we still force the information combining bound which means that they are assumed to be in BSC form. We call this approach a refined Gaussian approximation since unlike [28], $f_{c\to v}(x)$ is not assumed to be Gaussian.

To be more specific, given a value of mutual information I_{in}, at the input of the check nodes, we find a symmetric Gaussian distribution, $f_{\text{G}}(x)$ with $\sigma^2 = 2\mu$, whose mutual information according to (2.5) is equal to I_{in}. We use this Gaussian distribution at the input of the check nodes to find their output density $f_{\text{out}}(x)$ via density evolution [25]. We then find the mutual information $I_{c\to v}$ of this output density using (2.5). At the input of the variable nodes, we assume $f_{\text{BSC}(\epsilon)}(x)$, $\epsilon = h^{-1}(1 - I_{c\to v})$, which is the worst case pdf according to the information combining bounds. Then, the output pdf of degree-i variable nodes will be

$$g^{(i)}(x) = \left[\bigotimes_{k=1}^{i-1} f_{\text{BSC}(\epsilon)}(x)\right] \otimes f_{\text{BSC}(\eta)}(x), \tag{4.1}$$

where $\eta = h^{-1}(1 - \mathcal{C})$ and $\mathcal{C}$ is the channel capacity. Therefore, the output mutual

information using (2.5) is

$$I_{\text{out}} = \mathcal{I}\big(\sum \lambda_i g^{(i)}(x)\big) = \sum \lambda_i \mathcal{I}\big(g^{(i)}(x)\big).$$

For code design, we enforce $I_{\text{out}} > I_{\text{in}}$ as a constraint for choosing λ_i. Detailed formulations of code design are similar with the process introduced in Chapter 2.

This method raises two minor problems. First, if the channel is indeed a BEC, $f_{v\to c}(x)$ remains in the BEC forms for all iterations, i.e., it cannot be approximated with a Gaussian distribution. Second, for the first iteration where $f_{v\to c}(x)$ is the channel observation but not the result of summations at the variable nodes, a Gaussian approximation is not accurate. To resolve this issue, we append another set of constraints, that guarantees the convergence on the BEC, to the code design formulation. This way, for the first iteration, the worst case pdf is forced at the input of check nodes too [55, 56].

These codes are expected to have good universal properties because of the fact that for a wide range of channel types, the Gaussian approximation at the output of variable nodes is a reasonable assumption. Also, at the output of check nodes the worst case scenario is enforced. In fact, in the next section, it will be proved that these codes satisfy a universal stability condition.

This code design approach, compared to using both information combining bounds, results in codes with significantly smaller gap to capacity (particularly at low capacities). For example at $\mathcal{C} = 0.2$, sufficient condition for universal convergence results in codes that achieve only 70% of the capacity, but our approach gives codes which achieve more than 85% of the capacity. At capacities close to 1 bpcu, the worst case and the best case scenario pdfs at variable (check) nodes are alike, i.e.,

$$\lim_{\epsilon\to 0} f_{\text{BEC}(\epsilon)}(x) = \lim_{p\to 0} f_{\text{BSC}(p)}(x) \tag{4.2}$$

Therefore, according to (2.5), the mutual information of the best case scenario and the worst case scenario are approximately equal. It means that the output mutual information mainly depends on the input mutual information but not the LLR pdfs. Thus, the rate loss due to enforcing the sufficient condition of convergence

