The ability to simplify means to eliminate the unnecessary so that the necessary may speak.

– Hans Hofmann

University of Alberta

Low Complexity Iterative Decoding Methods

by

Mahdi Ramezani

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

Edmonton, Alberta
Fall 2008

Library and Archives Canada

Published Heritage Branch

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque et Archives Canada

Direction du Patrimoine de l'édition

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence
ISBN: 978-0-494-47396-2
Our file Notre référence
ISBN: 978-0-494-47396-2

NOTICE:
The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

AVIS:
L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

Canada

To my family ...

Abstract

This thesis presents low complexity iterative decoding schemes based on the low-density parity-check (LDPC) codes for both single-user and multiuser environments. First, a low complexity method based on the min-sum decoder is proposed which leads to design of codes with close-to-capacity rates. We design LDPC codes based on the proposed method and show that, compared to the min-sum decoder and linear scaling min-sum decoder, higher code rates can be achieved at essentially no extra complexity. Second, we propose a method for communicating over a two-user Gaussian broadcast channel based on LDPC codes. Unlike the existing work, our method does not require joint decoding at the receivers and each user can use his own LDPC code. Then, we optimize LDPC codes based on the proposed method and show that the complexity of the code design stage is significantly reduced.

Acknowledgements

I would like to express my heartfelt gratitude to my supervisor, Dr. Masoud Ardakani, for being such a caring person. I want to thank him for the chance he gave me to learn a lot from him during uncountable hours that we spent together. I cannot imagine a supervisor more caring than Masoud.

During the last two years, I had the chance to work with exceptional people in our research team. My special thanks go to my colleagues, my best friends in Edmonton, Ali Sanaei, Moslem Noori, and Raman Yazdani who helped me a lot during many hours of discussion. They created an incredible environment for me to enjoy my research time.

I would like to thank my friends in Edmonton who made memorable moments for me. To name a few out of many: Ali Sharifkhani, Arash Talebi, Hooman Erfanian, Sina Ghaemi, Payam Dehghani, Emma Frontana, Sara Nobari, Alireza Ghaderipoor, Mohsen Eslami, Amirmasoud Rabiei, Mahdi Hajiaghayi and Reza Nikjah. Also, I am grateful to Bill and Rhyl Stollery, my first landlords in Edmonton, who spread their pure love to me during the last two years.

This thesis is dedicated to my family whose endless support from thousands of miles away made me feel them beside myself.

Table of Contents

List of Figures

List of Tables

List of Symbols

Symbol	Definition	First Use
$\mathcal{N}(m, \sigma^2)$	A Gaussian density with mean m and variance σ^2	3
$\mathcal{X}$	Support set for the variable X	7
$\Pr\{E\}$	Probability of the event E	8
$\mathbb{E}$	Expected value operator	9
$\mathbb{F}_2$	The binary field	11
$\mathbf{H}$	Parity-check matrix	11
$\sum_{\sim x_i}$	Summation over all variables except x_i	12
m	Generic symbol for LLR values	16
$\mathbb{1}$	The set indicator function	16
$\lambda(x), \rho(x)$	Left and right degree distributions of an LDPC ensemble	21
$\mathfrak{C}_n(\lambda, \rho)$	LDPC Ensemble of length n and with degree distributions $\lambda(x)$ and $\rho(x)$	22
$X \to Y \to Z$	A Markov chain by random variables X, Y, and Z	23
$\mathrm{a}_{\mathrm{ch}}(x)$	The channel LLR pdf	23
$\Delta_t(x)$	The Dirac delta function at $x = t$	23
$\bar{\epsilon}$	$1 - \epsilon$ for an $\epsilon \in [0, 1]$	23
$\mathcal{P}(\cdot)$	The error probability functional	26
$\mathcal{B}(\cdot)$	The Bhattacharyya functional	27
$\otimes$	Special convolution over $\mathbb{R}$ used at variable nodes	28
$\Phi_X(\theta)$	The cumulant generating function of the random variable X	36
CH	Convex hull	49

List of Abbreviations

Abbrv.	Definition	First Use
LDPC	Low-density parity-check	1
MAP	Maximum a posteriori	2
BSC	Binary symmetric channel	3
BEC	Binary erasure channel	3
BIAWGN	Binary-input additive white Gaussian noise	3
DMC	Discrete memoryless channel	7
pdf	Probability density function	7
ML	Maximum-likelihood	10
LLR	Log-likelihood ratio	16
BPSK	Binary phase shift keying	17
BISO	Binary-input symmetric-output	23
EXIT	Extrinsic information transfer	29
SP	Sum-product	31
MS	Min-sum	31
LSMS	Linear scaling min-sum	31
SNR	Signal-to-noise ratio	35
i.i.d.	Independent and identically distributed	36
CSI	Channel state information	46
BICM	Bit-interleaved coded modulation	47
CM	Coded modulation	53
PAM	Pulse amplitude modulation	55

Chapter 1

Introduction

The focus of the current thesis is on the designing and analyzing some low complexity coding schemes based on an extremely powerful class of error correcting codes called *low-density parity-check* (LDPC) codes. LDPC codes have been shown to be capable of working very close to the Shannon limit with a practical decoding complexity.

In this chapter, we briefly review the basic communications problem and introduce some challenging problems with LDPC codes which are the focus of this thesis.

1.1 Overview

Throughout the history, human put a lot of effort to communicate fast and reliably over long distances. In the nineteen century, Morse could send telegraph messages from Washington to Baltimore. In that time, telegraph operators were more likely to make an error when a message was sent quickly or the distance was too far. In fact, there was a tradeoff between the rate and reliability of transmission. The same analogy is applicable to today's communication systems.

Shannon in 1948 [1] developed a mathematical framework to quantify the information contained in a random source. Fig. 1.1 shows a general communication system where the ultimate goal is to transmit a data stream from a source (transmitter) to a sink (receiver) quickly and reliably. A source encoder is used to transform the source outputs to a stream which usually consists of bits. The source encoder is used because there are dependencies among source outputs. In fact, the task of the source encoder is to remove this redundant information (redundant rate) and represent the source data with the minimum number of bits.

Then, the channel encoder adds some redundancy to this bit stream and passes

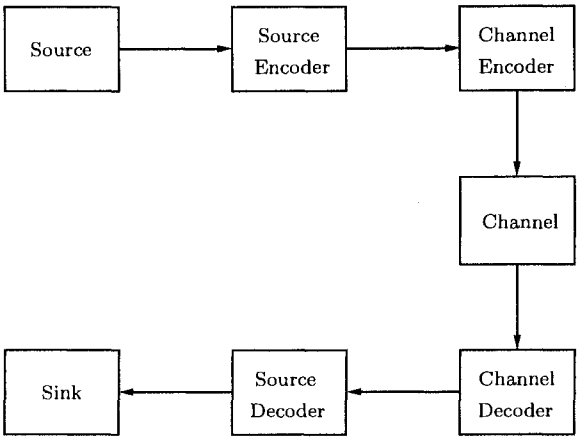


Figure 1.1: A communication system.

it across a noisy environment called *channel* which distorts the stream in different possible ways. The reason for adding redundancy is to combat the noise of the channel. The *rate* of a channel encoder is defined as the ratio of the number of input bits to the number of output bits which is always less than one. Depending on the application, various type of channel encoders may be used. In this thesis, we are interested in the *block codes* where the input bit stream is partitioned into several k-bit blocks and each block is mapped to an n-bit word ($n > k$) called a *codeword*. There are $n - k$ redundant bits and the code rate will be $R = \frac{k}{n} < 1$. There are 2^k possible input blocks leading to 2^k codewords of length n which are called the *codebook*.

Upon receiving the distorted stream, the channel decoder removes the redundancy and tries to recover data bits with the minimum probability of error. Finally, the source decoder maps the estimated stream to a sink data stream. There are many channel decoders that can be used to recover the data bits among which we are interested in the *maximum a posteriori* (MAP) decoders and the *iterative decoders*. A MAP decoder minimizes the posteriori probability of the transmitted signal based on the observation at the output of the channel. On the other hand, an iterative decoder uses a class of algorithms called *message passing* algorithms which are powerful from the complexity point of view. Basically, an iterative decoder has two constituent decoders such that a message or belief about the transmitted codeword is passed between them in order to improve the reliability of the estimation. We shall discuss iterative decoders more in Section 1.2.

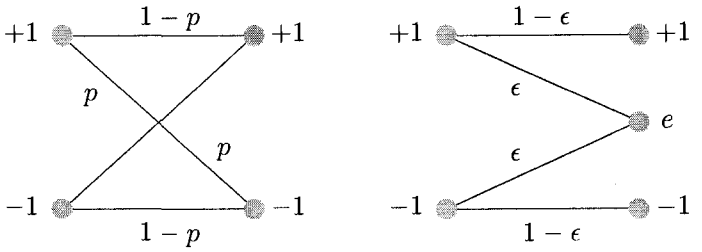


Figure 1.2: Left: The binary symmetric channel with error probability p, BSC(p). Right: The binary erasure channel with erasure probability ϵ, BEC(ϵ).

Given a channel, Shannon showed that there is a limit on the maximum code rate by which the reliable transmission is possible and called it the *channel capacity.* He showed that there exists a channel encoder-decoder pair that can be used to transmit data with arbitrarily small probability of error provided that the code rate is below the capacity.

For more illustration, consider the transmission over a channel which flips every bit with the probability of p as shown in Fig. 1.2. This channel is called the binary symmetric channel (BSC). One way to combat the noise of the BSC is to send each bit multiple times. Then at the receiver one can count the number of $+1$'s and -1's and decide that the symbol with a larger frequency was transmitted. By, let say N, repetitions of a single bit, the code rate will be $R = \frac{1}{N}$. More reliability can be achieved with more repetitions, however, we lose the rate of transmission by increasing N.

Another important channel model, which mostly occurs in the data networks, is when a bit is erased with probability of ϵ. This channel model is called the binary erasure channel (BEC) and is shown in Fig. 1.2. Also, the output of a channel can be continuous. The binary-input additive white Gaussian noise (BIAWGN) channel adds a random real number to the binary input. The additive noise is drawn according to a zero-mean Gaussian density with variance σ^2, i.e., $\mathcal{N}(0, \sigma^2)$.

By channel coding, we mean the channel encoder-decoder pair. Our goal in channel coding is that we help the transmission to make it as reliable as possible while maintaining the rate as close as possible to the channel capacity. The simplest channel coding method that we have already used in our illustrating example is called repetition coding.

Shannon in his fundamental paper [1] used a random codebook to demonstrate

the existence of a capacity-achieving channel code. In practice, however, we need a description of how we should embed information bits in a codeword, otherwise we should store the whole codebook. For example, a codebook of rate 0.8 with codewords of length 500 has 2^{400} codewords which is not possible to be stored in a memory. Therefore, random coding cannot be used in practice. Block codes are a powerful class of practical codes which are used in conjunction with the iterative decoders throughout this thesis.

Shannon showed that for memoryless channels, i.e., channels where the channel output only depends on the channel input at that time, the problem of source coding and channel coding can be solved separately. The focus of the current thesis is on the channel coding part, hence we assume that the source coding problem has been efficiently solved.

1.2 Codes on Graphs and Iterative Decoding

In this section, the importance of iterative decoding is highlighted.

Given a stationary memoryless channel, for any code rate less than the capacity and any $\varepsilon > 0$, the existence of a coding system which results in an average decoder error probability less than ε was proved by Shannon. As it was pointed out, he used random coding which needs infinite computational capability which is not applicable in practice. One important practical issue that we are concerned about is the encoding and decoding complexity.

Let us briefly describe the practical importance of the block codes when they are used along with an iterative decoder rather than a MAP decoder. To do so, assume that we are willing to achieve at least a fraction of $1-\delta$ of the capacity where $\delta \to 0$. Let $\Upsilon_{\mathrm{E}}(\delta)$ and $\Upsilon_{\mathrm{D}}(\delta)$ denote the encoding and decoding complexity normalized per information bit. For block codes under a MAP decoder and any fixed probability of error, it has been shown that[1] [2]

$$\Upsilon_{\mathrm{E}}(\delta) = O(1/\delta^2) \text{ and } \Upsilon_{\mathrm{D}}(\delta) = 2^{O(1/\delta^2)}.$$

However, for a block code with an iterative decoder (message passing decoder), it is conjectured that

$$\Upsilon_{\mathrm{E}}(\delta) = \Upsilon_{\mathrm{D}}(\delta) = O(1/\delta).$$

[1] We say a function $f(n)$ is $O(g(n))$ if there exists a constant k such that for all sufficiently large $n \in \mathbb{N}$, $|f(n)| \leq k|g(n)|$. Also, we say a function $f(n)$ is $\Theta(g(n))$ if there exist constants k and k' such that $k'|g(n)| \leq |f(n)| \leq k|g(n)|$ for all sufficiently large $n \in \mathbb{N}$.

According to this conjecture, the complexity per information bit grows *linearly* with $1/\delta$ whereas it grows exponentially with $1/\delta^2$ for the MAP decoder [2].

A coding system with rates close to the capacity was not developed until 1993 when *turbo codes* were invented [3]. Later, the LDPC codes were rediscovered by groups from two different communities [4–6] and [7–10]. Originally, LDPC codes were invented by Gallager in his PhD thesis [11] and have been forgotten for almost three decades. LDPC codes are block codes which can be represented on a graph called Tanner graph and has a sparse structure. The importance of LDPC codes is that under the message passing algorithms, they exhibit a linear-time decoding complexity with the code length n.

The constituent decoders for an LDPC code are variable nodes and check nodes. We will formally describe them in Section 2.3. The message passing decoder is called the *sum-product* decoder when it is used to obtain the optimal bitwise decoding (see Section 2.2.2). It has been shown that LDPC codes under the sum-product decoding, if carefully designed, can operate extremely close to the channel capacity [12].

In this thesis, we tackle two problems: first, we consider a suboptimal class of message passing decoders called the min-sum decoder which has a lower complexity than the sum-product decoder, but at the expense of some performance degradation. We try to improve the performance of LDPC codes under min-sum decoding and make it as close as possible to the sum-product decoder. Second, a broadcasting scenario is considered for which it has been shown in the literature that the LDPC codes can be used to achieve close-to-capacity rates. However, the previous work is quite complex and needs joint decoding at each of the receivers. We seek a low complexity method for broadcasting using LDPC codes.

1.3 Thesis Outline

This thesis is organized as follows: Chapter 2 reviews preliminary materials about iterative decoding, factor graphs, optimal decoding and LDPC codes.

It has been shown that under min-sum decoding, scaling the messages at the output of check nodes can improve the performance of a certain class of LDPC codes [13]. However, for highly optimized LDPC codes designed for the sum-product decoder, linear scaling can hinder the performance. The problem of code design for the min-sum and linear scaling min-sum (LSMS) decoders have been investigated in [14]. It is shown that the gap to the capacity for LSMS codes is better than the

min-sum codes, but compared to sum-product codes the gap is still considerable. In Chapter 3, a modified min-sum decoding is proposed and studied. We use the stability analysis of density evolution to show that the proposed method allows for a larger code rate. Finally, by designing codes based on the modified method, we show that compared to the min-sum and LSMS codes, a smaller gap to the capacity can indeed be achieved while the complexity of decoding remains essentially the same.[2]

In Chapter 4, we will use LDPC codes for communication over a two-user Gaussian broadcast channel. We assume that each user has its own LDPC code. It is shown in [16] that the optimal decoding of such system requires both users to have the code of each other and a joint decoding of both user messages is needed at each user. Also, a joint code design procedure should be performed. We propose a method which uses a novel labeling strategy and the bit-interleaved coded modulation. This method significantly reduces the code design complexity and does not require joint decoding. For different rate pairs on the boundary of the capacity region, a pair of LDPC codes are designed to demonstrate the results[3].

We conclude the thesis in Chapter 5 by summarizing the contributions of this thesis and by suggesting possible future research directions.

[2]The results of this chapter have been published in [15].

[3]The results of this chapter are being prepared for submission to IEEE Trans. on Commun.

Chapter 2

Iterative Decoding

The preliminary materials about iterative decoding are presented in this chapter. First, we will see some fundamental results on reliable information transmission. In Section 2.2, factor graphs and message passing algorithms which are powerful tools in the context of iterative decoding are presented. Finally, Section 2.3 introduces decoding analysis and design of LDPC codes.

2.1 Reliable Transmission of Information

As it was stated in Chapter 1, we are willing to transmit data from a source to a destination as reliably and quickly as possible. However, there exists a tradeoff between the reliability and the rate of transmission. The transmission takes place on a channel which distorts the transmitted signal. Finding a communication scheme for reliable information transmission has been a big challenge for decades. In this section, we formally define the channel coding system shown in Fig. 1.1. Let us start with the definition of a discrete channel.

Definition 2.1 [Discrete Memoryless Channel (DMC) [17]]: A discrete channel $(\mathcal{X}, p(y|x), \mathcal{Y})$ is defined as a system comprising a finite input alphabet $\mathcal{X}$, a finite output alphabet $\mathcal{Y}$, and a collection of conditional probability density functions (pdf) $p(y|x)$. A channel is said to be *memoryless* if the probability of observing an output symbol depends only on the current input symbol and is conditionally independent of previous channel inputs or outputs, i.e.,

$$p(y^n|x^n) = \prod_{i=1}^{n} p(y_i|x_i)$$

where $x^n = x_1 \cdots x_n$.

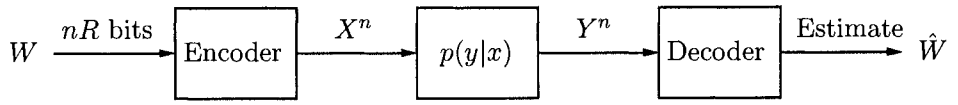


Figure 2.1: Transmission model.

Definition 2.2 [Channel Code [17]]: A $(2^{nR}, n)$ channel code where R is the code rate in bits/channel use consists of:

1. An equiprobable message set $\mathcal{W} = \{1, 2, \ldots, M\}$ where $M = 2^{nR}$
2. A codebook $\mathfrak{C}_n(R) = \{x^n(w) \in \mathcal{X}^n | w \in \mathcal{W}\}$ which has M codewords of length n and symbols from the input alphabet $\mathcal{X}$
3. A decoder which assigns a message index $\hat{w}(y^n) \in \mathcal{W}$ to each of the received observation y^n.

Fig. 2.1 shows the communication model over a DMC. To transmit a message, each $k = nR$ input information bits is mapped to an index $w \in \mathcal{W}$. Then the corresponding codeword $x^n(w)$ is transmitted over the channel which is equivalent to n independent uses of channel. Based on the observation y^n, the receiver decodes y^n to a message index $\hat{w}(y^n) \in \mathcal{W}$. An error is occurred when $\hat{w}(y^n) \neq w$. Also, the average probability of error is

$$\mathrm{P}_e^{\mathrm{avg}}(n) = \frac{1}{M} \sum_{w \in \mathcal{W}} \Pr\{\hat{w}(y^n) \neq w\}.$$

A rate R is said to be *achievable*, if there exists a sequence of $(2^{nR}, n)$ channel codes such that

$$\limsup_{n \to \infty} \mathrm{P}_e^{\mathrm{avg}}(n) = 0.$$

The *capacity* of a channel is the supremum of all the achievable rates. Shannon in his landmark paper [1] showed the following theorem:

Theorem 2.1 [DMC Capacity [1]]: For a DMC, the information capacity is

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

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. ▼

For constructing a "good" code, a probabilistic approach is usually used. Using a random process, an ensemble of codes is generated and one proves that with a close-to-one probability, all codes are "good". This code construction processes is known as *random coding* and is used by Shannon in [1]. By random coding, each symbol in a codeword is drawn according to the input density $p(x)$ and the average probability of error is calculated by averaging over all the codewords in a codebook and all possible codebooks. Using the random coding and the jointly-typical decoding [17, 18], Shannon proved another fundamental theorem:

Theorem 2.2 [SHANNON CODING THEOREM [1]]: Given a DMC, for any code rate $R < C$ and any $\varepsilon > 0$, there exits an encoder-decoder pair which permits the transmission of information over the channel at rate R and average decoder error probability less than ε.

Conversely, for any rate $R > C$, no matter which encoder-decoder pair is being used, the average probability of error will be bounded away from zero, i.e.,

$$\liminf_{n\to\infty} \mathbb{E}_{\mathfrak{C}_n(R)}(\mathrm{P}_e^{\mathrm{avg}}(n)) > 0$$

where the average is taken over all the codewords in a codebook and all possible codebooks.