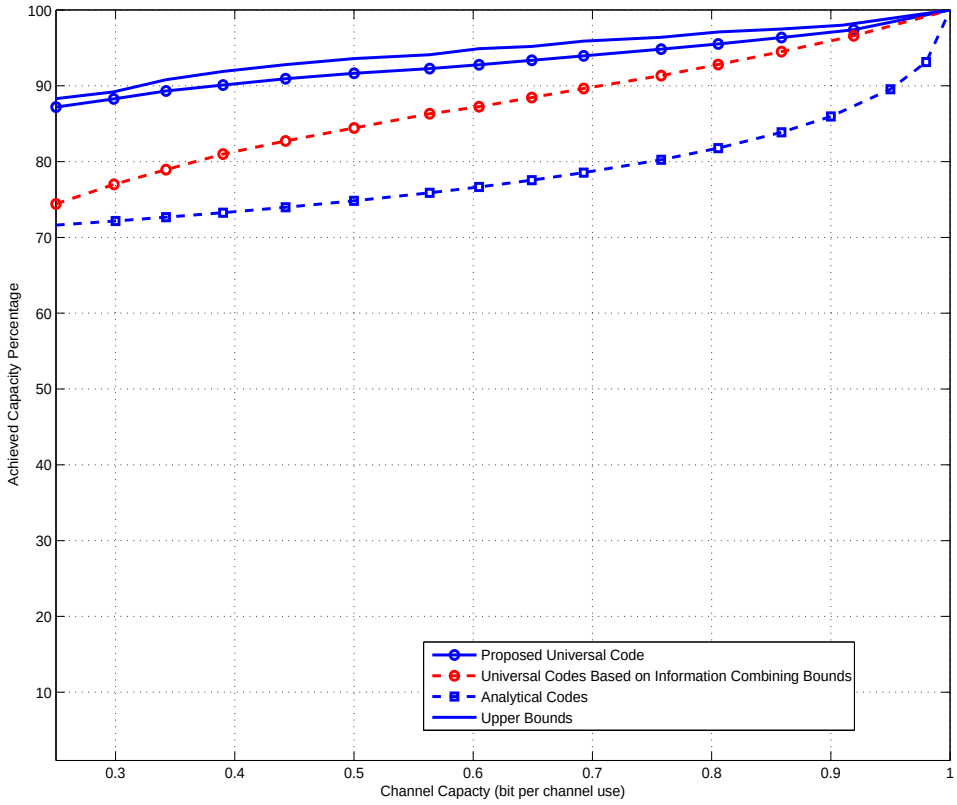

Figure 4.1: Comparison of the achievable rates of our codes (in terms of achievable percentage of capacity) with codes based on information combining bounds, analytical codes and an upper bounds.

is minor and a very high percentage of capacity is achievable. This can be seen in Fig. 4.1.

4.2.3 Stability Analysis

In this section we prove that codes that are designed using the refined Gaussian approximation satisfy a universal stability condition. In other words, they satisfy the stability condition for all channels with the given capacity.

Consider a channel with capacity $\mathcal{C}$ and a code designed by using our method. The message error rate at the input to check nodes is denoted by p_{in} and at the output of check nodes by ϵ. As the decoder gets close to perfect decoding, p_{in} is

getting closer to 0.

$$\begin{aligned}\lim_{p_{\text{in}}\to 0}\frac{\epsilon}{p_{\text{in}}} &= \lim_{p_{\text{in}}\to 0}\frac{1-\rho(1-2p_{\text{in}})}{2p_{\text{in}}}\\ &= \lim_{p_{\text{in}}\to 0}\frac{\rho'(1-2p_{\text{in}})}{2}\\ &= \sum \rho_i(i-1)\\ &= \rho'(1).\end{aligned}$$

It means that when the decoder is approaching perfect decoding, ϵ can be approximated as $\rho'(1)p_{\text{in}}$. Therefore, the density of BSC input to variable nodes is $f_{\text{BSC}(\epsilon)}(x)$. For degree-$i$ variable nodes, the output error rate will be $p^{(i)} = \int_{-\infty}^{0} g^{(i)}(x)dx$, where $g^{(i)}(x)$ is given in (4.1). For a degree-2 variable node, it follows that

$$\begin{aligned}p^{(2)} &= \int_{-\infty}^{0} f_{\text{BSC}(\eta)} \otimes f_{\text{BSC}(\epsilon)}(x)dx\\ &= \int_{-\infty}^{0} \Big((1-\epsilon)\,\delta_{\log(\frac{1-\epsilon}{\epsilon})}(x) + \epsilon\,\delta_{-\log(\frac{1-\epsilon}{\epsilon})}(x)\Big) \otimes\\ &\quad \Big((1-\eta)\,\delta_{\log(\frac{1-\eta}{\eta})}(x) + \eta\,\delta_{-\log(\frac{1-\eta}{\eta})}(x)\Big)\,dx\\ &\overset{\text{a}}{=} \int_{-\infty}^{0} \Big((1-\epsilon)\,\delta_{\log(\frac{1-\epsilon}{\epsilon})}(x) + \epsilon\,\delta_{-\log(\frac{1-\epsilon}{\epsilon})}(x)\Big)\,dx\\ &= \epsilon,\end{aligned}$$

where (a) results from $\epsilon \ll \eta$. The message error rate at the output of the variable nodes is

$$p_{\text{out}} = \sum_{i=2}^{d_v} \lambda_i p^{(i)} > \lambda_2 \epsilon = \lambda'(0)\rho'(1)p_{\text{in}}.$$

Our code design procedure enforces $I_{out} > I_{in}$, which for consistent Gaussian pdfs is equivalent to $p_{\text{out}} < p_{\text{in}}$. Thus, our method guarantees that

$$\lambda'(0)\rho'(1) < 1. \tag{4.3}$$

It is shown in [57] that (4.3) is the worst stability condition a channel might require. Thus, the method proposed here assures a universal stability condition.

4.2.4 Design Example

In this section we design codes with strong universal properties using the method proposed in Section 4.2.2. For a given channel capacity $\mathcal{C}$, the goal of the code design in this work is to maximize the code rate through optimizing $\lambda(x)$, when $\rho(x)$ is assumed given and convergence constraints of Section 4.2.2 are satisfied. The motivation for fixing $\rho(x)$ and optimizing $\lambda(x)$ stems from the fact (observed by various authors) that the code performance is much more sensitive to the choice of $\lambda(x)$ than $\rho(x)$. For choosing $\rho(x)$, following the guidelines of [28], we assume that $\rho(x)$ comprises at most two consecutive degrees, and we optimized $\rho(x)$ via search.

Example 4.2 *For $\mathcal{C} = 0.5$ bpcu, and allowing a maximum node degree of 50 in the code, the rate of the optimized code is 0.4584 (91.68% of the capacity). The degree distribution is as follows:*

$$\begin{aligned} \rho(x) &= 0.0899x^7 + 0.9101x^8. \\ \lambda(x) &= 0.1220x + 0.3236x^2 + 0.2963x^8 + \\ &\quad 0.0062x^9 + 0.2519x^{49}. \end{aligned} \tag{4.4}$$

A single randomly chosen code has been generated from this degree distribution. The code has been tested over 1000 randomly generated channels with capacity $\mathcal{C} = 0.5$ and in all cases convergence is observed. According to the concentration theorem [25], for a given degree distribution, the performance of a randomly chosen code converges to the average of the ensemble. Thus, we expect to observe a similar behavior from any other randomly constructed code from this degree distribution.

4.2.5 Numerical Results

The achievable rates over a wide range of channel capacities are plotted in Fig.4.1. The results are also compared with codes that are designed based on applying both information combining bounds. Besides, the achievable rates of analytically constructed universal codes of [54] is also plotted here for the comparison. All codes

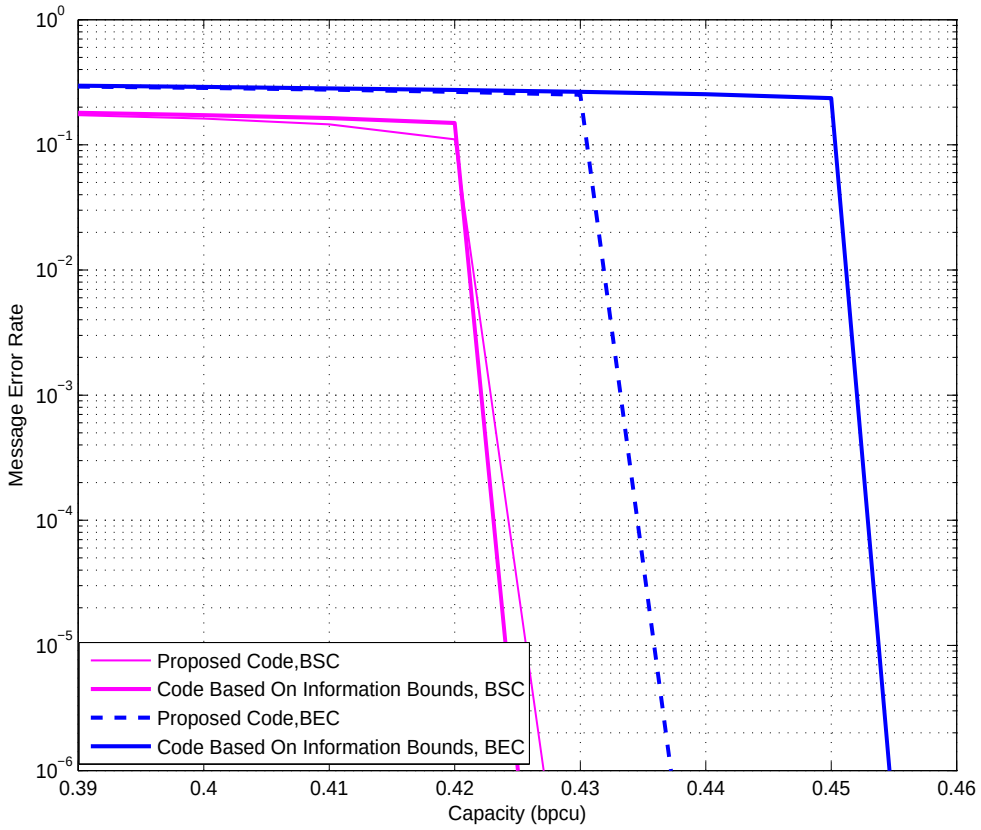

Figure 4.2: Comparison between message error rates of a rate 0.4 universal codes based on our proposed method and a rate 0.4 code based on the information combining bounds on the BSC and BEC channels. The curves are obtained by running density evolution for 400 iterations

are obtained with maximum allowed degree of 50 in their Tanner graph. It can been seen that our proposed code is much more efficient than other existing solutions.