▼

More precisely, according to Wolfowitz [19], for any DMC with capacity C and any $(2^{nR}, n)$ code where $R > C$,

$$\mathrm{P}_e^{\mathrm{avg}}(n) \geq 1 - \frac{4A}{n(R-C)^2} - e^{-\frac{n}{2}(R-C)}$$

where A depends only on the channel but not on n and R. Therefore, as the code length gets larger, the average probability of error approaches one!

In fact, Theorem 2.2 ensures that every randomly generated codebook with $R < C$ is likely to exhibit vanishing probability of error as the code length gets larger.

2.1.1 MAP and ML Decoding

Consider that we are willing to communicate over a channel $(\mathcal{X}, p(y|x), \mathcal{Y})$ with a code from the ensemble

$$\mathfrak{C}_n(R) = \{x^n(w) \in \mathcal{X}^n | w \in \mathcal{W}\}.$$

We choose an index w and transmit the codeword $x^n(w)$ over the channel. Upon observing the channel output y^n, the decoder outputs an estimate $\hat{w}(y^n)$. The probability of making an error given an observation y^n will be

$$\Pr\{\text{error}|y^n\} = 1 - p(\hat{w}(y^n)|y^n).$$

In order to minimize the probability of error, we define the MAP detection rule as:

$$\begin{aligned}\hat{w}^{\text{MAP}}(y^n) &= \arg\max_{w\in\mathcal{W}} \; p(w|y^n) \\ &= \arg\max_{w\in\mathcal{W}} \; p\big(y^n|x^n(w)\big)p(w)\end{aligned}$$

which maximizes the posteriori probability of the transmitted codeword based on the channel output observation. If the codewords are chosen uniformly, we have

$$\begin{aligned}\hat{w}^{\text{MAP}}(y^n) &= \arg\max_{w\in\mathcal{W}} \; p\big(y^n|x^n(w)\big)p(w) \\ &= \arg\max_{w\in\mathcal{W}} \; p\big(y^n|x^n(w)\big) = \hat{w}^{\text{ML}}(y^n)\end{aligned}$$

which is called the *maximum-likelihood* (ML) detection rule. In fact, under uniform priori assumption, the MAP and ML detection rules are the same.

Hereafter, we consider the uniform selection of codewords which means that

$$p(w) = \frac{1}{M}\;, \forall w \in \mathcal{W}.$$

It is noteworthy that it should not be mistaken with the case where the channel input symbols are uniformly distributed.

The seminal work of Shannon is remarkable by finding the maximum reliable information rate over a DMC. In practice, however, using random coding is not possible because we need to have a description of the codewords to embed information bits in them, or we need to store them in the memories of the transmitter and receiver. For example, for a code with reasonable length $n = 500$ and rate $R = 0.8$ bits/channel use, we need $M = 2^{400}$ codewords to be stored which is quite larger than the whole number of particles in the world! However, if there exists a description of how one should embed data bits into a codeword then there is no need to store all the codewords.

2.1.2 Linear Block Codes

It has been a challenging problem for decades to find a practical coding solution with reasonable encoding and decoding complexity. Linear block codes are among

the richest channel codes which have practical complexity. Let $\mathbb{F}_2$ denote the binary field. Linear codes are defined as:[1]

Definition 2.3 [Binary Linear Codes [20]]: A $(2^k, n)$ channel code is a binary linear code if and only if the 2^k codewords form a k-dimensional subspace of the vector space of all the n-tuples over $\mathbb{F}_2$.

The term linear is used since all the codewords are closed under addition and multiplication operations in $\mathbb{F}_2$. Since a linear code is a k-dimensional subspace of $\mathbb{F}_2^n$, there are k vectors of length n by which the code is spanned. More precisely, let

$$\mathbf{u} = [u_1, u_2, \ldots, u_k]^{\mathrm{T}} \in \mathbb{F}_2^k$$

be the vector of information bits. We call

$$\mathbf{G} = [\mathbf{g}_1, \mathbf{g}_2, \ldots, \mathbf{g}_k]^{\mathrm{T}} \in \mathbb{F}_2^{k \times n}$$

the *generator* matrix whose rows are the basis vectors of the code subspace. Since each codeword belongs to the subspace spanned by $\{\mathbf{g}_i\}_{i=1}^k$, we have

$$\begin{aligned} \mathbf{x} &= \mathbf{G}^{\mathrm{T}}\mathbf{u} \\ &= \sum_{i=1}^{k} u_i \mathbf{g}_i \end{aligned}$$

where $\mathbf{x} \in \mathbb{F}_2^n$ and $\mathbf{u}$ takes all the 2^k values to generate the codebook. Therefore, an n-tuple vector is a codeword if and only if it can be written as a linear combination of $\{\mathbf{g}_i\}_{i=1}^k$. Another representation for a linear code stems from the null space concept. In fact, $\mathbb{F}_2^n$ can be decomposed into two orthogonal subspaces such that each vector in one subspace is orthogonal to all vectors in the other subspace. Thus, an n-tuple vector $\mathbf{x}$ is a codeword if and only if $\mathbf{Hx} = \mathbf{0}$ where

$$\mathbf{H} = [\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_{n-k}]^{\mathrm{T}} \in \mathbb{F}_2^{(n-k) \times n}$$

is called the *parity-check* matrix which consists of $n - k$ vectors spanning the null space. In fact, all the rows of $\mathbf{G}$ are orthogonal to the rows of $\mathbf{H}$, i.e., $\mathbf{GH}^{\mathrm{T}} = \mathbf{0}$. We denote a linear code with the parity-check matrix $\mathbf{H}$ by

$$\mathfrak{C}_n(R, \mathbf{H}) = \{\mathbf{x} \in \mathbb{F}_2^n | \mathbf{Hx} = \mathbf{0}\} = \mathrm{Ker}\{\mathbf{H}\} \tag{2.1}$$

where the last equation shows that a linear code is the kernel subspace of its parity-check matrix in $\mathbb{F}_2^n$.

[1] In this work, we are interested in the binary linear block codes where the encoder splits the sequence of information bits into k-bit partitions and map them into codewords of length n.

2.2 Factor Graphs

A large number of computational problems in signal processing, communications and artificial intelligence can be solved efficiently by using factor graphs [2]. In this section, we are interested in the problem of optimal decoding of linear codes over a DMC[2].

The key point in the factor graphs is the distributive law, i.e., $\sum_{i,j} a_i b_j = \sum_i a_i \sum_j b_j$ where a_i's and b_j's belong to an arbitrary field. Consider a function $f(x_1, \ldots, x_6)$ which has a factorization of the form

$$f(x_1, \ldots, x_6) = f_1(x_1, x_2, x_3) f_2(x_1, x_4, x_6) f_3(x_4) f_4(x_4, x_5). \tag{2.2}$$

Let $\sum_{\sim x_j}$ denote the summation over all the variables except x_j. Computing the marginal of the variable x_1, i.e.,

$$f(x_1) = \sum_{\sim x_1} f(x_1, \ldots, x_6)$$

for all values of x_1 needs $\Theta(|\mathcal{X}|^6)$ operations where $\mathcal{X}$ is the support set for variables $x_1, x_2, \ldots, x_6$ and $|\mathcal{X}|$ denotes the cardinality of the set $\mathcal{X}$. However, according to the distributive law, this marginal computation can be done using

$$f(x_1) = \Big[\sum_{x_2, x_3} f_1(x_1, x_2, x_3) \Big] \Big[\sum_{x_4} f_3(x_4) \sum_{x_6} f_2(x_1, x_4, x_6) \sum_{x_5} f_4(x_4, x_5) \Big], \; x_1 \in \mathcal{X},$$

requiring $\Theta(|\mathcal{X}|^3)$ operations which is considerably less than $\Theta(|\mathcal{X}|^6)$.

Any such factorization can be represented by a bipartite[3] graph, called the factor graph. We show variables with circles and factors with squares. The *degree* of a particular node is the number of edges connected to that node. Fig. 2.2 shows the factor graph of the function $f(x_1, \ldots, x_6)$ defined in (2.2).

Now, let us look at a more general case where the underlying factor graph is a tree. Consider marginalization of a function g with respect to a variable z. The function g has a generic factorization as

$$g(z, \ldots) = \prod_{k=1}^{K} g_k(z, \ldots)$$

[2]Some examples and notations of this section are taken from [2]. The reader is referred to [21] and [22] for a comprehensive study of factor graphs.

[3]A graph is called bipartite if the set of vertices can be partitioned into two disjoint sets and every edge connects a vertex from one set to one in the other set.

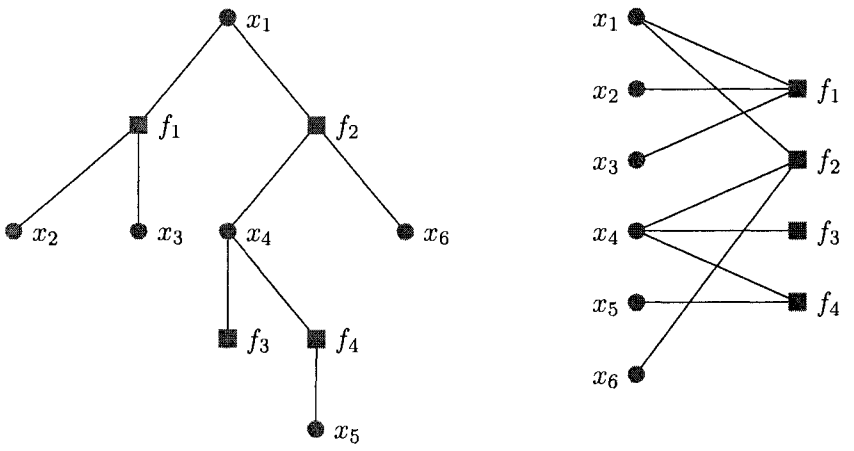


Figure 2.2: Left: factor graph associated to the function $f(x_1, \ldots, x_6)$ given in (2.2). Right: The same graph rearranged as a bipartite graph.

where the variable z appears in all factors, but all other variables are present in only one factor in order to maintain a tree. This factorization is depicted in Fig. 2.3. Moreover, each g_k must have a factorization as

$$g_k(z, \ldots) = \underbrace{s_k(z, z_1, \ldots, z_J)}_{\text{kernel}} \prod_{j=1}^{J} t_j(z_j, \ldots)$$

where z appears only in the *kernel* s_k and each of the z_j appears at most twice, possibly in the kernel and in at most one of the factors t_j [2]. Therefore, we get

$$\begin{aligned}
g(z) &= \sum_{\sim z} g(z, \ldots) \\
&= \sum_{\sim z} \prod_{k=1}^{K} g_k(z, \ldots) \\
&\overset{(a)}{=} \prod_{k=1}^{K} \sum_{\sim z} g_k(z, \ldots) \\
&= \prod_{k=1}^{K} \underbrace{\sum_{\sim z} s_k(z, z_1, \ldots, z_J) \prod_{j=1}^{J} t_j(z_j, \ldots)}_{\text{operation of the function node blown up in Fig. 2.3}} \\
&= \prod_{k=1}^{K} \sum_{\sim z} s_k(z, z_1, \ldots, z_J) \underbrace{\prod_{j=1}^{J} \sum_{\sim z_j} t_j(z_j, \ldots)}_{\text{product of marginals [similar to } (a)\text{]}}
\end{aligned} \tag{2.3}$$

which shows that the marginalization over a tree can be done recursively by breaking it down into smaller tasks until we reach the leaves of the tree. The marginalization

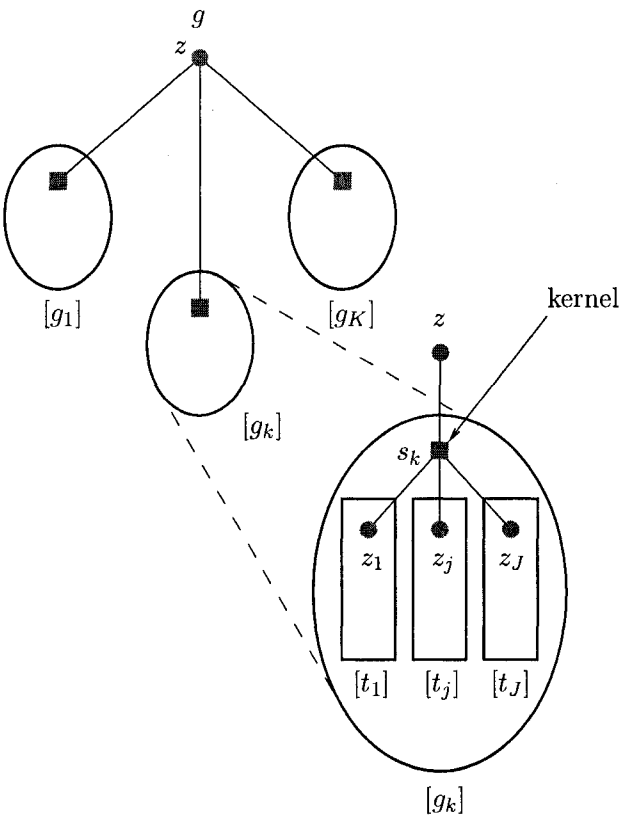


Figure 2.3: Recursive computation over a factor graph. The factor t_j has the same generic form as the factor g.

process starts from the leaves of the tree. As soon as a node receives all its marginals, which are functions over $\mathcal{X}$, from all its children, it processes and passes it to its parents.

Remark 2.1: It is worth mentioning that the marginalization algorithm does not depend on which node is the root of the tree. Thus, marginalization with respect to all the variables can be done with a single tree. ▲

2.2.1 Message Passing Algorithm

When factor graph is a tree, the marginalization process can be done through an efficient algorithm called *message passing* algorithm. In this case, marginals are messages or beliefs. For this reason, sometimes this algorithm is called belief propagation. Message passing algorithm is an iterative decoding algorithm where the output message sent along a particular edge of a node depends only on the messages

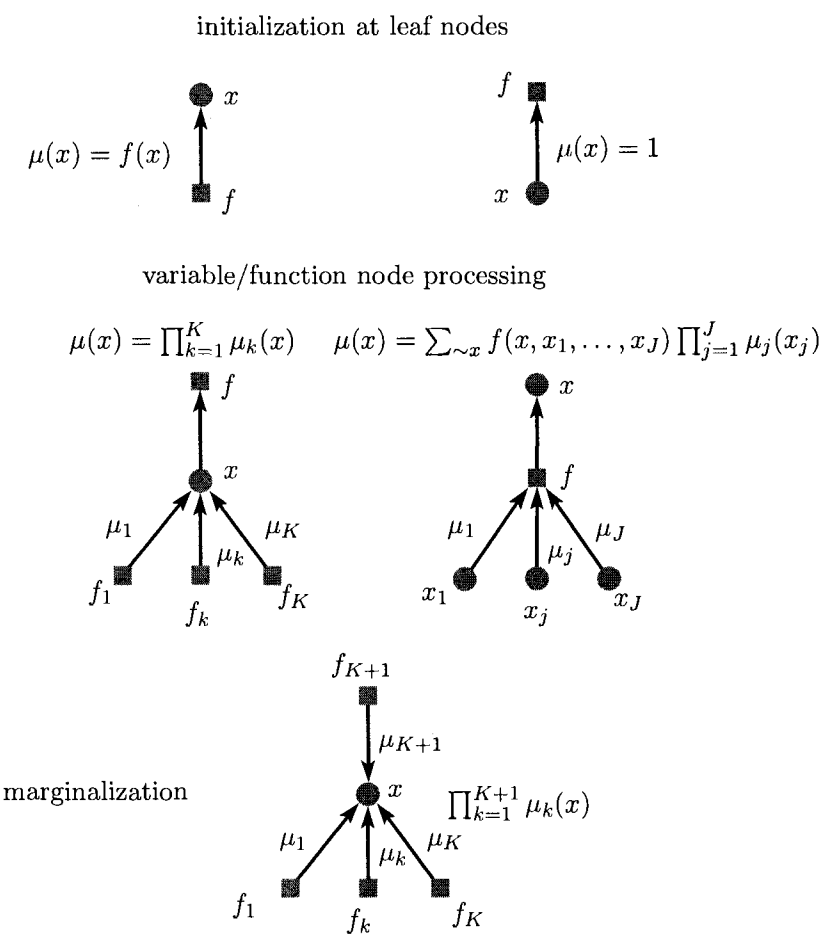


Figure 2.4: Message passing rules (reproduced from [2]).

received along all other edges. The nodes in the tree compute messages, which are functions over $\mathcal{X}$ and pass them to the next level [2].

According to (2.3), a variable node computes the pointwise multiplication of the incoming messages and a function node, which is blown up in Fig. 2.3, processes the incoming messages according to (2.3). A summary of the message passing algorithm is shown in Fig. 2.4 where $\mu(x)$, $x \in \mathcal{X}$ denotes the message that is to be sent out.

In this work, we are dealing with $\mathcal{X} = \{\pm 1\}$ by which we can simplify the message passing algorithm. For binary alphabet, a message is in the form of $[\mu(+1), \mu(-1)]$. Let us define the *likelihood ratio* as

$$r = \frac{\mu(+1)}{\mu(-1)}$$

and its logarithm as the *log-likelihood ratio* (LLR) given by

$$m = \log \frac{\mu(+1)}{\mu(-1)}.$$

For a variable node of degree $K+1$, since we have

$$\mu(x) = \prod_{k=1}^{K} \mu_k(x), \ x \in \mathcal{X},$$

it is straightforward to see that the message that is to be sent out to its parent function node is

$$r_{v \to f} = \prod_{k=1}^{K} r_k$$

or in the LLR domain

$$m_{v \to f} = \sum_{k=1}^{K} m_k. \tag{2.4}$$

We will see later that working with LLR values is much easier than the plain likelihood values.

Let $\mathbb{1}_{t \in \mathcal{T}}$ denote an indicator function which is one when the element t belongs to the set $\mathcal{T}$ and is zero otherwise. In Section 2.2.2, we will encounter function nodes which show parity-check equations. It is shown in [2] that the message emanating from a function node of degree $J+1$ with the kernel function

$$f(x, x_1, \ldots, x_J) = \mathbb{1}_{\prod_{j=1}^{J} x_j = x}$$

to its parent variable node is

$$m_{f \to v} = 2 \tanh^{-1} \left[\prod_{j=1}^{J} \tanh \left(\frac{m_j}{2} \right) \right] \tag{2.5}$$

where v indicates the variable node for x. In (2.5), the fact that

$$\frac{r-1}{r+1} = \tanh(m/2)$$

has been used.

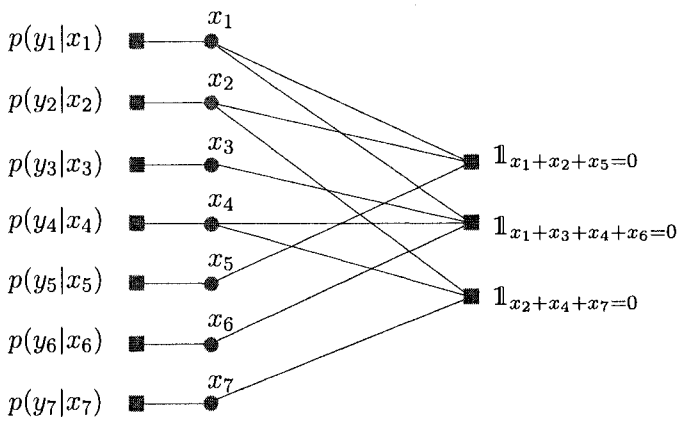


Figure 2.5: Factor graph for the code $\mathfrak{C}_7(4/7, \mathbf{H})$ given in Example 2.1.

2.2.2 Optimal Bitwise Decoding

Under the uniformly chosen codeword assumption, the bitwise MAP detection rule for a linear code $\mathfrak{C}_n(R, \mathbf{H})$ will be[4]

$$\begin{aligned}
\hat{x}_i^{\text{MAP}}(\mathbf{y}) &= \arg \max_{x_i \in \{\pm 1\}} \; p(x_i|\mathbf{y}) \qquad i = 1, 2, \ldots, n \\
&= \arg \max_{x_i \in \{\pm 1\}} \; \sum_{\sim x_i} p(\mathbf{x}|\mathbf{y}) \\
&= \arg \max_{x_i \in \{\pm 1\}} \; \sum_{\sim x_i} p(\mathbf{y}|\mathbf{x}) p(\mathbf{x}) \\
&= \arg \max_{x_i \in \{\pm 1\}} \; \sum_{\sim x_i} \left(\prod_{j=1}^{n} p(y_j|x_j) \right) \mathbb{1}_{\mathbf{x} \in \mathfrak{C}_n(R,\mathbf{H})} \\
&= \arg \max_{x_i \in \{\pm 1\}} \; \sum_{\sim x_i} \left(\prod_{j=1}^{n} p(y_j|x_j) \right) \left(\prod_{j=1}^{n-k} \mathbb{1}_{\mathbf{h}_j^{\mathrm{T}} \mathbf{x} = 0} \right) \qquad (2.6)
\end{aligned}$$

which can be read as a marginalization task of a function with factors in (2.6) with respect to x_i. Let us take a look at an example:

Example 2.1: Consider a rate 4/7 linear code with the following parity-check matrix:

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

The factor graph associated to (2.6) is a tree and is shown in Fig. 2.5. Each function node corresponding to one row of $\mathbf{H}$ is called a check node. Messages are in the

[4]We assume that the binary phase shift keying (BPSK) signaling is used, i.e., $0 \mapsto +1$ and $1 \mapsto -1$.

form of probabilities, e.g., $\mu = [p(y|X = +1), p(y|X = -1)]$ which can be shown by an LLR value. In fact, the sing of LLR shows the hard estimate while its magnitude reveals the reliability of decision.

All incoming LLR messages are processed by check nodes and then the resulting messages are sent back to variable nodes. These messages are processed by variable nodes and are sent back to check nodes. This is one iteration of message passing. The message passing paradigm is shown in Fig. 2.6. Since there is no message from check nodes at iteration $\ell = 0$, the message that is sent from variable node v, corresponding to the ith bit of the codeword, to check node c, i.e., $m_{v\to c}^{(0)}$ is the message from the channel (m_i). At iteration $\ell > 0$, $m_{v\to c}^{(\ell)}$ is the summation of messages from channel and all the check nodes except c, i.e., m_i, m_1', m_2', and m_3'. Also, check node c outputs $m_{c\to v}^{(\ell)}$ according to (2.5).

In this procedure, there are two sources of information:

• Intrinsic message: the message from the channel to the variable node v which only depends on y_i, i.e.,

$$m_i = \log \frac{p(y_i|X_i = +1)}{p(y_i|X_i = -1)}.$$

• Extrinsic messages: the message from previous iteration, i.e., $m_{v\to c}^{(\ell)}$ and $m_{c\to v}^{(\ell)}$.

In other words, at each iteration, message passing decoder combines these two pieces of information to get information about the transmitted codeword. After, say L iterations, a decision at variable node v is made using

$$\hat{x}_i = \text{sign}\{m_0 + \sum_c m_{c\to v}^{(L)}\}$$

where the summation is over all the check nodes connected to v. ◊

Remark 2.2: In Example 2.1, since the factor graph is a tree, the message passing algorithm will give the exact MAP estimation of the variables given the observed vector $\mathbf{y} = [y_1, \ldots, y_7]^{\mathrm{T}}$. However, when the underlying factor graph is not a tree, the message passing algorithm becomes suboptimal. It is shown that for the sparse graphs with cycles, message passing still performs very well [23, 24]. ▲

2.2.3 General Semirings and Optimum Block Decoding

All factor graph computations are valid for an arbitrarily field $\mathbb{F}$. However, all that was needed was actually working on a commutative semiring $\mathbb{K}$. Similar to a

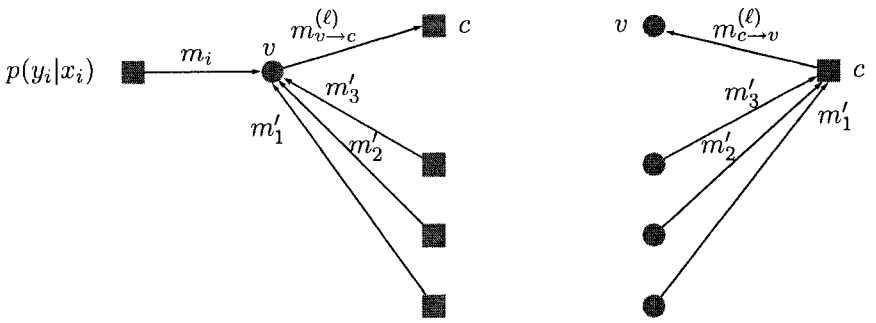


Figure 2.6: Message passing over the factor graph of the code in Example 2.1.

field, there are two operations, "+" and "×", in a semiring. The difference between a commutative semiring and a field is that we do not require the existence of an inverse with respect to either operation [2]. In the context of iterative decoding, we are interested in three important semirings listed in Table 2.1, where $(+, 0)$ and $(\times, 1)$ are two commutative semigroups with identity elements 0 and 1, respectively. The *sum-product* semiring that we have used until now for the bitwise MAP decoding given in (2.6) is the most important semiring. We shall see that the *min-sum* semiring is useful when we are dealing with optimum block decoding. Also, the Boolean algebra is used in binary message passing algorithms such as GallagerA and GallagerB [11] which are not the focus of the current thesis.

The optimum block decoding can be formulated as a marginalization problem. To this end, it is important to note that by marginalization of a function $f(x_1, x_2)$ over the min-sum semiring, we mean

$$f(x_1) = \min_{x_2} f(x_1, x_2) = \min_{\sim x_1} f(x_1, x_2).$$

Back to the optimum block decoding problem, assuming an equiprobable selection of codeword, we have

$$\begin{aligned}
\hat{\mathbf{x}}^{\mathrm{MAP}}(\mathbf{y}) &= \arg\max_{\mathbf{x}} \; p(\mathbf{x}|\mathbf{y}) \\
&= \arg\max_{\mathbf{x}} \; p(\mathbf{y}|\mathbf{x})p(\mathbf{x}) \\
&= \arg\max_{\mathbf{x}} \; \left(\prod_{j=1}^{n} p(y_j|x_j) \right) \mathbb{1}_{\mathbf{x}\in\mathfrak{C}_n(R,\mathbf{H})}.
\end{aligned}$$

One can compare detection of one bit in optimal block decoding with the bitwise

Table 2.1: Commutative semirings used in the iterative decoding context.

Name	$\mathbb{K}$	$(+,0)$	$(\times,1)$
Sum-Product	$\mathbb{R}^+$	$(+,0)$	$(\times,1)$
Min-Sum	$\mathbb{R}\cup\{+\infty\}$	$(\min,+\infty)$	$(+,0)$
Boolean	$\{0,1\}$	$(\mathrm{OR},0)$	$(\mathrm{AND},1)$

MAP decoding given in (2.6) as

$$\begin{aligned}\left[\hat{\mathbf{x}}^{\mathrm{MAP}}(\mathbf{y})\right]_i &= \arg\min_{x_i\in\{\pm 1\}}\min_{\sim x_i} -\left(\prod_{j=1}^{n} p(y_j|x_j)\right)\left(\prod_{j=1}^{n-k}\mathbb{1}_{\mathbf{h}_j^{\mathrm{T}}\mathbf{x}=0}\right) \qquad i=1,2,\ldots,n\\ &= \arg\min_{x_i\in\{\pm 1\}}\min_{\sim x_i} -\sum_{j=1}^{n}\log p(y_j|x_j) - \sum_{j=1}^{n-k}\log\mathbb{1}_{\mathbf{h}_j^{\mathrm{T}}\mathbf{x}=0}. \qquad (2.7)\end{aligned}$$

The only difference between (2.7) and (2.6) is that addition and multiplication operations are replaced with minimization and addition, respectively. Therefore, optimal block decoding can be performed using the min-sum algebra. Note that the term $\log\mathbb{1}_{\mathbf{h}_j^{\mathrm{T}}\mathbf{x}=0}$ acts as the identity element of the $(\times,1)$ semigroup in the min-sum semiring providing that the jth parity-check equation is not satisfied.

Similar to the message passing rules in (2.4) and (2.5) over the sum-product semiring, Wiberg [25] proved that the update rules using the min-sum algebra for a degree–$(K+1)$ variable node and degree–$(J+1)$ function node are

$$m_{v\to f} = \sum_{k=1}^{K} m_k$$

and

$$m_{f\to v} = \min_{j\in\{1,\ldots,J\}}\{|m_j|\}\prod_{j=1}^{J}\mathrm{sign}\{m_j\}, \qquad (2.8)$$

respectively. We will use computations over the min-sum semiring in Chapter 3.

2.3 Low-Density Parity-Check Codes

In this section, some basic background materials about LDPC codes needed for Chapter 3 and Chapter 4 are presented.

Motivated by the null space representation given in (2.1), a convenient way to describe a linear code is by showing it on a graph. To this end, consider a linear code $\mathfrak{C}_n(k/n,\mathbf{H})$ used for transmission over a binary-input memoryless channel $p(y|x)$. In Example 2.1, we have seen the factor graph associated with the bitwise (or blockwise)

MAP decoding of a code (see (2.6) and (2.7)). Let $\mathcal{G}$ be a bipartite graph with n left variable nodes and $n-k$ right function nodes called *check nodes*. The graph $\mathcal{G}$ which is exemplified by Fig. 2.5, is called *Tanner* graph. The codewords are those vectors associated with the variable nodes such that the parity-check equations at the check nodes are satisfied. In fact, each check node represents one row of the parity-check matrix $\mathbf{H}$. The binary entries of $\mathbf{H}$ indicates whether there is an edge between a pair of check node and variable node. The degree of a variable (check) node is the number of check (variable) nodes connected to it. Since for each code, there are many parity-check representations, there are many Tanner graphs for a given code. Although, they all represent the same code, they exhibit different performances.

LDPC codes are linear block codes with at least one sparse parity-check matrix. The sparsity property causes LDPC codes to have "good" performance under message passing decoding and exhibit linear-time decoding complexity [26].

LDPC codes can be either *regular* or *irregular*. A (d_v, d_c) regular LDPC code is a code such that each variable node has degree d_v and each check node has degree d_c. We call an LDPC code irregular when the node degrees are chosen according to some distribution.

Consider an LDPC code of length n and design rate $R = \frac{k}{n}$. Let λ_i and ρ_i show the fraction of edges connected to degree-i variable nodes and degree-i check nodes, respectively. It is customary to put all these coefficients into two polynomials as

$$\lambda(x) = \sum_{i=2}^{d_v^{\max}} \lambda_i x^{i-1}$$

and

$$\rho(x) = \sum_{i=2}^{d_c^{\max}} \rho_i x^{i-1}$$

where $d_v^{\max}$ and $d_c^{\max}$ are the maximum variable node and check node degrees. We call these two polynomials the *left and the right degree distributions* from the edge perspective, respectively[5]. Clearly, for a (d_v, d_c) regular LDPC code, we have $\lambda(x) = x^{d_v-1}$ and $\rho(x) = x^{d_c-1}$. It is obvious that we should have

$$\lambda(1) = \sum_i \lambda_i = 1$$

and

$$\rho(1) = \sum_i \rho_i = 1.$$

[5]Similarly, we can define degree distributions from the node perspective. However, through this work, we use the edge perspective representation.

Also, we can show that

$$\int_0^1 \lambda(x)dx = \sum_i \frac{\lambda_i}{i} = \frac{n}{E} \tag{2.9}$$

and

$$\int_0^1 \rho(x)dx = \sum_i \frac{\rho_i}{i} = \frac{n-k}{E}$$

where E is the total number of edges. Thus, the design rate is given by

$$R(\lambda,\rho) = 1 - \frac{\int_0^1 \rho(x)dx}{\int_0^1 \lambda(x)dx}. \tag{2.10}$$

Note that if the rows of the parity-check matrix are not linearly independent then the code rate R will be at least $R(\lambda,\rho)$. However, we assume that the parity-check matrix is of full rank hereafter, i.e., $R = R(\lambda,\rho)$.

Let $\mathfrak{C}_n(\lambda,\rho)$ denote an *ensemble* of LDPC codes of length n and degree distribution pair (λ,ρ)[6]. To construct an ensemble $\mathfrak{C}_n(\lambda,\rho)$, we place n variable nodes at left and $n(1-R)$ check nodes at right where R is given by (2.10). The number of edges needed for making a bipartite graph with degree distributions (λ,ρ) is given by (2.9). Let index all edges by the set

$$\mathcal{E} = \{1,\ldots,E\}.$$

Then, we randomly form a bipartite graph according to (λ,ρ) using all the edges in $\mathcal{E}$. By running all the permutations equally likely over $\mathcal{E}$, we will get the ensemble $\mathfrak{C}_n(\lambda,\rho)$. Therefore, each graph is picked uniformly random from the ensemble. The parity-check matrix (code) associated to each graph is formed in the way that each entry is one if the corresponding variable and check nodes are connected to each other an odd number of times [2]. This ensemble is usually called the standard LDPC ensemble.

2.3.1 Decoding Analysis

As it was pointed out in Section 2.2.1, message passing algorithms form an efficient way to compute the MAP decoding of a binary linear code over a memoryless channel. In order to analyze the performance of an ensemble of LDPC codes, we need to statistically analyze the message passing decoder. To illustrate, consider that

[6]We drop the argument x to simplify the notation.

we are willing to communicate over a channel using an ensemble of LDPC codes, $\mathfrak{C}_n(\lambda, \rho)$, whose rate is less than the capacity of the channel. We have two sources of randomness; one is the random messages (in the LLR domain) that are observed from the channel (due to different realization of the noise) and the other one is the LDPC code instance that we pick at random from the ensemble. In what follows, we discuss both of these sources.

Random Channel Messages

Consider that a binary codeword $\mathbf{X}$ is transmitted over a binary-input memoryless channel and $\mathbf{Y}$ is the resulting observation at the receiver. Let $\mathbf{M}(\mathbf{Y}) = [M_1, M_2, \ldots, M_n]$ be the LLR vector corresponding to $\mathbf{Y}$ where

$$M_i = \log \frac{p(Y_i|X_i = +1)}{p(Y_i|X_i = -1)}.$$

It is straightforward to see that $\mathbf{X}$, $\mathbf{Y}$, and $\mathbf{M}$ form a Markov chain $\mathbf{X} \rightarrow \mathbf{Y} \rightarrow \mathbf{M}$ and $\mathbf{M}$ is a sufficient statistic for estimating $\mathbf{X}$ given $\mathbf{Y}$, i.e., $I(\mathbf{X}; \mathbf{M}|\mathbf{Y}) = 0$. Throughout this work, we consider binary-input symmetric-output (BISO) memoryless channels which are defined as:

Definition 2.4 [BISO Channel [27]]: A binary-input memoryless channel where $\mathcal{X} = \{\pm 1\}$ and $\mathcal{Y} \subseteq \mathbb{R}$, is said to be symmetric if

$$p(y|x = -1) = p(-y|x = +1).$$

We will denote the pdf of

$$M = \log \frac{p(Y|X = +1)}{p(Y|X = -1)}$$

by $\mathrm{a}_{\mathrm{ch}}(m)$. Under the all-one codeword assumption, it is shown that for a BISO channel and for every m [27]

$$\mathrm{a}_{\mathrm{ch}}(-m) = e^{-m}\mathrm{a}_{\mathrm{ch}}(m) \tag{2.11}$$

which means that one side of the pdf can be obtained from the other side. Every such pdf satisfying (2.11) is called symmetric. Let us take a look at an example:

Example 2.2: Assume that the all-one codeword is transmitted. The LLR pdf of a BEC(ϵ), BSC(p) and BIAWGN(σ) are listed in Table 2.2 where $\Delta_t(x)$ is the Dirac delta function at $x = t$ and for an $\epsilon \in [0, 1]$, $\bar{\epsilon}$ stands for $1 - \epsilon$. It is easy to verify

Table 2.2: The LLR pdf for a BEC(ϵ), a BSC(p), and a BIAWGN(σ) channels under the all-one codeword transmission assumption where $r = \frac{2}{\sigma^2}$.

Possible LLR Values	$\mathrm{a}_{\mathrm{ch}}(x)$
$m = \{0, +\infty\}$	$\mathrm{a}_{\mathrm{BEC}(\epsilon)}(m) = \epsilon\Delta_0(m) + \bar{\epsilon}\Delta_\infty(m)$
$m = \{-\log\frac{\bar{p}}{p}, +\log\frac{\bar{p}}{p}\}$	$\mathrm{a}_{\mathrm{BSC}(p)}(m) = p\Delta_{-\log\frac{\bar{p}}{p}}(m) + \bar{p}\Delta_{+\log\frac{\bar{p}}{p}}(m)$
$m = \frac{2}{\sigma^2}y, m \in (-\infty, +\infty]$	$\mathrm{a}_{\mathrm{BIAWGN}(\sigma)}(m) = \frac{1}{\sqrt{4\pi r}}\exp\left\{-\frac{(m-r)^2}{4r}\right\}$

that all pdfs are symmetric, i.e., they satisfy (2.11)[7]. For a BIAWGN(σ) channel, we have

$$m = \log\frac{\frac{1}{\sqrt{2\pi\sigma^2}}e^{-\frac{(y-1)^2}{2\sigma^2}}}{\frac{1}{\sqrt{2\pi\sigma^2}}e^{-\frac{(y+1)^2}{2\sigma^2}}} = \frac{2}{\sigma^2}y$$

which leads to the pdf in Table 2.2. ◊

Over a BISO channel, if the message passing decoder being used satisfies the following conditions then the performance of such decoder is independent of the transmitted codeword [28].

• **Check node symmetry:** Consider a check node c of degree d_c, with input LLR messages $m_1, \ldots, m_{d_c-1}$. The LLR message from c to a variable node v at any iteration $\ell > 0$ should obey

$$m^{(\ell)}_{c\to v}(b_1 m_1, \ldots, b_{d_c-1}m_{d_c-1}) = m^{(\ell)}_{c\to v}(m_1, \ldots, m_{d_c-1})\prod_{i=1}^{d_c-1} b_i$$

for $b_i \in \{\pm 1\}$, $i = 1, 2, \ldots, d_c - 1$.

• **Variable node symmetry:** Consider a degree-d_v variable node v with an input LLR m_0 from the channel and input LLR messages $m_1, \ldots, m_{d_v-1}$. The LLR message from v to a check node c at any iteration $\ell > 0$ should satisfy

$$m^{(\ell)}_{v\to c}(-m_0, -m_1, \ldots, -m_{d_v-1}) = -m^{(\ell)}_{v\to c}(m_0, m_1, \ldots, m_{d_v-1})$$

and for $\ell = 0$

$$m^{(0)}_{v\to c}(-m_0) = -m^{(0)}_{v\to c}(m_0).$$

In this work, since we are interested only in the sum-product and min-sum decoders and they both fulfill the above conditions, we always assume that the all-

[7] In this work, for the sake of simplicity, we will drop the arguments of LLR pdfs and Dirac delta functions when there is no confusion.

one codeword was transmitted. Therefore, the range of LLR values that we use is $(-\infty, +\infty]$ since from (2.11), we have $\mathrm{a_{ch}}(-\infty) = 0$.

Random Code

Another source of randomness is the code instance that is picked from the ensemble. The question is that how different LDPC codes in an ensemble behave. In the following theorem, we see that with a high probability which grows exponentially fast with the block length, every randomly chosen LDPC code from the ensemble $\mathfrak{C}_n(\lambda, \rho)$ behaves close to the ensemble average.

Theorem 2.3 [CONCENTRATION AROUND ENSEMBLE AVERAGE [2, 28]]: Let $\mathfrak{C}$, a randomly chosen code from the ensemble $\mathfrak{C}_n(\lambda, \rho)$, be used for transmission over a BISO channel with LLR pdf $\mathrm{a_{ch}}$. Define $\mathrm{P_b^{MP}}(\mathfrak{C}, \mathrm{a_{ch}}, \ell)$ as the bit error probability after ℓ rounds of message passing decoder. Then, for any given $\delta > 0$, there exists an $\alpha > 0$, $\alpha = \alpha(\lambda, \rho, \delta)$ such that

$$\Pr\left\{\left|\mathrm{P_b^{MP}}(\mathfrak{C}, \mathrm{a_{ch}}, \ell) - \mathbb{E}_{\mathfrak{C}_n(\lambda,\rho)}(\mathrm{P_b^{MP}}(\mathfrak{C}, \mathrm{a_{ch}}, \ell))\right| > \delta\right\} \leq e^{-\alpha n}.$$

▼

Therefore, we will try to design an ensemble whose average performance is close to the Shannon limit, i.e., the code rate is close to the capacity of the underlying channel.