Also, to see how successful our codes are, an upper bound is suggested to compare against. The upper bound here is the rate of LDPC codes designed to work on both BSC and BEC, which is the necessary (not sufficient) condition of universal codes. From Fig. 4.1, we can see our codes are quite close to the upper bound.

Other comparisons are also done between a rate 0.4 code based on both the information combining bounds and a rate 0.4 code based on the proposed method. In Fig. 4.2, a curve of decoder's message error obtained after 400 iterations of density evolution for these two codes on BEC and BSC channels is plotted. And Fig. 4.2.4

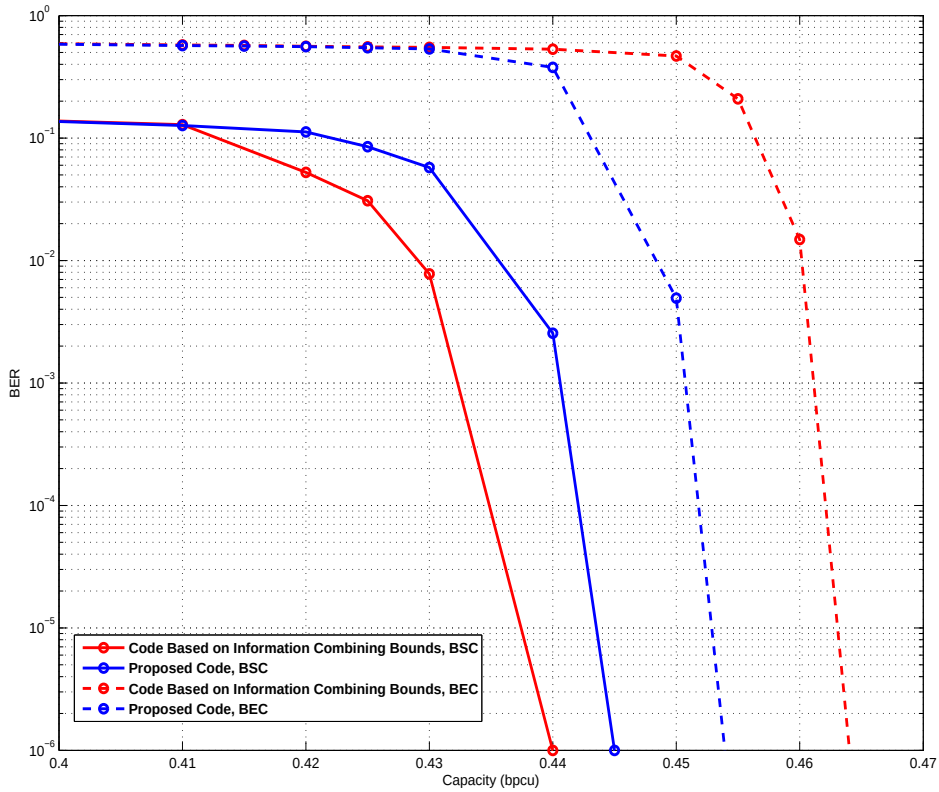

Figure 4.3: Comparison between BER of a rate 0.4 universal codes based on proposed method and a rate of 0.4 code based on the information bounds on the BSC and BEC channels. The curves are obtained for randomly constructed a code of length 76789

shows bit error rate comparisons between these codes for randomly constructed codes of length 76789. From both curves, we can see that although the code based on the information bounds performs slightly better on BSC, the code based on our proposed method has much stronger universal properties. Both codes are provided in Table 4.1, where we denote the codes designed by applying both information bounds as sufficient code.

Table 4.2 compares our codes in terms of their decoding threshold with codes designed for specific channels. By decoding threshold, we mean the worst capacity which a BEC, BSC, AWGN, or Rayleigh channel have to exhibit for successful convergence using density evolution. While AWGN codes are considered to have

Table 4.1: Degree distributions for proposed code and sufficient code

	Proposed Code	**Sufficient Code**
λ_2	0.14555	0.1436
λ_3	0.2970	0.2925
λ_5	0.0633	0.73
λ_7		0.0713
λ_9	0.1015	0.1654
λ_{10}	0.1539	0.73
λ_{18}		0.1069
λ_{50}	0.2388	0.2203
ρ_7	0.2539	
ρ_8	0.7461	1

Table 4.2: Comparison between the decoding threshold (in terms of required capacity) of LDPC codes of different rates.