2.3.2 Functionals over Symmetric Densities

There are three important functionals over symmetric densities. For a BISO channel $\mathrm{a_{ch}}$, since the channel is symmetric, the optimal input density is uniform over $\mathcal{X} = \{\pm 1\}$ [17]. Also, since M is a sufficient statistic for estimating X given Y, the

capacity functional will be

$$\begin{aligned}
\mathcal{C}(\mathrm{a_{ch}}) &= \max_{p(x)} I(X;Y) = I(X;Y) = I(X;M) \\
&= H(X) - H(X|M) \\
&= 1 - \mathbb{E}\left(p(x)p(m|x)\log_2\left[\frac{1}{p(x|m)}\right]\right) \\
&= 1 - \frac{1}{2}\mathbb{E}\left(p(m|x)\log_2\left[\frac{\sum_{x\in\mathcal{X}} p(x)p(m|x)}{p(x)p(m|x)}\right]\right) \\
&= 1 - \int \mathrm{a_{ch}}(m)\log_2\left[\frac{\mathrm{a_{ch}}(m)+\mathrm{a_{ch}}(-m)}{\mathrm{a_{ch}}(m)}\right]dm \\
&= \mathbb{E}\big(1 - \log_2[1+e^{-m}]\big)
\end{aligned}$$

where we used the fact that $\mathrm{a_{ch}}(m) = p(m|x=+1)$. The error probability of MAP detection over a BISO channel when $x=+1$ is transmitted is given by $\mathcal{P}(\cdot)$ functional as

$$\begin{aligned}
\mathcal{P}(\mathrm{a_{ch}}) &= \Pr\big\{p(x=+1|y) < p(x=-1|y)\big\} + \frac{1}{2}\Pr\big\{p(x=+1|y) = p(x=-1|y)\big\} \\
&= \Pr\big\{m(y) < 0\big\} + \frac{1}{2}\Pr\big\{m(y)=0\big\} \\
&= \int_{-\infty}^{0^-} \mathrm{a_{ch}}(m)dm + \frac{1}{2}\int_{0^-}^{0^+} \mathrm{a_{ch}}(m)dm.
\end{aligned}$$

For a symmetric channel, $\mathcal{P}(\mathrm{a_{ch}})$ can be written as

$$\mathcal{P}(\mathrm{a_{ch}}) = \frac{1}{2}\int \mathrm{a_{ch}}(m)e^{-(\frac{m}{2}+|\frac{m}{2}|)}dm = \frac{1}{2}\mathbb{E}\big(e^{-(\frac{m}{2}+|\frac{m}{2}|)}\big).$$

In Chapter 3, we will frequently use another functional which gives us an upper bound on the error probability. Again, suppose that $x=+1$ is transmitted. For a positive s, according to the Chernoff bound, we have

$$\Pr\big\{m(y) < 0\big\} \le \mathbb{E}(e^{-sm}).$$

The Bhattacharyya functional is defined as the minimum Chernoff upper bound by

$$\begin{aligned}
\mathcal{B}(\mathrm{a_{ch}}) &= \inf_{s>0}\ \mathbb{E}(e^{-sm}) \\
&= \inf_{s>0}\ \int \mathrm{a_{ch}}(m)e^{-sm}dm.
\end{aligned}$$

For a symmetric channel, we get

$$\begin{aligned}
\mathcal{B}(\mathrm{a_{ch}}) &= \inf_{s>0} \int \mathrm{a_{ch}}(m) e^{-sm} dm \\
&= \inf_{s>0} \frac{1}{2} \int \left[\mathrm{a_{ch}}(m) e^{-sm} + \mathrm{a_{ch}}(-m) e^{sm}\right] dm \\
&= \inf_{s>0} \int \mathrm{a_{ch}}(m) e^{-\frac{m}{2}} \cosh m\big(s - 1/2\big) dm \\
&= \int \mathrm{a_{ch}}(m) e^{-\frac{m}{2}} dm \\
&= \mathbb{E}(e^{-\frac{m}{2}}).
\end{aligned}$$

The Bhattacharyya functional arises in many other two-class decision problems to bound the probability of error [29]. The following lemma shows the extremes of the Bhattacharyya functional.

Lemma 2.1 [EXTREMES OF THE BHATTACHARYYA FUNCTIONAL [2]]**:** For a fixed symmetric density $\mathrm{a}(x)$, we have

$$2\mathcal{P}(\mathrm{a}) \leq \mathcal{B}(\mathrm{a}) \leq 2[\mathcal{P}(\mathrm{a})\overline{\mathcal{P}(\mathrm{a})}]^{\frac{1}{2}}$$

where the left (right) side is tight for a BEC (BSC).

The three functionals $\mathcal{C}(\cdot)$, $\mathcal{P}(\cdot)$ and $\mathcal{B}(\cdot)$ can be used for any symmetric density, not necessarily LLR pdfs.

2.3.3 Density Evolution

Density evolution is presented by Richardson and Urbanke in [27,28] for asymptotic analysis of message passing decoders over BISO channels. It is also used for other codes defined on graphs under iterative decoding [30–33].

It is shown that a randomly chosen edge in a given Tanner graph spans a tree up to any fixed depth, with probability which approaches one as the code length gets large (local-tree property) [28]. This means that as the code length gets large, the random messages at the input of variable nodes and check nodes become independent. In density evolution, we assume that the code length is large enough such that the random messages are independent at each iteration. Therefore, the pdf of messages at each iteration is numerically (in some cases analytically) tractable which means that the performance of a given instance of LDPC code over a channel can be determined asymptotically.

At the ℓth iteration, let us define a_ℓ and b_ℓ as the pdf of random messages which are passed from variable (check) nodes to check (variable) nodes. We assume that the block length is large enough to have the local-tree property. Under sum-product decoding, since the input messages to a variable node are independent, according to (2.4), the output pdf is the convolution of the input pdfs. Define $\otimes$ as the convolution operator over $\mathbb{R}$ with special consideration when input pdfs contain Δ_∞ [27]. For check nodes, let $\boxtimes$ be a particular convolution defined in [27] which gives the updated pdf by check nodes. Then, the density evolution under sum-product decoding is given by

$$\mathrm{a}_\ell = \mathrm{a}_{\mathrm{ch}} \otimes \sum_i \lambda_i \mathrm{b}_\ell^{\otimes(i-1)} \tag{2.12}$$

where

$$\mathrm{b}_\ell = \sum_i \rho_i \mathrm{a}_{\ell-1}^{\boxtimes(i-1)},$$

and $\otimes(i-1)$ and $\boxtimes(i-1)$ mean $i-1$ times corresponding convolution of a density with itself. In (2.12), a_0 can be any symmetric density, however, most of the times it is the channel LLR pdf. We will use

$$\mathrm{a}_\ell = \mathrm{a}_{\mathrm{ch}} \otimes \lambda(\rho(\mathrm{a}_{\ell-1}))$$

as a shorthand for (2.12). For a BISO channel, it is shown in [27] that under sum-product decoding, all a_ℓ and b_ℓ densities are symmetric and $\mathcal{P}(\mathrm{a}_\ell)$ is a non-increasing function of ℓ. Successful decoding means that the pdf of random messages evolves to the zero-error pdf, i.e.,

$$\lim_{\ell\to\infty} \mathrm{a}_\ell = \Delta_\infty.$$

There are barely cases that one can use (2.12) to obtain the pdf of messages at each iteration. Moreover, (2.12) is computationally complex. Chung in his PhD thesis [34] and also [12] proposed a quantized version of (2.12) called *discrete density evolution* which quantizes LLR values and replaces pdfs with probability mass functions. For the purpose of code design and performance analysis in the current work, we will use discrete density evolution as a powerful and quite accurate method.

Since computing the exact density evolution is a complex task, there has been some approximations in the literature. Chung *et al.* in [35] assume that all extrinsic messages are Gaussian. In [36], Ardakani and Kschischang consider that only messages from variable nodes to check nodes are Gaussian. These methods have

considerably lower complexity than the exact density evolution. In the literature, density evolution is extended for asymmetric channels [37] and non-binary LDPC codes [38].

2.3.4 EXIT Chart and Code Optimization

A fast and efficient method for asymptotic analysis of iterative decoders is using extrinsic information transfer (EXIT) chart. This method which was first introduced by ten Brink [39, 40] is similar to the density evolution method; however, it tracks the evolution of one parameter associated to the pdf of random messages. This parameter can be entropy, error probability, or any other meaningful parameter. However, entropy is usually used as the most faithful parameter [2]. The evolution of the selected parameter is visualized on a chart called EXIT chart. We will show that an LDPC code optimization problem using EXIT chart method can be formulated as a linear program which can be efficiently solved.

In the EXIT chart method, at each iteration of density evolution, we replace the output pdf by a pdf from a family of pdfs which has the same entropy. This family can be a symmetric Gaussian pdf which obeys the constraints in Example 2.2 [35]. Let us define $\phi_i(H_0, h)$, the elementary EXIT curve for degree-i variable nodes, as the output entropy of a degree-i variable node which has two arguments: the channel entropy $H_0 = 1 - \mathcal{C}(\mathrm{a_{ch}})$ and the output entropy of the previous iteration, i.e., $h = 1 - \mathcal{C}(\mathrm{a}_{\ell-1})$. The condition for successful decoding is that

$$\sum_i \lambda_i \phi_i(H_0, h) < h, \ \forall h \in [0, H_0]$$

which is a linear constraint on the design parameters $\{\lambda_i\}_{i\geq 2}$. Now, assume that the right degree distribution $\rho(x)$ is fixed. Given a maximum left degree $d_v^{\max}$, according to (2.10), it suffices to maximize $\sum_i \frac{\lambda_i}{i}$ to obtain the maximum code rate which is a linear cost function. Therefore, given a BISO channel, one can write a linear optimization problem in $\{\lambda_i\}_{i\geq 2}$ variables to design a code with the highest rate as

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

Further optimizations on $\rho(x)$ will get better results, however, the performance of LDPC codes is not too sensitive to $\rho(x)$. Suggestions and guidelines for choosing $\rho(x)$ are given in [27, 41]. During this thesis, we will use this procedure to optimize LDPC ensembles in order to have close-to-capacity code rates.

Chapter 3

Modified Linear Scaling Min-Sum Decoder

3.1 Introduction

Min-sum (MS) decoder is the second most attractive decoder for LDPC codes, because compared to a sum-product (SP) decoder, it has substantially lower complexity at the expense of some performance degradation. Moreover, for an LDPC code with large length for which the local-tree assumption holds, SP decoder gives the optimal bitwise decoding (see Section 2.2.2), while MS decoder results in the optimal block decoding (see Section 2.2.3). Both SP and MS decoders exhibit the *threshold phenomenon* where the probability of error vanishes when the channel parameter, e.g., noise power in the Gaussian channel, falls below a value called threshold.

Therefore, improving the performance of MS decoder attracted much attention. As a case in point, it has been shown that for regular LDPC codes under MS decoding, if the messages are scaled down properly, the performance in terms of the gap to the capacity is improved [13,42]. However, we will see that for irregular LDPC codes designed for SP decoder, which have very small gap to the capacity, scaling the messages may deteriorate the performance even worse than the MS decoder.

Authors in [14] design LDPC codes based on MS and linear scaling min-sum (LSMS) decoders. They show that LSMS codes exhibit smaller gaps to the capacity than MS codes, however, the gap is still large compared to SP codes.

In this chapter, we propose a low complexity modified MS decoder which allows for higher code rates and hence smaller gaps to the capacity. The motivation behind the proposed decoder comes from the stability condition theorems which will be discussed in Section 3.2.

In [43], it has been shown that over a BIAWGN channel, LSMS decoder puts a tight limit on the fraction of edges connected to degree-two variable nodes, i.e., λ_2. This stability condition is equivalent to the stability condition for a SP code over a zero-capacity channel. Since λ_2 significantly affects the code rate, LSMS codes cannot achieve close-to-capacity rates. In Section 3.3, we show that the tight limit on λ_2 imposed by the LSMS decoder holds for all BISO channels. Then, in Section 3.4, we show that the proposed modified method relaxes this severe stability condition and hence can achieve higher rates. Irregular LDPC codes based on the modified method are designed in Section 3.5 which shows that higher code rates can indeed be obtained.

3.2 Stability Condition for Density Evolution

Convergence analysis of LDPC codes is often performed by characterizing the fixed points of density evolution. For SP decoding, we have the following theorem:

Theorem 3.1 [FIXED POINT CHARACTERIZATION FOR SP [2]]**:** Consider a given degree distribution pair (λ, ρ) and assume that transmission takes place over a BISO channel with LLR pdf a_{ch}.

- [Convergence] The sequence of densities a_ℓ converges to a symmetric density a_∞ which is a fixed point solution to $\mathrm{a} = \mathrm{a}_{\mathrm{ch}} \otimes \lambda(\rho(\mathrm{a}))$.
- [Sufficiency] If there does not exist a symmetric density $\mathrm{a} \neq \Delta_\infty$ such that $\mathrm{a} = \mathrm{a}_{\mathrm{ch}} \otimes \lambda(\rho(\mathrm{a}))$ then $\mathcal{P}(\mathrm{a}_\ell)$ converges to zero as ℓ tends to infinity, or, equivalently, $\mathrm{a}_\infty = \Delta_\infty$.
- [Necessity] If there exists a symmetric density $\mathrm{a} \neq \Delta_\infty$ such that $\mathrm{a} = \mathrm{a}_{\mathrm{ch}} \otimes \lambda(\rho(\mathrm{a}))$ then $\mathcal{P}(\mathrm{a}_\ell)$ does not converge to zero as ℓ tends to infinity, or, equivalently, $\mathrm{a}_\infty \neq \Delta_\infty$.

▼

Successful decoding for a given degree distribution pair on a given channel means that density evolution should only have a zero-error fixed point, i.e., Δ_∞.

3.2.1 Stability Condition for SP Decoder

It is desirable that the perfect decoding fixed point be stable, i.e., if the decoder gets close to the perfect decoding, it converges to the zero-error fixed point. The stability condition is basically a joint condition on the degree distribution of the code

ensemble and the channel on which the transmission takes place. For a fixed channel, stability condition puts a constraint on the code degree distributions. It ensures that once the probability of error gets small enough, decoding will be successful.

From the EXIT chart's point of view, stability condition can be interpreted as a condition on the slope of elementary EXIT curve for degree-two variable nodes near perfect decoding. More precisely, consider a density at the output of check nodes at iteration ℓ as

$$\mathrm{b}_\ell = \epsilon\Delta_0 + \bar{\epsilon}\Delta_\infty$$

where $\epsilon \in [0,1]$ and $\epsilon \to 0$. Then, the slope of the output entropy at degree-d_v variable nodes, where $d_v \geq 3$, will be

$$\begin{aligned}\lim_{\epsilon\to 0}\frac{d}{d\epsilon}H(\mathrm{a}_\ell) &= \lim_{\epsilon\to 0}\frac{d}{d\epsilon}H\big(\epsilon^{d_v-1}\mathrm{a}_{\mathrm{ch}} + (1-\epsilon^{d_v-1})\Delta_\infty\big)\\ &= \lim_{\epsilon\to 0}\frac{d}{d\epsilon}H\big(\epsilon^{d_v-1}\mathrm{a}_{\mathrm{ch}}\big)\\ &= 0\end{aligned}$$

which shows that the elementary EXIT curves for all variable nodes with $d_v \geq 3$ approach perfect decoding fixed point with a zero slope and hence will not cross the $y = x$ line. On the other hand, the EXIT curve of degree-two variable nodes might cross the bisector of the first quadrant, hence no convergence.

In the context of stability condition, all BISO channels are mapped to real numbers through the Bhattacharyya functional defined in Section 2.3.2. For SP decoding, we have the following stability condition theorem:

Theorem 3.2 [STABILITY CONDITION FOR SP [2, 27]]: Given a degree distribution pair (λ, ρ) and a symmetric channel a_{ch}, for an arbitrary a_0, we have:

• [Necessity] If $\lambda'(0)\rho'(1)\mathcal{B}(\mathrm{a}_{\mathrm{ch}}) > 1$, then there exits a strictly positive constant $\xi = \xi(\lambda, \rho, \mathrm{a}_{\mathrm{ch}})$ such that

$$\liminf_{\ell\to\infty}\mathcal{P}(\mathrm{a}_\ell) \geq \xi,$$

for all $\mathrm{a}_0 \neq \Delta_\infty$.

• [Sufficiency] If $\lambda'(0)\rho'(1)\mathcal{B}(\mathrm{a}_{\mathrm{ch}}) < 1$ then there exists a strictly positive constant $\xi = \xi(\lambda, \rho, \mathrm{a}_{\mathrm{ch}})$ such that if, for some $\ell \in \mathbb{N}$, $\mathcal{P}(\mathrm{a}_\ell) \leq \xi$ then

$$\lim_{\ell\to\infty}\mathcal{P}(\mathrm{a}_\ell) = 0.$$

▼

Example 3.1: For a BSC(p) with the LLR pdf given in Example 2.2, we have

$$\mathcal{B}(\mathrm{a}_{\mathrm{BSC}(p)}) = \mathbb{E}(e^{-\frac{x}{2}}) = 2\sqrt{p\bar{p}}$$

which results in

$$\lambda'(0)\rho'(1) < \frac{1}{2\sqrt{p\bar{p}}}.$$

Therefore, an upper bound for the threshold of the code is obtained as

$$p^*_{\mathrm{SP}}(\lambda,\rho) \leq \frac{1}{2}\left(1 - \left[1 - \frac{1}{\lambda'(0)\rho'(1)}\right]^{\frac{1}{2}}\right).$$

$\Diamond$

It should be emphasized that stability condition does not guarantee the convergence of the code except in few occasions. As a case in point, for *circuit codes* where $\lambda(x) = x$, stability condition determines the threshold of the code. Also, for LDPC codes where $\lambda'(0) = 0$, the zero-error fixed point is always stable.

3.2.2 Stability Condition for MS Decoder

MS decoder differs from SP decoder in certain areas. First of all, the pdf of random messages at each iteration of MS decoding will not be symmetric anymore. Also, the probability of error as a function of iteration number is not monotone [2]. For the MS decoder, we can derive density evolution formula. For an $x \geq 0$, define

$$\varphi_+(x) = \int_x^{+\infty} \mathrm{a}_{\ell-1}(t)dt, \quad \varphi_-(x) = \int_{-\infty}^{-x} \mathrm{a}_{\ell-1}(t)dt.$$

Then, density evolution is defined as [34, 44]

$$\mathrm{a}_\ell = \mathrm{a}_{\mathrm{ch}} \otimes \lambda(\mathrm{b}_\ell)$$

where $\mathrm{b}_\ell(x)$ is

$$\mathrm{b}_\ell(x) = \sum_k \rho_k \frac{k-1}{2}\Big(\left[\mathrm{a}_{\ell-1}(x) + \mathrm{a}_{\ell-1}(-x)\right]\left[\varphi_+(|x|) + \varphi_-(|x|)\right]^{k-2} + \left[\mathrm{a}_{\ell-1}(x) - \mathrm{a}_{\ell-1}(-x)\right]\left[\varphi_+(|x|) - \varphi_-(|x|)\right]^{k-2}\Big).$$

Bhattad *et al.* in [14] extend the stability condition theorem to the MS decoder when $\mathrm{a}_0 = \mathrm{a}_{\mathrm{ch}}$. They show that for MS decoder, the sufficiency part of Theorem 3.2 remains the same.

3.3 Linear Scaling and Stability Condition

Let m_{MS} and m_{SP} be the LLR messages at the output of MS and SP check nodes of the same degree with the same set of inputs. It has been shown [42] that

- $\text{sign}(m_{\text{MS}}) = \text{sign}(m_{\text{SP}})$,
- $|m_{\text{MS}}| > |m_{\text{SP}}|$.

For the regular codes, the performance under the MS decoding can be improved if the messages at the output of MS check nodes, i.e., m_{MS} are scaled down by a factor α where $\alpha > 1$. The optimum value of α is obtained through density evolution. It depends on the signal-to-noise ratio (SNR) and should be adjusted for each iteration [13,42]. However, for practical reasons, it is usually assumed that α is a fixed constant. In the following example, we see that for irregular codes that are specifically designed for SP decoder, scaling down the messages may significantly degrade the performance in terms of the decoding threshold of the code.

Example 3.2: Consider the following rate one-half LDPC code designed for SP decoding over a BIAWGN channel taken from [27]:

$$\begin{aligned}\lambda(x) &= 0.23802x + 0.20997x^2 + 0.03492x^3 + 0.12015x^4 \\ &\quad + 0.01587x^6 + 0.00480x^{13} + 0.37627x^{14} \\ \rho(x) &= 0.98013x^7 + 0.01987x^8\end{aligned}$$

Using density evolution, we see that the best LSMS decoder for this code is MS decoder ($\alpha = 1$). For example, $\alpha = 1.1$ gives the threshold of 2.69 dB which is far from the threshold of MS decoder (1.49 dB) and SP decoder (0.34 dB). ◊

Example 3.2 illustrates that when the LSMS decoder is used, one must design specific codes for LSMS codes. In [14], the problem of code design for the MS and LSMS decoders is considered and it is shown that codes specifically designed for the LSMS decoder achieve smaller gaps from the capacity compared to MS codes. Nevertheless, the gap is still large compared to the gap of SP codes.

In this section, we discuss why codes designed based on the LSMS decoding still perform far from the SP codes. Moreover, is there any strategy better than linear scaling? The later will be answered in Section 3.4.

In the following theorem, the sufficiency part of the stability condition for the LSMS algorithm is stated. The same result is shown in [43], but only for the BIAWGN channel.

For a random variable X, we use the cumulant generating function which is

$$\Phi_X(\theta) = \log \mathbb{E}(e^{\theta X}) = \sum_{j=1}^{\infty} \kappa_j \frac{\theta^j}{j!}$$

where $\{\kappa_j\}_{j\in\mathbb{N}}$ are the cumulants of the random variable X. Also, for a set of independent (not necessarily identically distributed) random variables $\{X_k\}_{k=1}^n$, we can write

$$\Phi_{\sum_{k=1}^n X_k}(\theta) = \sum_{k=1}^{n} \Phi_{X_k}(\theta).$$

Theorem 3.3 [STABILITY CONDITION FOR THE LINEAR SCALING MIN-SUM DECODER]: Consider a degree distribution pair (λ, ρ) with linear scaling min-sum decoder and a symmetric channel with LLR pdf $\mathrm{a_{ch}}$. For $\mathrm{a_0} = \mathrm{a_{ch}}$, if

$$\lambda'(0)\rho'(1) < 1$$

then the fixed point Δ_∞ is stable.

Proof: Following the same lines in [27, 43], we start with a density

$$\mathrm{b_0} = 2\epsilon\Delta_0 + \overline{2\epsilon}\Delta_\infty$$

close to perfect decoding where $\mathcal{P}(\mathrm{b_0}) = \epsilon$. After a complete iteration, we get

$$\mathrm{b_1}(x) = 2\epsilon\lambda'(0)\rho'(1)\alpha \mathrm{a_{ch}}(\alpha x) + \overline{2\epsilon\lambda'(0)\rho'(1)}\Delta_\infty + O(\epsilon^2).$$

Finally, after n whole iterations, we arrive at

$$\mathrm{b}_n(x) = 2\epsilon\big(\lambda'(0)\rho'(1)\big)^n \bigotimes_{k=1}^{n} \alpha^k \mathrm{a_{ch}}(\alpha^k x) + \overline{2\epsilon\big(\lambda'(0)\rho'(1)\big)^n}\Delta_\infty + O(\epsilon^2).$$

For the probability of error to get arbitrary small, we need

$$\log[\lambda'(0)\rho'(1)] < -\lim_{n\to\infty} \frac{1}{n} \log \underbrace{\mathcal{P}\Big(\bigotimes_{k=1}^{n} \alpha^k \mathrm{a_{ch}}(\alpha^k x)\Big)}_{\Pr\left\{\sum_{k=1}^n M_k/\alpha^k < 0\right\}} \tag{3.1}$$

where $\{M_k\}_{k=1}^n$ are n independent and identically distributed (i.i.d.) samples with the common density $\mathrm{a_{ch}}$. For $\alpha > 1$, let

$$M = \sum_{k=1}^{n} \frac{M_k}{\alpha^k}.$$

According to the Gärtner-Ellis theorem [45], we have

$$\log[\lambda'(0)\rho'(1)] < -\inf_{\theta} \lim_{n\to\infty} \frac{1}{n} \log \mathbb{E}(e^{\theta M}). \tag{3.2}$$

Let us derive the cumulants of M in terms of the cumulants of $\{M_k\}_{k=1}^n$. To this end, since $\{M_k\}_{k=1}^n$ are i.i.d., let $\{\gamma_m\}_{m\in\mathbb{N}}$ be the cumulants of the random variable M_k (independent of k) and $X_k = \frac{M_k}{\alpha^k}$. Since we have

$$\begin{aligned}
\Phi_M(\theta) &= \sum_{j=1}^{\infty} \kappa_j \frac{\theta^j}{j!} \\
&= \sum_{k=1}^{n} \sum_{m=1}^{\infty} \frac{\gamma_m}{m!} \left(\frac{\theta}{\alpha^k}\right)^m \\
&= \sum_{m=1}^{\infty} \gamma_m \frac{\theta^m}{m!} \sum_{k=1}^{n} \frac{1}{\alpha^{km}},
\end{aligned}$$

for a fixed $j \in \mathbb{N}$, we get

$$\kappa_j = \gamma_j \sum_{k=1}^{n} \frac{1}{\alpha^{kj}} = \frac{\gamma_j}{\alpha^j} \times \frac{1-\alpha^{-jn}}{1-\alpha^{-j}}.$$

Therefore, the right hand side of (3.2) becomes

$$\begin{aligned}
-\inf_{\theta} \lim_{n\to\infty} \frac{1}{n} \log \mathbb{E}(e^{\theta M}) &= -\inf_{\theta} \lim_{n\to\infty} \frac{1}{n} \sum_{j=1}^{\infty} \gamma_j \frac{\theta^j}{j!} \times \frac{1-\alpha^{-jn}}{\alpha^j - 1} \\
&= -\inf_{\theta} \lim_{n\to\infty} \frac{1}{n} \sum_{j=1}^{\infty} \underbrace{\frac{\gamma_j}{j!(\alpha^j-1)}}_{a_j} \theta^j = 0.
\end{aligned}$$

Thus, $\log[\lambda'(0)\rho'(1)] < 0$ which completes the proof.

Note that if for each M_k, $\Phi_{M_k}(\theta)$ exits then the summation $\sum_{j=1}^{\infty} a_j\theta^j$ exits too, because using the Laurent series' ratio test, it can be seen that

$$\lim_{j\to\infty} \left|\frac{a_{j+1}}{a_j}\right| = \lim_{j\to\infty} \left|\frac{\gamma_{j+1}}{\gamma_j(j+1)}\right|$$

which is equal to the ratio test for $\Phi_{M_k}(\theta)$.

□

Remark 3.1: For $\alpha = 1$, one can use the large deviations theory to conclude that (3.1) results in [2]

$$\lambda'(0)\rho'(1)\mathcal{B}(\mathrm{a_{ch}}) < 1.$$

▲

Remark 3.2: The stability condition in Theorem 3.3 corresponds to the stability condition of a useless (zero-capacity) channel when SP decoder is used. As a result of this severe limit on $\lambda'(0) = \lambda_2$, the code rate is adversely affected. In other words, LSMS decoder imposes a very harsh stability condition on the code. This also explains why applying linear scaling to codes optimized for SP can result in a very poor performance (see Example 3.2.)

▲

3.4 Modified Min-Sum Decoder

In this section, we propose a simple modified MS decoder which allows for larger values of λ_2 at essentially no extra complexity. The new scheme shown in Fig. 3.1, scales down LLRs less than x_0 and leaves LLRs larger than x_0 intact, i.e.,

$$M_{\text{out}} = \begin{cases} M_{\text{in}} & |M_{\text{in}}| \geq x_0 \\ \frac{M_{\text{in}}}{\alpha} & |M_{\text{in}}| < x_0 \end{cases} \tag{3.3}$$

where M_{in} is the LLR message at the output of an ordinary MS check node and M_{out} is the LLR message at the output of the modified MS check node. Also, $\alpha > 1$ and x_0 are two constants which are to be optimized. Our motivation for this modification stems from the fact that for large LLR values, the MS and SP check update rules are almost equivalent (see (2.5) and (2.8)). Thus, scaling down the messages in not necessary (and possibly harmful). The LSMS decoder, however, scales down even large LLRs which results in a low convergence rate and imposes a strict stability condition. In the following theorem, we show that the stability condition is improved using the proposed method.

To this end, if the pdf of M_{in} be $\mathrm{a}(x)$, then it can be shown that the pdf of M_{out} will be

$$\mathcal{T}_{\alpha}^{x_0}(\mathrm{a})(x) = \begin{cases} \mathrm{a}(x) & |x| > x_0 \\ \alpha \mathrm{a}(\alpha x) & |x| < x_0/\alpha \\ 0 & \text{otherwise} \end{cases}$$

which is shown in Fig. 3.1. The mapping $\mathcal{T}_{\alpha}^{x_0}(\cdot)$ has the following useful property:

Lemma 3.1 [LOWER AND UPPER BOUNDS ON $\mathcal{B}(\mathcal{T}_{\alpha}^{x_0}(\cdot))$]: For an $\alpha > 1$, there exists a constant $\mu \geq 1$, $\mu = \mu(\mathrm{a}_{\text{ch}}, \alpha, x_0)$, such that

$$\mathcal{B}(\mathrm{a}_{\text{ch}}) \leq \mathcal{B}\big(\mathcal{T}_{\alpha}^{x_0}(\mathrm{a}_{\text{ch}})\big) \leq \mu \mathcal{B}(\mathrm{a}_{\text{ch}}).$$

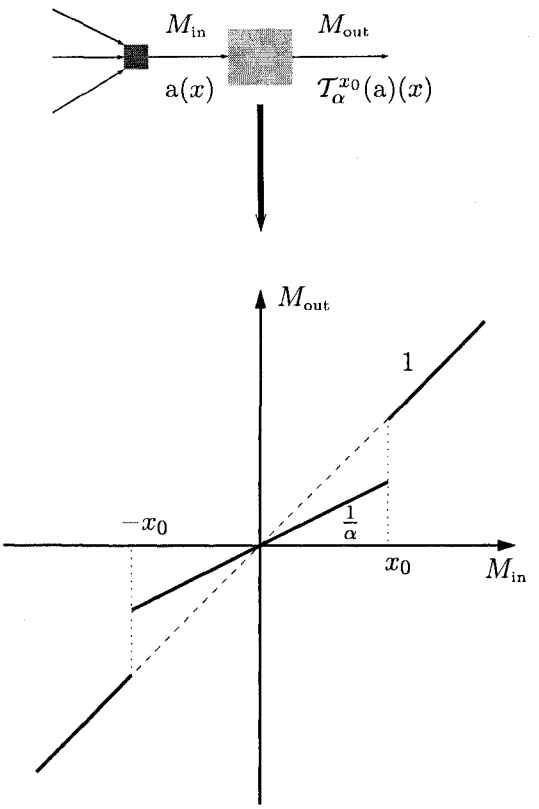


Figure 3.1: Modified scaling with $\alpha > 1$.

Proof: Since a_{ch} is a symmetric density, we see that for an $\alpha > 1$, the difference $\mathcal{B}\big(\mathcal{T}_{\alpha}^{x_0}(\text{a}_{\text{ch}})\big) - \mathcal{B}(\text{a}_{\text{ch}})$ will be

$$\begin{aligned}&\int_{|x|<\frac{x_0}{\alpha}} \alpha \text{a}_{\text{ch}}(\alpha x) e^{-\frac{x}{2}} dx - \int_{|x|<x_0} \text{a}_{\text{ch}}(x) e^{-\frac{x}{2}} dx \\ &= 2\int_0^{x_0} \text{a}_{\text{ch}}(x) e^{-\frac{x}{2}} \Big[\cosh\Big(\frac{\beta x}{2}\Big) - 1\Big] dx \geq 0 \end{aligned} \tag{3.4}$$

where $\beta = 1 - 1/\alpha$ and $\beta \in (0,1)$. Therefore, there exists a constant $\mu \geq 1$ such that

$$\mathcal{B}\big(\mathcal{T}_{\alpha}^{x_0}(\text{a}_{\text{ch}})\big) \leq \mu \mathcal{B}(\text{a}_{\text{ch}})$$

and it depends only on the channel LLR pdf, α and x_0. □

Theorem 3.4 [STABILITY CONDITION FOR THE MODIFIED LINEAR SCALING MIN–SUM DECODER]**:** For a (λ, ρ) LDPC code over a symmetric channel with LLR pdf a_{ch} under modified scaling min-sum decoding given in (3.3) and for every

$$\mu \geq \frac{\mathcal{B}\big(\mathcal{T}_{\alpha}^{x_0}(\text{a}_{\text{ch}})\big)}{\mathcal{B}(\text{a}_{\text{ch}})},$$

if

$$\lambda'(0)\rho'(1)\mu\mathcal{B}(\mathrm{a_{ch}}) < 1$$

then the fixed point Δ_∞ will be stable.

Proof: We follow the same lines in [27, 43]. Consider a perturbation of Δ_∞ as

$$\mathrm{b}_0 = 2\epsilon\Delta_0 + \overline{2\epsilon}\Delta_\infty$$

where $\mathcal{P}(\mathrm{b}_0) = \epsilon$. After a complete iteration, we have

$$\mathrm{b}_1 = 2\epsilon\lambda'(0)\rho'(1)\mathcal{T}_\alpha^{x_0}(\mathrm{a_{ch}}) + \overline{2\epsilon\lambda'(0)\rho'(1)}\Delta_\infty + O(\epsilon^2).$$

If we consider n iterations of density evolution, we get

$$\mathrm{b}_n = 2\epsilon\big(\lambda'(0)\rho'(1)\big)^n \times \underbrace{\mathcal{T}_\alpha^{x_0}\Big(\mathrm{a_{ch}} \otimes \mathcal{T}_\alpha^{x_0}\Big(\mathrm{a_{ch}} \otimes \cdots \mathcal{T}_\alpha^{x_0}\big(\mathrm{a_{ch}} \otimes \mathcal{T}_\alpha^{x_0}(\mathrm{a_{ch}})\big)\Big)\Big)}_{n \text{ times}} +$$
$$\overline{2\epsilon\big(\lambda'(0)\rho'(1)\big)^n}\Delta_\infty + O(\epsilon^2).$$

Using Lemma 3.1 and the multiplicative property of the Bhattacharyya functional in a variable node [2], we arrive at

$$\mathcal{B}(\mathrm{b}_n) < 2\epsilon\big(\lambda'(0)\rho'(1)\mu\mathcal{B}(\mathrm{a_{ch}})\big)^n + O(\epsilon^2).$$

If $\lambda'(0)\rho'(1)\mu\mathcal{B}(\mathrm{a_{ch}}) < 1$ then there exits a positive constant $\eta \in (0, 1)$ such that

$$\lambda'(0)\rho'(1)\mu\mathcal{B}(\mathrm{a_{ch}}) + \eta < 1.$$

For a sufficiently large n, once the modified MS decoder gets close enough to the zero-error fixed point, then

$$\mathcal{B}(\mathrm{b}_n) \leq 2\epsilon(\lambda'(0)\rho'(1)\mu\mathcal{B}(\mathrm{a_{ch}}) + \eta)^n.$$

Thus, as $n \to \infty$, both $\mathcal{B}(\mathrm{b}_n)$ and $\mathcal{P}(\mathrm{b}_n)$ will vanish since according to Lemma 2.1, for a small probability of error, we have

$$2\mathcal{P}(\mathrm{a}) \leq \mathcal{B}(\mathrm{a}) \leq 2\sqrt{\mathcal{P}(\mathrm{a})}.$$

This shows that Δ_∞ is a stable fixed point. □

Lemma 3.2 [RANGE OF $\mathcal{B}\big(\mathcal{T}_\alpha^{x_0}(\cdot)\big)$]**:** For a fixed channel and every pair of (α, x_0), we have

$$0 \leq \mathcal{B}\big(\mathcal{T}_\alpha^{x_0}(\mathrm{a_{ch}})\big) \leq 1.$$

Proof: The positiveness of $\mathcal{B}(\mathcal{T}_\alpha^{x_0}(\mathrm{a_{ch}}))$ is trivial. For the right side, first, we prove that $\mathcal{B}(\mathcal{T}_\alpha^{x_0}(\mathrm{a_{ch}}))$ is an increasing function of both x_0 and β. This is true since from (3.4), we have

$$\begin{aligned}\frac{\partial}{\partial x_0}\mathcal{B}(\mathcal{T}_\alpha^{x_0}(\mathrm{a_{ch}})) &= \frac{\partial}{\partial x_0}\left[\mathcal{B}(\mathcal{T}_\alpha^{x_0}(\mathrm{a_{ch}})) - \mathcal{B}(\mathrm{a_{ch}})\right] \\ &= 2\mathrm{a_{ch}}(x_0)e^{-\frac{x_0}{2}}\left[\cosh\left(\frac{\beta x_0}{2}\right) - 1\right] \\ &\geq 0\end{aligned}$$

and

$$\begin{aligned}\frac{\partial}{\partial \beta}\mathcal{B}(\mathcal{T}_\alpha^{x_0}(\mathrm{a_{ch}})) &= \frac{\partial}{\partial \beta}\left[\mathcal{B}(\mathcal{T}_\alpha^{x_0}(\mathrm{a_{ch}})) - \mathcal{B}(\mathrm{a_{ch}})\right] \\ &= \int_0^{x_0} x\mathrm{a_{ch}}(x)e^{-\frac{x}{2}}\sinh\left(\frac{\beta x}{2}\right)dx \\ &\geq 0.\end{aligned}$$

Second, we know that $\beta \in (0,1)$ which results in

$$\begin{aligned}\sup_{\alpha,x_0}\mathcal{B}(\mathcal{T}_\alpha^{x_0}(\mathrm{a_{ch}})) &= \mathcal{B}(\mathcal{T}_\infty^{\infty}(\mathrm{a_{ch}})) \\ &= \int_0^{\infty}\mathrm{a_{ch}}(x)(1+e^{-x})dx \\ &= 1.\end{aligned}$$

□

Remark 3.3: Since we have

$$\mu \geq \frac{\mathcal{B}(\mathcal{T}_\alpha^{x_0}(\mathrm{a_{ch}}))}{\mathcal{B}(\mathrm{a_{ch}})},$$

modified MS codes allow for λ_2 values in the range of

$$\lambda_2 \in [0, \lambda_2^*)$$

where

$$\lambda_2^* = \frac{1}{\rho'(1)\mathcal{B}(\mathcal{T}_\alpha^{x_0}(\mathrm{a_{ch}}))}. \tag{3.5}$$

According to Lemma 3.1 and Lemma 3.2, it is clear that for a fixed right degree distribution,

$$\lambda_2^{\mathrm{LSMS}} \leq \lambda_2^* \leq \lambda_2^{\mathrm{SP}}$$

where $\lambda_2^{\mathrm{LSMS}}$ and λ_2^{SP} are the maximum λ_2 values given by the stability condition of LSMS and SP decoders, respectively.

▲

Table 3.1: LDPC code design results for the BIAWGN channel.

SNR [dB]	-5	-3	-1	$+1$	$+3$
C	0.3495	0.4867	0.6430	0.7951	0.9124
Rate	0.3227	0.4606	0.6196	0.7802	0.9044
α^*	1.4	1.4	1.4	1.4	1.4
x_0^*	2.50	3.00	3.25	3.75	4.25
λ_2	0.2807	0.2311	0.2165	0.1668	0.1594
λ_2^*	0.3004	0.2486	0.2271	0.1747	0.1604
$\rho'(1)$	4.5	6.5	9.5	19.5	44

3.5 Code Design

In this section, we design codes for the proposed modified MS decoder introduced in Section 3.4.

Over a BIAWGN channel, we consider a fixed energy per symbol $\frac{E_s}{N_0}$ and different pairs of (α, x_0). For each pair, according to Section 2.3.4, we design a code with the highest rate and maximum allowed variable degree of 30 (and 40 for the highest $\frac{E_s}{N_0}$). We allow a maximum of 800 iterations of density evolution. Then, (α^*, x_0^*) which has the highest rate is selected as the best code. Then, the above procedure is repeated for other values of $\frac{E_s}{N_0}$.

Table 3.1 shows the optimization results where for each value of $\frac{E_s}{N_0}$, the achieved code rate and the best (α, x_0) are reported. Interestingly, $\alpha^* = 1.4$ is optimal for all cases and x_0^* is increasing with the channel condition. Also, for each value of $\frac{E_s}{N_0}$, the value of λ_2 of the best code is compared with the maximum λ_2 from the stability condition, i.e., λ_2^* given in (3.5). As it can be seen, for the best code we found, λ_2 is tightly upper bounded by λ_2^*. Optimal degree distributions for each value of $\frac{E_s}{N_0}$ are reported in Table 3.2.

Table 3.2: Optimized degree distributions for the BIAWGN channel.