Code Rate (bpcu)	**0.40**	**0.55**	**0.7**
Proposed LDPC Codes	0.43	0.59	0.73
LDPC Codes Designed for AWGN	0.51	0.62	0.76
LDPC Codes Designed for BEC	0.54	0.68	0.85
LDPC Codes Designed for BSC	0.46	0.61	0.75
LDPC Codes Designed for Rayleigh	0.48	0.65	0.78

good universal properties [58], Table 4.2 shows that our codes are much stronger in this sense. At lower channel capacities, e.g 0.4 bpcu, this improvement is even more pronounced. All the codes in Table 4.2 have a maximum degree of 30 in their Tanner graph and the AWGN codes are taken from [59].

4.2.6 Extreme Distributions Under Min-Sum Decoder

Current research on universal codes all focus on the sum-product decoder, however, designing universal code under min-sum decoder is also important due to its low complexity. As the pdf of the decoder messages at each iteration under min-sum decoding is not symmetric, the information combining bounds of [55, 56] are not applicable. Thus, finding the extreme distributions under min-sum decoder be-

comes very interesting. In this section, we deal with problem based on numerical results and we make the following two conjectures.

Conjecture 4.1 *Among all channels with capacity $\mathcal{C}$, min-sum decoding can achieve the highest rate on the BEC.*

Under BEC, LLRs are either 0 or infinity. As a result, according to the update rules introduced in Chapter 2, the output LLR at a check node of both min-sum and sum-product are the same. In other words, min-sum decoding does not incur any penalty on the BEC and is equivalent to sum-product (optimal) decoding. This supports the above conjecture.

Conjecture 4.2 *Among all channels with capacity $\mathcal{C}$, min-sum decoding achieves the smallest rate on the BSC.*

This conjecture is supported by the fact that when the absolute value of the LLR values processed at a check node are closer together the approximation becomes less accurate. On the BSC and in the early iterations, the absolute value of the LLRs are very close. In fact, in the first iteration, they are all equal.

Fig. 4.4 is also provided to support our conjectures.

4.3 Conclusion

In this chapter, we deal with the problem of designing universal LDPC codes.

A universal LDPC code is a code that is designed for a given capacity, independent of the actual channel model and guaranteed to converge on a multitude of channels. Using a refined Gaussian approximation on decoder messages and a known bound on information combining, LDPC codes with strong universal properties can be found. Over a wide range of rates, the suggested universal codes achieve a large percentage of the channel capacity. Our extensive tests have also verified successful convergence of these codes on all tested channels.

Based on numerical observations, we also proposed two conjectures regarding the extreme distributions under min-sum decoding.