SNR [dB]	-5	-3	-1	$+1$	$+3$
λ_2	0.2807	0.2311	0.2165	0.1668	0.1594
λ_3	0.2568	0.2523	0.2594	0.2489	0.2611
λ_4	0.0017	0.0019	0.0018	0.0026	0.0015
λ_5	0.0032	0.0035	0.0049	0.0058	0.0046
λ_6	0.0111	0.0117	0.0928	0.0232	0.2116
λ_7	0.1644	0.1609	0.1625	0.1933	0.0719

Continued on the next page

Table3.2 (continued)

SNR [dB]	−5	−3	−1	+1	+3
λ_8	0.0613	0.0911	0.0121	0.0653	0.0097
λ_9	0.0139	0.0160	0.0061	0.0217	0.0054
λ_{10}	0.0071	0.0074	0.0045	0.0118	0.0045
λ_{11}	0.0049	0.0047	0.0042	0.0080	0.0045
λ_{12}	0.0039	0.0034	0.0044	0.0061	0.0054
λ_{13}	0.0034	0.0028	0.0051	0.0049	0.0076
λ_{14}	0.0031	0.0024	0.0066	0.0042	0.0131
λ_{15}	0.0030	0.0022	0.0096	0.0038	0.0301
λ_{16}	0.0031	0.0021	0.0161	0.0035	0.0951
λ_{17}	0.0032	0.0020	0.0312	0.0034	0.0678
λ_{18}	0.0034	0.0020	0.0576	0.0033	0.0201
λ_{19}	0.0036	0.0020	0.0480	0.0033	0.0085
λ_{20}	0.0040	0.0021	0.0228	0.0033	0.0046
λ_{21}	0.0046	0.0022	0.0115	0.0035	0.0029
λ_{22}	0.0052	0.0024	0.0067	0.0037	0.0020
λ_{23}	0.0062	0.0026	0.0043	0.0041	0.0014
λ_{24}	0.0075	0.0029	0.0030	0.0046	0.0011
λ_{25}	0.0092	0.0034	0.0022	0.0053	0.0009
λ_{26}	0.0118	0.0042	0.0017	0.0065	0.0007
λ_{27}	0.0157	0.0055	0.0014	0.0084	0.0006
λ_{28}	0.0218	0.0081	0.0011	0.0122	0.0005
λ_{29}	0.0321	0.0153	0.0009	0.0225	0.0004
λ_{30}	0.0500	0.1519	0.0008	0.1458	0.0004
λ_{31}					0.0003
λ_{32}					0.0003
λ_{33}					0.0003
λ_{34}					0.0003
λ_{35}					0.0002
λ_{36}					0.0002
λ_{37}					0.0002
λ_{38}					0.0002
λ_{39}					0.0002
λ_{40}					0.0002
ρ_5	0.5				
ρ_6	0.5				
ρ_7		0.5			
ρ_8		0.5			
ρ_{10}			0.5		

Continued on the next page

Table3.2 (continued)

SNR [dB]	-5	-3	-1	$+1$	$+3$
ρ_{11}			0.5		
ρ_{20}				0.5	
ρ_{21}				0.5	
ρ_{45}					1
C	0.3495	0.4867	0.6430	0.7951	0.9124
Rate	0.3227	0.4606	0.6196	0.7802	0.9044

Fig. 3.2 compares the achieved rates and corresponding gaps to the capacity of the codes based on the proposed modified method with the LSMS codes reported in [14], MS codes taken from [46] and SP codes reported in [47]. Fig. 3.2 shows that the codes based on the proposed method have lower gaps to the capacity compared to MS and LSMS codes while the complexity of decoding remains essentially the same.

3.6 Conclusion

In this chapter, LDPC code design for a low-complexity decoder was considered. We observed that the performance of the irregular codes designed for SP decoder with the LSMS decoder can be very poor. Also, it was shown that the LDPC codes specifically designed for LSMS decoder cannot achieve close-to-capacity code rates since the stability condition is severe and forces λ_2 to be small. The proposed scheme was shown to exhibit a stability condition similar to the SP decoder and to allow for higher values of λ_2. The LDPC codes that are designed based on our proposed scheme have lower gaps to capacity compared to both MS and LSMS codes while the decoding complexity remains essentially the same.

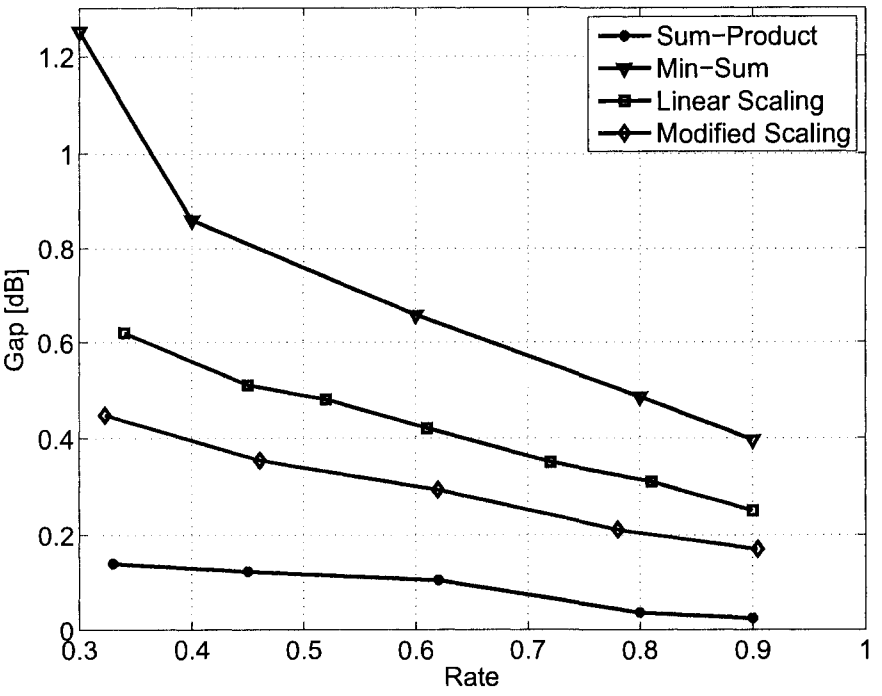


Figure 3.2: Comparison of the achieved gap to the capacity for different decoders.

Chapter 4

Low Complexity LDPC Coding for Gaussian Broadcast Channels

4.1 Introduction

The problem of simultaneous communication of a single source to multiple receivers, which is known as the broadcast channel, was first introduced by Cover in [48]. So far, the capacity region of certain classes of broadcast channels are known, however, the capacity region of a broadcast channel in general is still unknown.

Searching for a practical coding scheme, Berlin *et al.* [16], based on the achievable rate region given in [49], studied the code design problem for a two-user fading Gaussian broadcast channel. They used LDPC codes as the coding framework. From Chapter 2, we know that LDPC codes, if properly designed, are highly capable of operating at an SNR close to the Shannon limit with a vanishing probability of error [27]. It is shown in [16] that the performance loss due to using a binary code instead of a Gaussian code at low SNRs is negligible. Having the channel state information (CSI) only at the receivers, the authors in [16] show that at low SNRs, using superposition encoding and joint decoding, close-to-capacity LDPC codes can be found. In their scheme, since the user messages are superimposed, the message updating rule based on the factor graph associated with the MAP estimation, imposes a mapper node which connects the users' Tanner graphs. This mapper node not only increases the decoding complexity, also it requires both users to have the codebook of each other. Moreover, the codes should be jointly designed which is a complex task.

On the other hand, as it is pointed out in [16], the performance loss at high

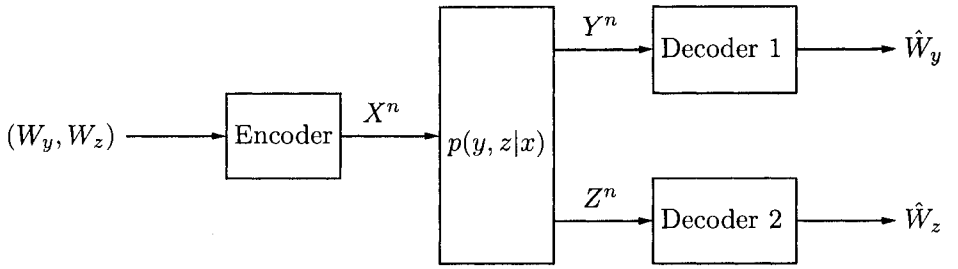


Figure 4.1: Broadcast channel.

SNRs incurred by the binary input can be significant. One solution to get around this problem is to use higher order constellations instead of BPSK signaling. This solution will make the mapper node even more complicated.

In this chapter, we propose a suboptimal scheme whose complexity is considerably less than the scheme proposed in [16]. Motivated by the bit-interleaved coded modulation (BICM) scheme [50], a novel labeling method is proposed which removes the mapper node. Therefore, each user can use its own LDPC code and there is no need to have the code of the other user.

In Section 4.2, we briefly review the main results known for broadcast channels. We discuss using LDPC codes for a two-user Gaussian broadcast channel in Section 4.3. Our method is proposed in Section 4.4 and LDPC codes based on our method are designed in Section 4.5. Finally, we conclude this chapter in Section 4.6.

4.2 Broadcast Channels

Definition 4.1 [Broadcast Channel [17]]: A two-user broadcast channel, depicted in Fig. 4.1, consists of an input alphabet $\mathcal{X}$, two output alphabets $\mathcal{Y}$ and $\mathcal{Z}$, and a set of channel transition probabilities $p(y, z|x)$ where $(x, y, z) \in \mathcal{X} \times \mathcal{Y} \times \mathcal{Z}$.

A broadcast channel is said to be memoryless if different uses of channel given the input sequence are independent from each other, i.e.,

$$p(y^n, z^n|x^n) = \prod_{i=1}^{n} p(y_i, z_i|x_i).$$

Definition 4.2 [Broadcast Code [17]]: A $(2^{nR_y}, 2^{nR_z}, n)$ broadcast code consists of:

1. Two equiprobable message sets $\mathcal{W}_y = \{1, 2, \ldots, M_y\}$ and $\mathcal{W}_z = \{1, 2, \ldots, M_z\}$ where $M_y = 2^{nR_y}$ and $M_z = 2^{nR_z}$

2. A codebook $\mathfrak{C}_n(R_y, R_z) = \left\{x^n(w_y, w_z) \in \mathcal{X}^n \middle| (w_y, w_z) \in \mathcal{W}_y \times \mathcal{W}_z\right\}$ which has $M_y \times M_z$ codewords of length n and symbols from the input alphabet $\mathcal{X}$

3. Two decoders which assign two message indices $\hat{w}_y(y^n) \in \mathcal{W}_y$ and $\hat{w}_z(z^n) \in \mathcal{W}_z$ to each received observation pair (y^n, z^n).

The goal is to send private messages to both receivers with a vanishing probability of error. The users can have a common message, however, in this work, we are only interested in the private messages. Two private messages are drawn independently from two message sets $\mathcal{W}_y$ and $\mathcal{W}_z$, and then the corresponding codeword is transmitted over the broadcast channel. A pair of rates (R_y, R_z) is said to be achievable if there exists a $(2^{nR_y}, 2^{nR_z}, n)$ broadcast code with vanishing average probabilities of error at both of the receivers, as $n \to \infty$ [48]. The capacity region of a broadcast channel is the convex closure of all the achievable rates. The following lemma will prove useful in our analysis [17]:

Lemma 4.1 [DEPENDENCY OF THE BROADCAST CHANNEL CAPACITY ON ITS MARGINALS [17]]**:** The capacity region of a broadcast channel depends only on the conditional marginal densities of $p(y, z|x)$, i.e., $p(y|x)$ and $p(z|x)$.

Proof: Define $\mathrm{P}_{e,y} = \Pr\{\hat{W}_y \neq W_y\}$ and $\mathrm{P}_{e,z} = \Pr\{\hat{W}_z \neq W_z\}$ as the single user error probabilities depending only on $p(y|x)$ and $p(z|x)$, respectively. We have the union bound as

$$\begin{aligned}\mathrm{P}_e &= \Pr\{(\hat{W}_y, \hat{W}_z) \neq (W_y, W_y)\} \\ &= \Pr\{\hat{W}_y \neq W_y \cup \hat{W}_z \neq W_z\} \\ &\leq \mathrm{P}_{e,y} + \mathrm{P}_{e,z}.\end{aligned}$$

Also, it can be seen that

$$\max\{\mathrm{P}_{e,y}, \mathrm{P}_{e,z}\} \leq \mathrm{P}_e.$$

Therefore,

$$\mathrm{P}_{e,y} \to 0 \text{ and } \mathrm{P}_{e,z} \to 0 \Longleftrightarrow \mathrm{P}_e \to 0$$

which completes the proof. □

The single letter characterization of the capacity region of a general broadcast channel is unknown yet; in special cases, however, the capacity region is known.

Here, we confine our attention to two cases: degraded broadcast channels and more capable broadcast channels.

Definition 4.3 [Degraded Broadcast Channel-More Capable Broadcast Channel [48, 51]]: If the channel transition probability can be factorized as

$$p(y,z|x) = p(z|y)p(y|x)$$

then the broadcast channel is physically degraded which implies that X, Y, and Z forms a Markov chain, i.e., $X \rightarrow Y \rightarrow Z$ [17]. A broadcast channel is more capable if [51]

$$I(X;Y) \geq I(X;Z), \quad \text{for all } p(x).$$

According to the data processing inequality, the set of degraded broadcast channels are a subset of more capable broadcast channels.

Let $\succeq$ denote the generalized inequality with respect to the nonnegative orthant. Also, we show the convex hull by CH. Bergmans [52] proved the following theorem:

Theorem 4.1 [Capacity of a Degraded Broadcast Channel [52]]: The capacity region of a degraded broadcast channel $X \rightarrow Y \rightarrow Z$ is the set of rates (R_y, R_z) such that

$$\underset{p(v,x)\in\mathfrak{D}^{\text{deg}}}{\text{CH}} \left\{ (R_y, R_z) \succeq \mathbf{0} \;\middle|\; \begin{array}{l} R_z \leq I(V;Z) \\ R_y \leq I(X;Y|V) \end{array} \right\}$$

where

$$\mathfrak{D}^{\text{deg}} = \Big\{ p(v,x) : (V,X) \sim p(v)p(x|v),\ |\mathcal{V}| \leq \min\{|\mathcal{X}|, |\mathcal{Y}|, |\mathcal{Z}|\} \Big\}.$$

▼

The idea, which is depicted in Fig. 4.2, is that the auxiliary random variable V serves as a cloud center distinguishable by both receivers. There are totally M_z clouds available and each cloud contains M_y codewords. The "weaker" user, i.e., Z, can only see the clouds while the user Y can also see codewords within a cloud. In fact, user Y first stripes off the message of user Z (decodes the cloud) and then it can see the individual codewords within a cloud [52, 53]. This method is called *superposition coding.*

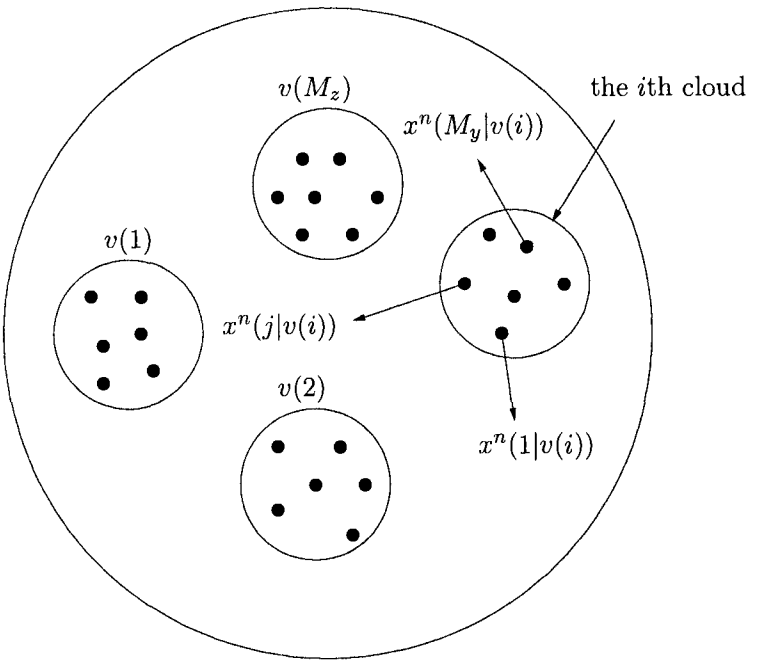


Figure 4.2: Bergmans coding.

Also, for a more capable broadcast channel, i.e., $V \rightarrow X \rightarrow (Y, Z)$, El Gamal [51] proved that the capacity region is

$$\underset{p(v,x)\in\mathfrak{D}^{\text{mcap}}}{\text{CH}} \left\{ (R_y, R_z) \succeq \mathbf{0} \;\middle|\; \begin{array}{l} R_z \leq I(V;Z) \\ R_y + R_z \leq \min\left\{I(X;Y), I(V;Z) + I(X;Y|V)\right\} \end{array} \right\} \tag{4.1}$$

where

$$\mathfrak{D}^{\text{mcap}} = \Big\{ p(v,x) : (V,X) \sim p(v)p(x|v),\ |\mathcal{V}| \leq |\mathcal{X}| + 2 \Big\}.$$

Since a degraded broadcast channel is a special case of a more capable channel, one can verify that for a degraded broadcast channel, the region in (4.1) coincides with the Bergmans region given in Theorem 4.1 [54].

4.2.1 Gaussian Broadcast Channels

The focus of this chapter is on the Gaussian broadcast channels which are defined as [16]

$$\begin{aligned} Y &= AX + N_y \\ Z &= BX + N_z \end{aligned} \tag{4.2}$$

where the additive white Gaussian noises are zero mean and have variance N_0, independent from the input X which is power constrained by $\mathbb{E}(|X|^2) \leq P$. Also, A and B are two ergodic memoryless processes, known at the receivers. In general, the broadcast channel given in (4.2) is neither degraded nor more capable. However, if the fading processes are constant (unfaded Gaussian) and $|A| > |B|$ then (4.2) will be degraded and the capacity region according to Theorem 4.1 is given by

$$\bigcup_{\alpha\in[0,1]} \left\{ (R_y, R_z) \succeq \mathbf{0} \;\middle|\; \begin{array}{l} R_y \leq C(\alpha|A|^2\gamma) \\ R_z \leq C(|B|^2\gamma) - C(\alpha|B|^2\gamma) \end{array} \right\} \tag{4.3}$$

where

$$C(x) = \frac{1}{2}\log_2[1+x]$$

and $\gamma = \frac{P}{N_0}$. The boundary of this region is achieved by

$$X = \sqrt{\alpha P}U + \sqrt{\bar{\alpha}P}V \tag{4.4}$$

where $\mathbb{E}(|X|^2) = P$, $\alpha \in [0,1]$ represents the fraction of power allocated for user Y, and $(U,V) \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_2)$ is a pair of independent normal random variables [48,55,56]. For $|A| = |B|$, since two channels are statistically the same, we can use time-sharing between the users. It is not difficult to see that the capacity region in this case coincides with the time-sharing region [49]

$$\left\{ (R_y, R_z) \succeq \mathbf{0} \;\middle|\; \frac{R_y}{C(|A|^2\gamma)} + \frac{R_z}{C(|B|^2\gamma)} \leq 1 \right\}.$$

Berlin and Tuninetti in [16], consider the performance achievable by a binary linear code instead of the Gaussian input given in (4.4). This means that (U,V) in (4.4) are drawn uniformly from $\{+1,-1\} \times \{+1,-1\}$. In this case, it is not difficult to show that the capacity region is given by [16]

$$\bigcup_{\alpha\in[0,1]} \left\{ (R_y, R_z) \succeq \mathbf{0} \;\middle|\; \begin{array}{l} R_y \leq J(\alpha|A|^2\gamma) \\ R_z \leq J(|B|^2\gamma) - J(\alpha|B|^2\gamma) \end{array} \right\} \tag{4.5}$$

where $J(t)$ is the capacity of a BIAWGN$(\frac{2}{\sqrt{t}})$ channel, i.e.,

$$J(t) = 1 - \int \frac{1}{\sqrt{2\pi t}} e^{-\frac{(x-t/2)^2}{2t}} \log_2[1+e^{-x}]dx$$

which is achieved by choosing X uniformly from $\{+\sqrt{P}, -\sqrt{P}\}$.

4.3 LDPC Codes for Gaussian Broadcast Channels

In this section, we review the method adapted by [16] to communicate over a two-user Gaussian broadcast channel using LDPC codes.

Let

$$\mathbf{x} = \sqrt{\alpha P}\mathbf{x}_y + \sqrt{\bar{\alpha} P}\mathbf{x}_z \tag{4.6}$$

be the superimposed transmitted vector where $\mathbf{x}_y$ and $\mathbf{x}_z$ are the binary codewords of the users Y and Z, respectively. We assume that these codewords are selected from two LDPC ensembles, $\mathfrak{C}_n(\lambda_y, \rho_y)$ and $\mathfrak{C}_n(\lambda_z, \rho_z)$. Also, by transmission of $\mathbf{x}$, we observe two vectors $\mathbf{y}$ and $\mathbf{z}$.

We pick two LDPC codes from the ensembles and denote the corresponding parity-check matrices by $\mathbf{H}_y$ and $\mathbf{H}_z$. The MAP estimate of the ith bit of the vector $\mathbf{x}_y$, denoted by $x_{i,y}$, is

$$\begin{aligned}
\hat{x}_{i,y} &= \arg \max_{x_{i,y}\in\{\pm 1\}} \; p(x_{i,y}|\mathbf{y}) \qquad i = 1, 2, \ldots, n \\
&= \arg \max_{x_{i,y}\in\{\pm 1\}} \; \sum_{\sim x_{i,y}} p(\mathbf{x}_y, \mathbf{x}_z|\mathbf{y}) \\
&= \arg \max_{x_{i,y}\in\{\pm 1\}} \; \sum_{\sim x_{i,y}} p(\mathbf{y}|\mathbf{x}_y, \mathbf{x}_z) p(\mathbf{x}_y, \mathbf{x}_z) \mathbb{1}_{\{\mathbf{H}_y\mathbf{x}_y=\mathbf{0}, \mathbf{H}_z\mathbf{x}_z=\mathbf{0}\}} \\
&= \arg \max_{x_{i,y}\in\{\pm 1\}} \; \sum_{\sim x_{i,y}} p(\mathbf{y}|\mathbf{x}) \times \mathbb{1}_{\mathbf{x}=\sqrt{\alpha P}\mathbf{x}_y+\sqrt{\bar{\alpha} P}\mathbf{x}_z} \times \mathbb{1}_{\{\mathbf{H}_y\mathbf{x}_y=\mathbf{0}, \mathbf{H}_z\mathbf{x}_z=\mathbf{0}\}} \\
&= \arg \max_{x_{i,y}\in\{\pm 1\}} \; \sum_{\sim x_{i,y}} \left(\prod_{i=1}^{n} p(y_i|x_i) \times \mathbb{1}_{x_i=\sqrt{\alpha P}x_{i,y}+\sqrt{\bar{\alpha} P}x_{i,z}} \right) \times \\
&\qquad \left(\prod_{j=1}^{n-k_y} \mathbb{1}_{\mathbf{h}_{j,y}^{\mathrm{T}}\mathbf{x}_y=0} \right) \left(\prod_{j=1}^{n-k_z} \mathbb{1}_{\mathbf{h}_{j,z}^{\mathrm{T}}\mathbf{x}_z=0} \right)
\end{aligned} \tag{4.7}$$

where $\mathbf{h}_{j,y}^{\mathrm{T}}$ and $\mathbf{h}_{j,z}^{\mathrm{T}}$ are the jth row of $\mathbf{H}_y$ and $\mathbf{H}_z$, respectively. A similar rule can be obtained for $x_{i,z}$, the ith bit of $\mathbf{x}_z$. Similar to (2.6), the MAP estimate of $x_{i,y}$ is shown by the factor graph depicted in Fig. 4.3.

In Fig. 4.3, the function node connecting the two Tanner graphs, which is called the mapper node by [16], increases the complexity in the following senses:

- The mapper node needs both users to have the code of each other in order to jointly decode the codewords.
- The decoding complexity increases considerably. This is because at each iteration of message passing decoder, according to Fig. 4.3, the messages from one

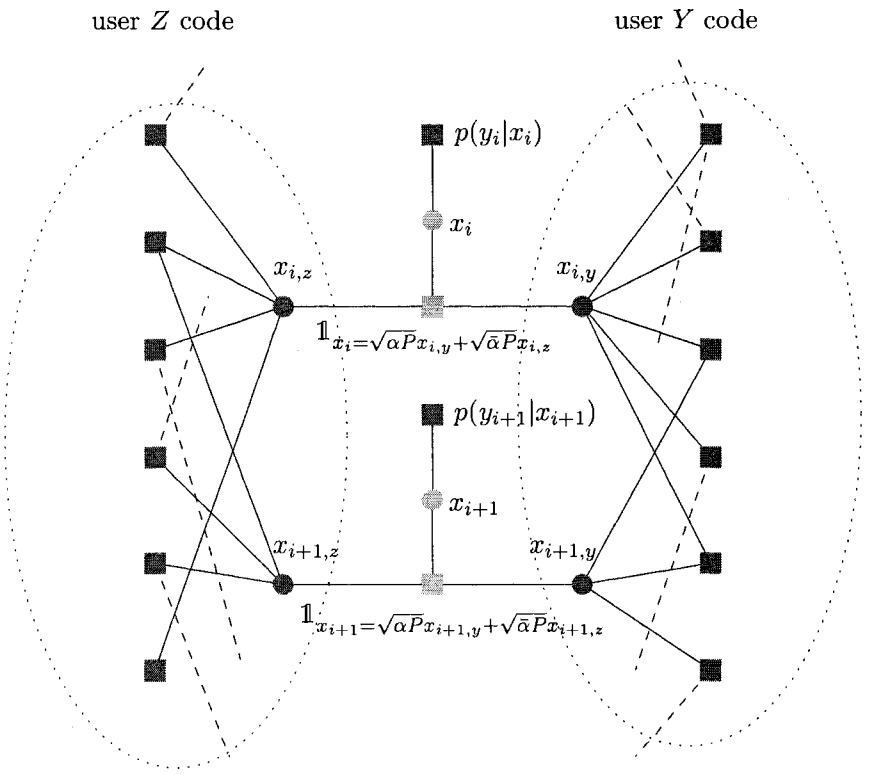


Figure 4.3: The factor graph associated to (4.7). The left (right) most function nodes are the check nodes of the LDPC code of the user Z (Y).

Tanner graph should be passed to the other graph to enhance the reliability of decision [16].

- The code design stage becomes more cumbersome than the single user case. In [16], authors use the differential evolution method [57] for the code design.

4.4 A Low Complexity LDPC Coding Scheme

In this section, we propose a method to use LDPC codes over a two-user Gaussian broadcast channel based on the BICM scheme.

4.4.1 Bit-Interleaved Coded Modulation

The BICM is a bandwidth-efficient coding method [50] which motivated us to propose a new scheme for communicating over a Gaussian broadcast channel.

First, let us explain coded modulation (CM) which is similar to BICM. In CM, the information bits are encoded using a binary code and the resulting sequence is

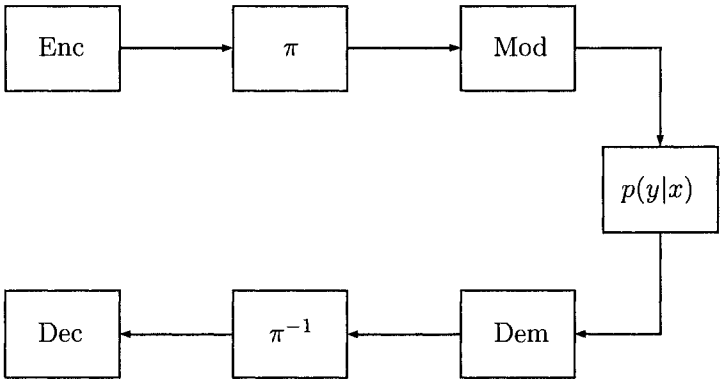


Figure 4.4: The system model for CM and BICM. In CM, π interleaves symbols while in the BICM, it is used to interleave bits.

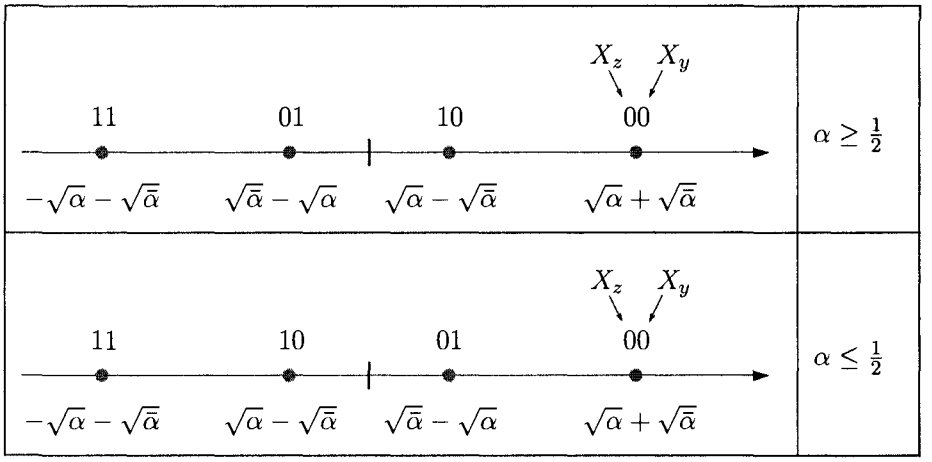
split into groups of $d = \log_2 D$ bits. Then, these groups are interleaved and mapped to a D-ary constellation to transmit over a channel with pdf of $p(y|x)$. In fact, the interleaver removes the dependency between the groups. At the receiver, a branch metric is computed for each received point. Then, branch metrics are de-interleaved and passed through the decoder.

The basic diagram of CM and BICM is shown in Fig. 4.4. Similar to the CM, BICM is based on the concatenation of a binary code, an interleaver and a high-order D-ary modulation [58]. The difference is that in CM, the interleaver is a symbol interleaver while in the BICM, it is used to interleave bits. The coded bits in BICM are fully interleaved and then every d bits is grouped together and sent over the channel using a D-ary constellation. At the receiver, after computing the LLR values of the coded bits and de-interleaving, a binary decoder is used as if the LLR values were the observations at a BPSK channel output [58]. It has been shown [50] that if we use Gray labeling for the D-ary constellation, then the capacity of BICM is extremely close to the capacity of the CM method, even for low SNRs. It means that using a bit interleaver and Gray labeling, a binary decoder can be used to get a performance which is almost the same as the CM.

Let $\mathcal{K}$ denote the set of points in the D-ary constellation used for the CM and BICM methods. The capacity under the CM and BICM are [50]

$$C_{\text{CM}} = \log_2 D - \mathbb{E}_{X,Y}\left[\log_2 \frac{\sum_{k\in\mathcal{K}} p(y|k)}{p(y|x)}\right]$$

Table 4.1: Binary labeling when $X = \sqrt{\alpha P}X_y + \sqrt{\bar{\alpha} P}X_z$. For simplicity, the symbol P is removed.



and

$$C_{\text{BICM}} = \log_2 D - \sum_{i=1}^{d} \mathbb{E}_{b,Y}\left[\log_2 \frac{\sum_{k\in\mathcal{K}} p(y|k)}{\sum_{k\in\mathcal{K}_i^b} p(y|k)}\right],$$

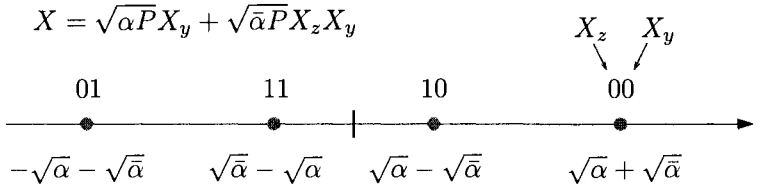
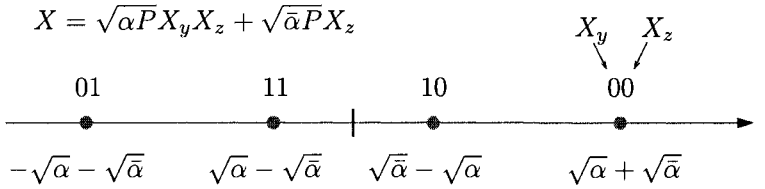
respectively, where $\mathcal{K}_i^b$ denotes the subset of constellation points whose ith bit is equal to b. It has been shown that $C_{\text{BICM}} \leq C_{\text{CM}}$ which means that BICM is a suboptimal method [50].

4.4.2 The Proposed Method

The superimposed codeword given in (4.6) can be viewed as a mapping which maps two independent bits to a point in a 4-PAM-like constellation shown in Table 4.1. This mapping uses a binary labeling method.

Now, consider two sequences of LDPC coded bits, each of which is intended for one user. We interleave both of these coded sequences separately and try to use Gray labeling for the 4-PAM-like constellation. Table 4.2 shows this scheme. The difference here is that in order to maintain Gray labeling, we have to swap the position of bits when α falls below one half. It can be shown that the equations in Table 4.2 satisfy the power constraint $\mathbb{E}(|X|^2) = P$. Also, note that the two part of these equations are not independent anymore. In fact, by using Gray labeling we have reduced dependency. Interleaving removes the dependency altogether to validate our decoding approach. We could use the dependency to improve our performance, but it is so minor that it does not worth it. In binary labeling, the

Table 4.2: Gray labeling where the symbol P is removed for simplicity.

$X = \sqrt{\alpha P}X_y + \sqrt{\bar{\alpha}P}X_zX_y$ 	$\alpha \geq \frac{1}{2}$
$X = \sqrt{\alpha P}X_yX_z + \sqrt{\bar{\alpha}P}X_z$ 	$\alpha \leq \frac{1}{2}$

dependency must be used (that is what the mapper node does) because it is too strong.

To analyze the proposed method, let us determine the capacity region using our method. For $\alpha \geq \frac{1}{2}$, we have

$$\begin{aligned}
R_z &\leq I(V;Z) \\
&= \sum_{x_z \in \{\pm 1\}} \int p(x_z)p(z|x_z) \log_2 \left[\frac{p(z|x_z)}{p(z)}\right] dz \\
&= \sum_{x_z \in \{\pm 1\}} \int \frac{1}{2} p(z|x_z) \log_2 \left[\frac{2p(z|x_z)}{\sum_{a \in \{\pm 1\}} p(z|X_z = a)}\right] dz \\
&= 1 - \frac{1}{2} \int p(z|X_z = +1) \log_2 \left[1 + \frac{p(z|X_z = -1)}{p(z|X_z = +1)}\right] dz - \\
&\quad \frac{1}{2} \int p(z|X_z = -1) \log_2 \left[1 + \frac{p(z|X_z = +1)}{p(z|X_z = -1)}\right] dz
\end{aligned}$$

and

$$\begin{aligned}
R_y &\leq I(X;Y|V) \\
&= H(Y|X_z) - H(Y|X, X_z) \\
&= H(Y|X_z) - H(AX + N_y|X, X_z) \\
&= \frac{1}{2}\left[H(Y|X_z = +1) + H(Y|X_z = -1)\right] - \frac{1}{2}\log_2(2\pi e N_0)
\end{aligned}$$

where $p(z|x_z = +1)$ is a mixture of two Gaussian pdfs. In a similar manner, we can derive formulations for $\alpha \leq \frac{1}{2}$. Finally, the capacity region using the BICM scheme and Gray labeling is given according to Theorem 4.1.

4.4.3 Stability Analysis

In order to analyze the stability condition on the code ensembles (see Section 3.2), we first describe how to obtain the LLR pdf for each of the users. For user Y and $\alpha \geq \frac{1}{2}$, according to Table 4.2, the LLR message received from the channel $p(y|x_y)$ is

$$\begin{aligned} m_y &= \log_2 \frac{p(y|X_y = +1)}{p(y|X_y = -1)} \\ &= \log_2 \frac{\sum_{x_z} p(x_z) p(y|X_y = +1, x_z)}{\sum_{x_z} p(x_z) p(y|X_y = -1, x_z)} \\ &= \log_2 \frac{p(y|X_y = +1, X_z = +1) + p(y|X_y = +1, X_z = -1)}{p(y|X_y = -1, X_z = +1) + p(y|X_y = -1, X_z = -1)} \\ &= \log_2 \frac{g_A(+1,+1) + g_A(+1,-1)}{g_A(-1,-1) + g_A(-1,+1)} \end{aligned}$$

where

$$g_A(p,q) = \frac{1}{\sqrt{2\pi N_0}} \exp\left\{ \frac{-1}{2N_0} \left(y - A\sqrt{P}(p\sqrt{\alpha} + q\sqrt{\bar{\alpha}})\right)^2 \right\}$$

and $p, q \in \{\pm 1\}$. For $\alpha \leq \frac{1}{2}$, we obtain

$$\begin{aligned} m_y &= \log_2 \frac{p(y|X_y = +1)}{p(y|X_y = -1)} \\ &= \log_2 \frac{g_A(+1,+1) + g_A(-1,-1)}{g_A(-1,+1) + g_A(+1,-1)}. \end{aligned}$$

Similarly, for user Z, we get

$$m_z = \log_2 \frac{g_B(+1,+1) + g_B(-1,-1)}{g_B(+1,-1) + g_B(-1,+1)}$$

and

$$m_z = \log_2 \frac{g_B(+1,+1) + g_B(-1,+1)}{g_B(-1,-1) + g_B(+1,-1)},$$

for and $\alpha \geq \frac{1}{2}$ and $\alpha \leq \frac{1}{2}$, respectively.

Lemma 4.2 [ASYMMETRY OF $p(y|x_y)$ AND $p(z|x_z)$]: The pdfs $p(y|x_y)$ and $p(z|x_z)$ are not symmetric.

Proof: We have

$$\begin{aligned} p(y|x_y) &= \sum_{x_z} p(x_z) p(y|x_y, x_z) \\ &= \frac{1}{2}\left[p(y|x_y, X_z = +1) + p(y|x_y, X_z = -1)\right]. \end{aligned}$$

For $\alpha \geq \frac{1}{2}$, we get

$$p(y|X_y = -1) = \frac{1}{2}\big[g_A(-1,-1) + g_A(-1,+1)\big] = p(-y|X_y = +1),$$

but this does not hold for $\alpha \leq \frac{1}{2}$ since,

$$p(y|X_y = -1) = \frac{1}{2}\big[g_A(-1,+1) + g_A(+1,-1)\big] \neq p(-y|X_y = +1).$$

This result can be extended to $p(z|x_z)$.

□

Let us show the pdf of m_y by $\mathrm{a}_{\mathrm{ch},y}(m)$. Since the channel $p(y|x_y)$ is not symmetric, the LLR pdf cannot be obtained by the all-one codeword assumption. However, according to [59], the LLR pdf can be obtained using

$$\mathrm{a}_{\mathrm{ch},y}(m) = \frac{1}{2}\big[\mathrm{a}_{\mathrm{ch},y}(m|X_y = +1) + \mathrm{a}_{\mathrm{ch},y}(-m|X_y = -1)\big]. \tag{4.8}$$

Note that if the channel was symmetric, (4.8) would lead to $\mathrm{a}_{\mathrm{ch},y}(m) = \mathrm{a}_{\mathrm{ch},y}(m|X_y = +1)$. Similarly, we denote the LLR pdf of user Z by $\mathrm{a}_{\mathrm{ch},z}(m)$.

Then, according to Theorem 3.2, we have the following theorem:

Theorem 4.2 [STABILITY CONDITION FOR USER Y AND Z]**:** Under sum-product decoding and for fixed right degree distributions $\rho_y(x)$ and $\rho_z(x)$, if

$$\lambda_{2,t}\rho_t'(1)\mathcal{B}(\mathrm{a}_{\mathrm{ch},t}) < 1, \quad t \in \{y, z\},$$

then the zero-error fixed point is stable for both users.

We denote the maximum λ_2 values allowable by stability condition by $\lambda_{2,y}$ and $\lambda_{2,z}$ which are

$$\lambda_{2,y}^{\mathrm{SP}} = \frac{1}{\rho_y'(1)\mathcal{B}(\mathrm{a}_{\mathrm{ch},y})} \tag{4.9}$$

and

$$\lambda_{2,z}^{\mathrm{SP}} = \frac{1}{\rho_z'(1)\mathcal{B}(\mathrm{a}_{\mathrm{ch},z})}, \tag{4.10}$$

respectively.

4.5 Simulation Results and Code Design

In this section, we compare the capacity regions resulted from a Gaussian input, binary labeling method from [16] and our proposed method based on the BICM scheme and Gray labeling. Then, we design codes based on our method.

Consider a two-user Gaussian broadcast channel given in (4.2) where

$$|A|^2\gamma = 5.059 \text{ dB, and } |B|^2\gamma = 3.871 \text{ dB.}$$

We chose these values to have a fair comparison with the region given in [16]. In Fig. 4.5, we compare the capacity region when the Gaussian input is used (see (4.3)) with the region given in [16] and (4.5), and with the region based on our method in Section 4.4. It can be seen that the most of the region is covered by our proposed method, but with considerably lower complexity. In fact, our method does not need joint decoding for each user. In other words, both users can use a single decoder. Therefore, both code design and decoding steps substantially become easier. We use the code optimization procedure discussed in Section 2.3.4 for each of the users, separately. It is noteworthy that the LLR pdf for each user is obtainable using the discussion in Section 4.4.