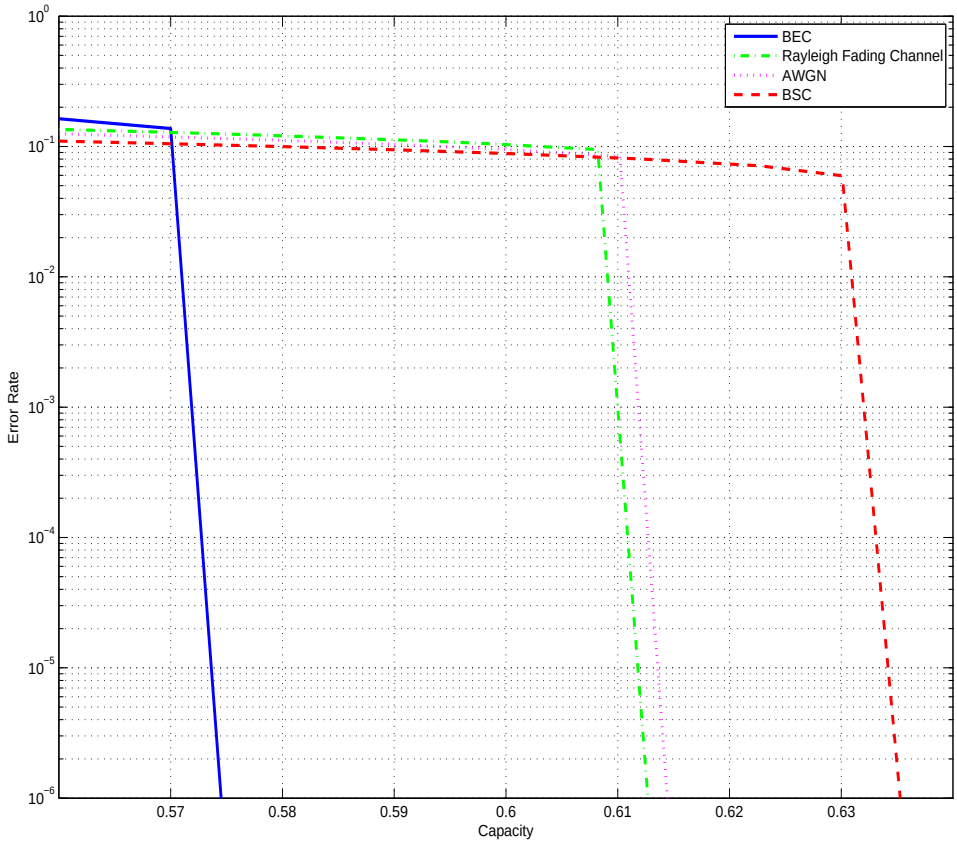

Figure 4.4: Comparison of (3,6)-regular LDPC code over different channels under min-sum decoding via density evolution.

Chapter 5

Conclusion

In this chapter, we first briefly summarize the contributions of this thesis and then propose some possible future research directions that may attract future researchers.

5.1 Contributions

In this thesis, several contributions have been made to the field of LDPC codes. We provide theoretical and practical results for the design and decoding of LDPC codes. The results can also be used for other codes decoded by iterative decoding, such as turbo codes.

In the first contribution, we studied the iterative decoding on non-SISO channels. An optimum and efficient piece-wise linear approximation is proposed to approximate the channel LLRs. This method is optimum in the sense of maximizing the achievable rate of the channel. Two different channel scenarios are considered. For the first scenario, we assume the channel state information is perfectly known at the receiver of MIMO-BICM channels. For the second scenario, two-way relay channels are considered when no channel state information is known. For both scenarios, we found optimized piece-wise linear approximation functions to calculate LLR. We also showed that the performance of our method is extremely close to the channel capacity where true LLRs are applied.

The second contribution made in this thesis is that we investigated the design method of a class of LDPC codes called universal codes. For a universal code, convergence is guaranteed for a given channel capacity and regardless of the chan-

nel type. A new universal code design method was proposed. We designed LDPC codes under this method and observed that a large percentage of the channel capacity can be achieved. Based on extensive observations, we also made conjectures for the extreme distributions under min-sum decoding.

5.2 Possible Future Research

In this section, we present some problems that can be the subject of future research.

The optimum piece-wise linear LLR calculation proposed in this work is probably the beginning of this research direction, since we only considered the scenarios under the binary phase shift keying modulation. Thus, extending this method to higher order modulations is an interesting direction.

The channels considered in Chapter 3 were all uncorrelated or flat fading channels. Possible future work can be done on extending this method to frequency-selective, correlated fading channels and orthogonal frequency division multiplexing (OFDM) systems.

It is also quite interesting if one can find the close form functions for the optimized parameters in our method, since right now we only consider optimization through numerical methods.

Since we made conjectures about the extreme distributions under min-sum decoding mainly based on numerical observations, the analytical proof of these two conjectures can be quite important for both theoretical and practical purposes. Other research directions can also include designing universal codes under min-sum decoding.

Bibliography

[1] C. E. Shannon, “A mathematical theory of communication,” *Bell System Technical Journal*, vol. 27, pp. 379–423 and 623–656, Jul./Oct. 1948.

[2] C. Berrou, A. Glavieux, and P. Thitimajshima, “Near shannon limit error-correcting coding and decoding:turbo-codes. 1,” in *Communications, 1993. ICC 93. Geneva. Technical Program, Conference Record, IEEE International Conference on*, vol. 2, Geneva, Switzerland, May 1993, pp. 1064–1070.

[3] R. M. Tanner, “A recursive approach to low complexity codes,” *IEEE Trans. Inf. Theory*, vol. 27, no. 5, pp. 533–547, Sep. 1981.

[4] N. Wiberg, “Codes and decoding on general graphs,” Ph.D. dissertation, Linköping University, Sweeden, 1996. [Online]. Available: http://www.it.isy.liu.se/publikationer/LIU-TEK-THESIS-440.pdf

[5] R. G. Gallager, *Low-Density Parity-Check Codes*. Cambridge, MA: MIT Press, 1963.

[6] *Digital Video Broadcasting (DVB)*, European Standard (Telecommunications series) Std. ETSI EN 302 307 V1.1.1, 2004.

[7] *Optional B-LDPC coding for OFDMA PHY*, IEEE Std. IEEE 802.16e-04/78, 2004.