In Fig. 4.5, the achievable rates are shown by the cross points. Also, the optimized degree distributions for user Y and Z are reported in Table 4.3 and Table 4.4, respectively. In these tables, we compare the λ_2 values of the optimized codes with the maximum allowable values forced by the stability condition given in (4.9) and (4.10). Note that we did not put any stability constraint on λ_2 during the code optimization.

In Fig. 4.6, λ_2 values are compared. We can see that according to Theorem 4.2, in all cases, $\lambda_{2,y}$ and $\lambda_{2,z}$ are upper bounded by $\lambda_{2,y}^{\text{SP}}$ and $\lambda_{2,z}^{\text{SP}}$, respectively, which shows that the optimized codes are stable near the zero-error fixed point. For more discussions about the stability analysis, refer to Section 3.2.

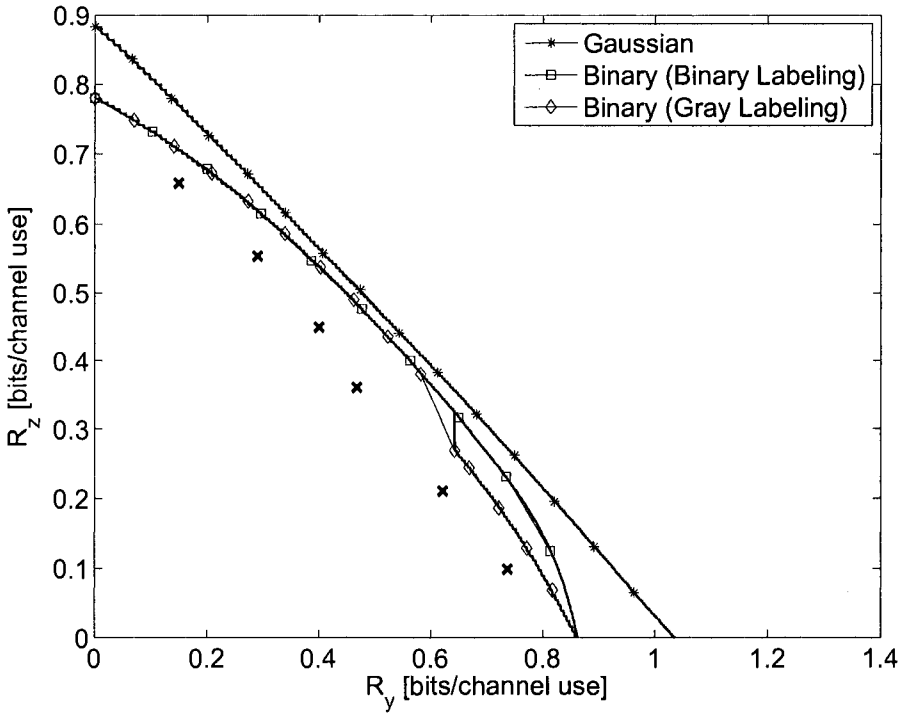


Figure 4.5: Comparison of the capacity region of a two-user Gaussian broadcast channel with different inputs. The cross points show the achieved rates by the proposed method given in Table 4.3 and Table 4.4.

Table 4.3: Optimized degree distributions for user Y with $|A|^2\gamma = 5.059$ dB.

α	0.1	0.2	0.3	0.4	0.8	0.9
$\lambda_{2,y}$	0.2738	0.2594	0.1911	0.1720	0.1597	0.1391
$\lambda_{3,y}$	0.1612	0.1974	0.1670	0.1677	0.1597	0.1707
$\lambda_{4,y}$	0.0487					
$\lambda_{5,y}$		0.0181		0.0456		
$\lambda_{6,y}$		0.1655	0.1725	0.0993	0.0971	0.1291
$\lambda_{7,y}$	0.0974				0.0971	0.0500
$\lambda_{8,y}$	0.0971					
$\lambda_{11,y}$		0.0483		0.0327		
$\lambda_{12,y}$		0.0458		0.1263		
$\lambda_{13,y}$			0.0769			
$\lambda_{14,y}$			0.0569		0.1222	0.1241
$\lambda_{15,y}$						0.0260
$\lambda_{32,y}$		0.0206				
$\lambda_{33,y}$		0.2449				
$\lambda_{49,y}$					0.0515	
$\lambda_{50,y}$	0.3217		0.3355	0.3565	0.3082	0.3609
$\rho_{5,y}$	1	0.1724				
$\rho_{6,y}$		0.8276				
$\rho_{8,y}$			0.4706			
$\rho_{9,y}$			0.5294			
$\rho_{10,y}$				1		
$\rho_{14,y}$					0.4828	
$\rho_{15,y}$					0.5172	
$\rho_{22,y}$						1
$\mathcal{B}(\mathrm{a}_{\mathrm{ch},y})$	0.8574	0.7501	0.6610	0.6022	0.4422	0.3290
$\lambda_{2,y}^{\mathrm{SP}}$	0.2916	0.2762	0.2013	0.1845	0.1673	0.1447
Rate	0.1500	0.2904	0.4017	0.4686	0.6218	0.7372

4.6 Conclusion

In this chapter, a low complexity method for communicating over a two-user Gaussian broadcast channel based on LDPC codes was presented. It was shown that comparing with the existing work in [16], our method is considerably less complex. We showed that in our method, each user can use a single LDPC code and the need for joint decoding at the receivers is eliminated. Also, we demonstrate that the code optimization problem can be broken down into two single-user code design problems, hence LDPC codes for a broadcast channel can be designed with a significantly lower complexity.

Table 4.4: Optimized degree distributions for user Z with $|B|^2\gamma = 3.871$ dB.

α	0.1	0.2	0.3	0.4	0.8	0.9
$\lambda_{2,z}$	0.1494	0.1697	0.2378	0.2522	0.2850	0.2977
$\lambda_{3,z}$	0.1700	0.1703	0.2017	0.2002	0.1980	0.1666
$\lambda_{5,z}$			0.0196	0.0219	0.0176	0.0651
$\lambda_{6,z}$	0.1238	0.1491	0.1606	0.1484	0.1592	0.0903
$\lambda_{7,z}$	0.0618	0.0205				
$\lambda_{10,z}$			0.0114	0.0929		
$\lambda_{11,z}$			0.1061	0.0117	0.0860	0.0674
$\lambda_{12,z}$						0.0274
$\lambda_{13,z}$		0.1305				
$\lambda_{14,z}$	0.0470	0.0066				
$\lambda_{15,z}$	0.0953					
$\lambda_{29,z}$			0.0236			
$\lambda_{30,z}$			0.2392			
$\lambda_{33,z}$				0.2674		
$\lambda_{34,z}$				0.0052		
$\lambda_{37,z}$					0.2309	
$\lambda_{38,z}$					0.0233	
$\lambda_{49,z}$						
$\lambda_{50,z}$	0.3527	0.3534				0.2854
$\rho_{4,z}$						0.4444
$\rho_{5,z}$					1	0.5556
$\rho_{6,z}$				0.4615		
$\rho_{7,z}$			0.2727	0.5385		
$\rho_{8,z}$			0.7273			
$\rho_{12,z}$		1				
$\rho_{16,z}$	0.4848					
$\rho_{17,z}$	0.5152					
$\mathcal{B}(\mathrm{a}_{\mathrm{ch},z})$	0.4138	0.5096	0.5977	0.8125	0.4422	0.8946
$\lambda_{2,z}^{\mathrm{SP}}$	0.1558	0.1784	0.2487	0.2679	0.3077	0.3144
Rate	0.6588	0.5542	0.4506	0.3605	0.2102	0.0989

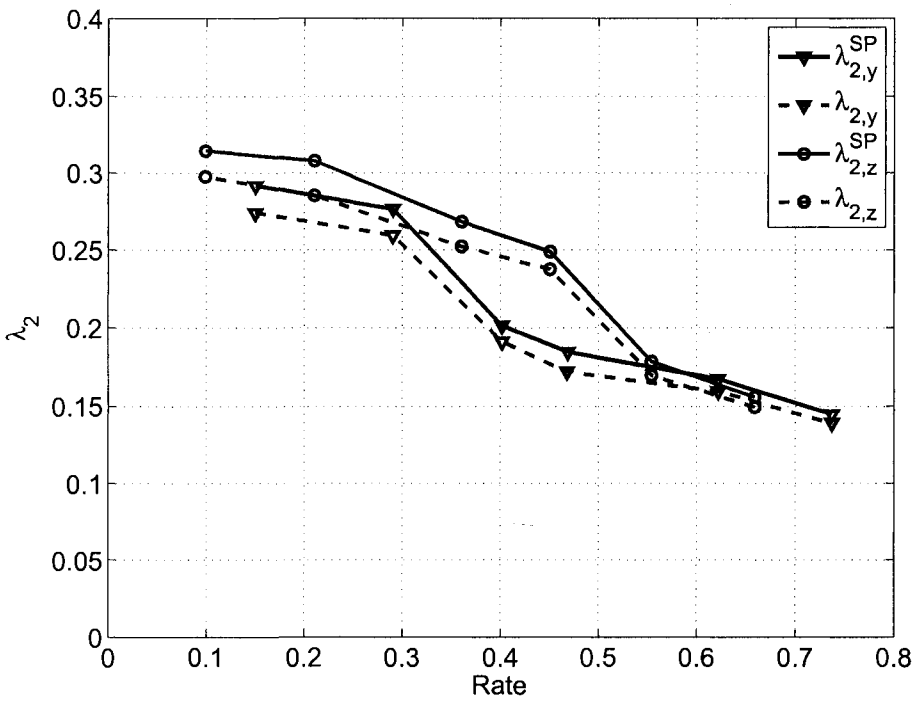


Figure 4.6: Comparison of λ_2 values of the designed codes and λ_2 constraints by the stability condition of the SP decoder given in (4.9) and (4.10).

Chapter 5

Conclusion

In this chapter, we summarize the contributions of this thesis and outline possible future research directions for an interested reader.

5.1 Contributions

The contribution of this thesis is twofold: first, a low complexity LDPC decoder is devised and second, a low complexity LDPC coding method for the Gaussian broadcast channels is proposed.

In Chapter 3, a low complexity decoder for LDPC codes has been proposed. Using the stability analysis, we have showed that the performance of irregular LDPC codes designed for SP decoder with the linear scaling method can be very poor. It was shown that the LSMS decoder forces λ_2 values to be small which prevents LSMS codes to achieve high rates. The proposed method was proved to exhibit a more relaxed stability condition. By code optimization, we showed that our codes have superior performance with respect to both MS and LSMS codes.

In Chapter 4, a method for communicating over a two-user Gaussian broadcast channel which has considerably lower complexity than the method used in [16] was proposed. The previously studied method [16] needs both receivers to have the code of each other. Also, both LDPC codes should be jointly optimized which is a cumbersome task. We showed that using a novel labeling method and the BICM scheme, the complexity can be significantly reduced in a way that each user only requires his own LDPC code and the code design can be done separately for each of the users.

5.2 Future Research

In this section, we present some suggestions for future research.

A Low Complexity LDPC Decoder

- It is quite interesting if one can find the optimal function such that if it is concatenated with a MS check node, it gives the best performance in terms of the gap to the capacity.
- We optimized our codes for a fixed scaling factor (α) and cutoff LLR (x_0). However, one can use different values for each iteration.
- Given a fixed BISO channel, it is interesting if one can propose an approximation to the scaling factor for LSMS decoder. Recent works consider only optimization through density evolution.
- Finding the necessary part of the stability condition theorems for the LSMS decoder and the proposed decoder is a possible direction.

LDPC Codes for Gaussian Broadcast Channels

- Since using binary codes at a high SNR is not efficient, the extension of our proposed method and also [16] to higher level modulation is of great interest.
- We assumed that the CSI is known perfectly at the receivers. Designing codes for fading Gaussian broadcast channel where the fading coefficients are not known to the receivers is a challenging problem. One problem which causes difficulty to this extension is that the capacity region of a general broadcast channel is not known. For general fading processes that are even perfectly known at the receiver, the channel is neither degraded nor more capable.
- Extension of our proposed Gray labeling method to more than two users is possible.

Bibliography

[1] C. E. Shannon, "A mathematical theory of communication," *Bell Sys. Tech. Journal*, vol. 27, pp. 379–423 and 623–656, Oct. 1948.

[2] T. Richardson and R. Urbanke, *Modern Coding Theory*. New York: Cambridge University Press, 2008.

[3] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo-codes. 1," in *Proc. IEEE Int. Conf. Commun. (ICC)*, vol. 2, Geneva, Switzerland, May 1993, pp. 1064–1070.

[4] D. J. C. MacKay and R. M. Neal, "Near Shannon limit performance of low density parity check codes," *Electronics Letters*, vol. 32, Aug. 1996.

[5] D. J. C. MacKay, "Good error-correcting codes based on very sparse matrices," in *Proc. IEEE Int. Symp. on Infor. Theory (ISIT)*, Ulm, Jun./Jul. 1997.

[6] ——, "Good error-correcting codes based on very sparse matrices," *IEEE Trans. Inf. Theory*, vol. 45, no. 2, pp. 399–431, Mar. 1999.

[7] M. Sipser and D. A. Spielman, "Expander codes," *IEEE Trans. Inf. Theory*, vol. 42, pp. 1710–1722, Nov. 1996.

[8] M. G. Luby, M. Mitzenmacher, M. A. Shokrollahi, D. A. Spielman, and V. Stemann, "Practical loss-resilient codes," in *Proc. of the 29th annual ACM symposium on Theory of computing*. New York, NY, USA: ACM, 1997, pp. 150–159.

[9] M. G. Luby, M. Mitzenmacher, M. A. Shokrollahi, and D. A. Spielman, "Efficient erasure correcting codes," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 569–584, Feb. 2001.

[10] ——, "Improved low-density parity-check codes using irregular graphs," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 585–598, Feb. 2001.

[11] R. G. Gallager, *Low Density Parity Check Codes*. MIT Press, 1963.

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

[13] J. Chen and M. P. C. Fossorier, "Density evolution for two improved BP-based decoding algorithms of LDPC codes," *IEEE Commun. Lett.*, vol. 6, no. 5, pp. 208–210, May 2002.

[14] K. Bhattad, V. Rathi, and R. Urbanke, "Degree optimization and stability condition for the min-sum decoder," in *IEEE Infor. Theory Workshop*, Sep. 2007, pp. 190–195.

[15] M. Ramezani, R. Yazdani, and M. Ardakani, "Stability analysis of an improved min-sum decoder," *IEEE Commun. Lett.*, vol. 12, no. 8, pp. 581–583, Aug. 2008.

[16] P. Berlin and D. Tuninetti, "LDPC codes for fading gaussian broadcast channels," *IEEE Trans. Inf. Theory*, vol. 51, no. 6, pp. 2173–2182, Jun. 2005.

[17] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, 2nd ed. New York: Wiley, 2006.

[18] R. G. Gallager, *Information Theory and Reliable Communication.* Wiley, 1968.

[19] J. Wolfowitz, "The coding of messages subject to chance errors," *Illinois J. of Math.*, vol. 1, no. 4, pp. 591–606, Dec. 1957.

[20] S. Lin and D. J. Costello, *Error Control Coding*, 2nd ed. Prentice Hall, 2004.

[21] 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.

[22] H. A. Loeliger, "An introduction to factor graphs," *IEEE Signal Process. Mag.*, vol. 21, no. 1, pp. 28–41, Jan. 2004.

[23] S. M. Aji, G. B. Horn, and R. J. McEliece, "Iterative decoding on graphs with a single cycle," in *Proc. IEEE Int. Symp. on Infor. Theory (ISIT)*, Cambridge, MA, USA, Aug. 1998.

[24] Y. Weiss and W. T. Freeman, "On the optimality of solutions of the max-product belief-propagation algorithm in arbitrary graphs," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 736–744, Feb. 2001.

[25] N. Wiberg, "Codes and decoding on general graphs," Ph.D. dissertation, U. Linköping, Sweden, 1996.

[26] M. A. Shokrollahi, "LDPC Codes: An Introduction," http://www.ipm.ac.ir/IPM/homepage/Amin2.pdf, Digital Fountain Inc., Apr. 2003.

[27] 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.

[28] 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.

[29] M. Hellman and J. Raviv, "Probability of error, equivocation, and the Chernoff bound," *IEEE Trans. Inf. Theory*, vol. 16, no. 4, pp. 368–372, Jul. 1970.

[30] H. D. Pfister and I. Sason, "Accumulate-repeat-accumulate codes: capacity-achieving ensembles of systematic codes for the erasure channel with bounded complexity," *IEEE Trans. Inf. Theory*, vol. 53, no. 6, pp. 2088–2115, Jun. 2007.

[31] 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, Aug. 2004.

[32] O. Etesami and A. Shokrollahi, "Raptor codes on binary memoryless symmetric channels," *IEEE Trans. Inf. Theory*, vol. 52, no. 5, pp. 2033–2051, May 2006.

[33] M. Lentmaier, D. V. Truhachev, K. S. Zigangirov, and D. J. Costello, "An analysis of the block error probability performance of iterative decoding," *IEEE Trans. Inf. Theory*, vol. 51, no. 11, pp. 3834–3855, Nov. 2005.

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

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

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

[37] C. C. Wang, S. R. Kulkarni, and H. V. Poor, "Density evolution for asymmetric memoryless channels," *IEEE Trans. Inf. Theory*, vol. 51, no. 12, pp. 4216–4236, Dec. 2005.

[38] V. Rathi and R. Urbanke, "Density evolution, thresholds and the stability condition for non-binary LDPC codes," in *Communications, IEE Proceedings-*, vol. 152, Dec. 2005, pp. 1069–1074.

[39] S. ten Brink, "Convergence of iterative decoding," *Electronics Letters*, vol. 35, no. 10, pp. 806–808, May 1999.

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

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

[42] J. Chen and M. P. C. Fossorier, "Near optimum universal belief propagation based decoding of low-density parity check codes," *IEEE Trans. Commun.*, vol. 50, no. 3, pp. 406–414, Mar. 2002.

[43] K. Bhattad, "LDPC code design for min-sum based decoding," Texas A&M University, Technical Report WCL-TR-07-102, Mar. 2006.

[44] X. Wei and A. N. Akansu, "Density evolution for low-density parity-check codes under Max-log-MAP decoding," *Electronics Letters*, vol. 37, pp. 1125–1126, Aug. 2001.

[45] J. Bucklew, *Large Deviation Techniques in Decision, Simulation and Estimation.* John Wiley, New York, 1990.

[46] K. Bhattad, "LdpcOpt, LDPC degree optimization for the min–sum decoder," http://lthcwww.epfl.ch/research/bhattad/.

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

[48] T. Cover, "Broadcast channels," *IEEE Trans. Inf. Theory*, vol. 18, no. 1, pp. 2–14, Jan. 1972.

[49] D. Tuninetti and S. Shamai (Shitz), "Gaussian broadcast channels with state information at the receivers," in *Proc. DIMACS Workshop on Network Information Theory*, Mar. Piscataway, NJ, 2003.

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

[51] A. El Gamal, "The capacity of a class of broadcast channels," *IEEE Trans. Inf. Theory*, vol. 25, no. 2, pp. 166–169, Mar. 1979.

[52] P. Bergmans, "Random coding theorem for broadcast channels with degraded components," *IEEE Trans. Inf. Theory*, vol. 19, no. 2, pp. 197–207, Mar. 1973.

[53] T. M. Cover, "Comments on broadcast channels," *IEEE Trans. Inf. Theory*, vol. 44, no. 6, pp. 2524–2530, Oct. 1998.

[54] I. Csiszár and J. Körner, *Information Theory: Coding Theorems for Discrete Memoryles Systems.* Academic Press, 1981.

[55] P. Bergmans, "A simple converse for broadcast channels with additive white Gaussian noise (corresp.)," *IEEE Trans. Inf. Theory*, vol. 20, no. 2, pp. 279–280, Mar. 1974.

[56] R. G. Gallager, "Capacity and coding for degraded broadcast channels," *Probl. Infor. Transm.*, vol. 10, no. 3, pp. 185–193, Jul. 1974.

[57] K. Price and R. Storn, "Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, Dec. 1997.

[58] S. Y. Le Goff, "Signal constellations for bit-interleaved coded modulation," *IEEE Trans. Inf. Theory*, vol. 49, no. 1, pp. 307–313, Jan. 2003.

[59] J. Hou, P. H. Siegel, L. B. Milstein, and H. D. 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, Sep. 2003.