[8] D. Divsalar, H. Jin, and R. J. McEliece, “Coding theorems for ‘turbo-like’ codes,” in *Proc. 36th Allerton Conference on Communications, Control, and Computing*, Monticello, Illinois, 1998, pp. 201–210.

[9] M. Luby, "LT codes," in *Proc. The 43rd Annual IEEE Symposium on Foundations of Computer Science*, Nov. 2002, pp. 271–280.

[10] A. Shokrollahi, "Raptor codes," *IEEE/ACM Transactions on Networking (TON)*, vol. 14, pp. 2551–2567, 2006.

[11] F. Tosato and P. Bisaglia, "Simplified soft-output demapper for binary interleaved cofdm with application to hiperlan/2," in *Communications, 2002. ICC 2002. IEEE International Conference on*, vol. 2, 2002, pp. 664 – 668 vol.2.

[12] P. Fertl, J. Jalden, and G. Matz, "Capacity-based performance comparison of MIMO-BICM demodulators," in *Signal Processing Advances in Wireless Communications, 2008. SPAWC 2008. IEEE 9th Workshop on*, 2008, pp. 166 –170.

[13] R. Yazdani and M. Ardakani, "Linear llr approximation for iterative decoding on wireless channels," *Communications, IEEE Transactions on*, vol. 57, no. 11, pp. 3278 –3287, 2009.

[14] ——, "Efficient LLR calculation for non-binary modulations over fading channels," *IEEE Trans. Commun., submitted for publication*, Nov. 2009.

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

[16] R. G. Gallager, *Information Theory and Reliable Communication*. New York, NY: John Wiley and Sons, 1968.

[17] T. Richardson and R. Urbanke, *Modern Coding Theory*, Oct. 2007. [Online]. Available: http://lthcwww.epfl.ch/mct/

[18] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. New York: Wiley, 1991.

[19] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 498–519, Feb. 2001.

[20] M. G. Luby, M. Mitzenmacher, M. A. Shokrollahi, and D. A. Spielman, "Improved low-density parity-check codes using irregular graphs," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 585–598, Feb. 2001.

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

[22] F. R. Kschischang and B. J. Frey, "Iterative decoding of compound codes by probability propagation in graphical models," *IEEE J. Sel. Areas Commun.*, vol. 16, no. 2, pp. 219–230, Feb. 1998.

[23] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Mateo, CA: Morgan Kaufmann Publishers, 1988.

[24] M. Ardakani and F. R. Kschischang, "Properties of optimum binary message-passing decoders," *IEEE Trans. Inf. Theory*, vol. 51, no. 10, pp. 3658–3665, Oct. 2005.

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

[26] A. Roumy, S. Guemghar, G. Caire, and S. Verdu, "Design methods for irregular repeat-accumulate codes," *IEEE Trans. Inf. Theory*, vol. 50, no. 8, pp. 1711–1727, August 2004.

[27] O. Etesami and A. Shokrollahi, "Raptor codes on binary memoryless symmetric channels," *Information Theory, IEEE Transactions on*, vol. 52, no. 5, pp. 2033 – 2051, may 2006.

[28] S.-Y. Chung, T. J. Richardson, and R. L. Urbanke, “Analysis of sum-product decoding of low-density parity-check codes using a Gaussian approximation,” *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 657–670, Feb. 2001.

[29] M. Ardakani and F. R. Kschischang, “A more accurate one-dimensional analysis and design of LDPC codes,” *IEEE Trans. Commun.*, vol. 52, no. 12, pp. 2106–2114, Dec. 2004.

[30] S. ten Brink, “Convergence behavior of iteratively decoded parallel concatenated codes,” *IEEE Trans. Commun.*, vol. 49, pp. 1727–1737, Oct. 2001.

[31] D. Divsalar, S. Dolinar, and F. Pollara, “Low complexity turbo-like codes,” in *Proc. 2nd International Symposium on Turbo Codes and Related Topics*, Brest, France, 2000, pp. 73–80.

[32] H. El Gamal and A. R. Hammons, “Analyzing the turbo decoder using the Gaussian approximation,” *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 671–686, Feb. 2001.

[33] M. Ardakani, T. H. Chan, and F. R. Kschischang, “EXIT-chart properties of the highest-rate LDPC code with desired convergence behavior,” *IEEE Commun. Lett.*, vol. 9, no. 1, pp. 52–54, Jan. 2005.

[34] M. Ardakani, “A linear-programming approach to the design of LDPC codes for non-uniform channels,” in *Proc. IEEE International Conference on Communications*, Istanbul, Turkey, 2006.

[35] M. G. Luby and M. Mitzenmacher, “Efficient erasure correcting codes,” *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 569–584, Feb. 2001.

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

[37] A. Sanderovich, M. Peleg, and S. Shamai, “LDPC coded MIMO multiple access with iterative joint decoding,” *Information Theory, IEEE Transactions on*, vol. 51, no. 4, pp. 1437 – 1450, april 2005.

[38] G. Caire, G. Taricco, and E. Biglieri, “Bit-interleaved coded modulation,” *Information Theory, IEEE Transactions on*, vol. 44, no. 3, pp. 927 –946, May 1998.

[39] B. Rankov and A. Wittneben, “Spectral efficient protocols for half-duplex fading relay channels,” *Selected Areas in Communications, IEEE Journal on*, vol. 25, no. 2, pp. 379 –389, february 2007.

[40] P. Popovski and H. Yomo, “Physical Network Coding in Two-Way Wireless Relay Channels,” in *Communications, 2007. ICC ’07. IEEE International Conference on*, june 2007, pp. 707 –712.

[41] A. Sanaei and M. Ardakani, “Ldpc code design considerations for non-uniform channels,” *Communications, IEEE Transactions on*, vol. 58, no. 1, pp. 101 –109, 2010.

[42] P. D.Tse, *Fundamentals of Wireless Communication*. Cambrigge Univ. Press, 2005.

[43] L. Zhao, J. Huber, and W. Gerstacker, “Design and analysis of bit interleaved coded space-time modulation,” *Communications, IEEE Transactions on*, vol. 56, no. 6, pp. 904 –914, 2008.

[44] S. Muller-Weinfurtner, “Coding approaches for multiple antenna transmission in fast fading and OFDM,” *Signal Processing, IEEE Transactions on*, vol. 50, no. 10, pp. 2442 – 2450, Oct. 2002.

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

[46] J. Boutros, F. Boixadera, and C. Lamy, "Bit-interleaved coded modulations for multiple-input multiple-output channels," in *Spread Spectrum Techniques and Applications, 2000 IEEE Sixth International Symposium on*, vol. 1, Sep. 2000, pp. 123 –126 vol.1.

[47] C. E. Shannon, "Two-way communication channels," *Proc. Fourth Berkeley Symp. on Math. Statist. and Prob., Vol. 1*, pp. 611–644, 1961.

[48] B. Rankov and A. Wittneben, "Achievable rate regions for the two-way relay channel," in *Information Theory, 2006 IEEE International Symposium on*, july 2006, pp. 1668 –1672.

[49] P. Larsson, N. Johansson, and K.-E. Sunell, "Coded bi-directional relaying," in *Vehicular Technology Conference, 2006. VTC 2006-Spring. IEEE 63rd*, vol. 2, may 2006, pp. 851 –855.

[50] C. Hausl and J. Hagenauer, "Iterative network and channel decoding for the two-way relay channel," in *Communications, 2006. ICC '06. IEEE International Conference on*, vol. 4, june 2006, pp. 1568 –1573.

[51] S. Zhang, Y. Zhu, S.-C. Liew, and K. B. Letaief, "Joint design of network coding and channel decoding for wireless networks," in *Wireless Communications and Networking Conference, 2007.WCNC 2007. IEEE*, march 2007, pp. 779 –784.

[52] J. Hagenauer, "Viterbi decoding of convolutional codes for fading-and-burst-channels," *Zurich Seminar Digital Commnun.*, 1980.

[53] F. Peng, W. Ryan, and R. Wesel, "Surrogate-channel design of universal LDPC codes," *IEEE Commun. Lett.*, vol. 10, no. 6, 2006.

[54] I. Sason and B. Shuval, "On universal LDPC code ensembles," in *Information Theory Proceedings (ISIT), 2010 IEEE International Symposium on*, 2010, pp. 689 –693.

[55] I. Land, S. Huettinger, P. Hoeherand, and J. Huber, “Bounds on information combining,” *IEEE Trans. Inf. Theory*, vol. 51, no. 2, pp. 612–619, 2005.

[56] I. Sutskover, S. Shamai, and J. Ziv, “Extremes of information combining,” *IEEE Trans. Inf. Theory*, vol. 51, no. 4, pp. 1313–1325, 2005.

[57] S.-Y. Chung, “On the construction of some capacity-approaching coding schemes,” Ph.D. dissertation, MIT, 2000.

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

[59] A. Amraoui, “LdpcOpt, optimization of the degree distrbutions of LDPC ensembles,” http://lthcwww.epfl.ch/research/ldpcopt/.