TECHNISCHE UNIVERSITÄT MÜNCHEN
LEHRSTUHL FÜR NACHRICHTENTECHNIK
Prof. Dr.sc.tech. Gerhard Kramer

Master's Thesis

Probabilistic Shaping with Low-Density Graph Codes and Message Passing

Vorgelegt von:
Ömer Pepeoğlu

München, June 2023

Betreut von:
M.Sc. Constantin Runge
M.Sc. Thomas Wiegart

Master's Thesis am
Lehrstuhl für Nachrichtentechnik (LNT)
der Technischen Universität München (TUM)
Titel : Probabilistic Shaping with Low-Density Graph Codes and Message Passing
Autor : Ömer Pepeoğlu

Ömer Pepeoğlu
Technische Universität München
Lehrstuhl für Nachrichtentechnik
Theresienstr. 90
80333 München
omer.pepeoglu@tum.de

Ich versichere hiermit wahrheitsgemäß, die Arbeit bis auf die dem Aufgabensteller bereits bekannte Hilfe selbständig angefertigt, alle benutzten Hilfsmittel vollständig und genau angegeben und alles kenntlich gemacht zu haben, was aus Arbeiten anderer unverändert oder mit Abänderung entnommen wurde.

München, June 30, 2023

Ort, Datum (Ömer Pepeoğlu)

Acknowledgements

There are many people who contributed directly or indirectly to this thesis, and I would like to acknowledge them. First of all, I am grateful to Constantin Runge and Thomas Wiegart for proposing the thesis topic, fruitful discussions during the regular meetings, and all their feedback on the thesis. I wish them the best with the rest of their doctoral research and career. I would like to thank Prof. Gerhard Kramer for his suggestions regarding the thesis and the inspiring research environment provided at the chair.

I had the opportunity to attend many exciting courses during my studies at TUM. I appreciate all the professors and doctoral researchers who helped me acquire the skills that led to this thesis. Surviving in a new academic environment could be challenging. I cannot find words to thank my professors at İTÜ for preparing me to make this transition seamless. I am always proud to “bee” engineer.

Considering the pandemic in the early days of my master’s, this whole period would definitely have been much more difficult without my friends. First and foremost, I am grateful to my beloved friend Kamil Karaçuha for everything. He was with me at every step of this period, always made me feel his support, and showed how a distance-independent friendship is possible. I had the opportunity to meet many bright people in Munich and would like to thank them for all the good memories. I am especially grateful to my MSCE colleagues, Doğukan Atik, Eren Küçükali, Gülde Akkuzu, Oğulcan Çakıcı, Özgün Karağöz, and Utku Ergin, in alphabetical order, for their both academic and social solidarity. I hope all my friends whose names I cannot include here will forgive me.

All this would have been impossible without my beloved family. I appreciate my dear parents Alican & Hediye Pepeoğlu for their endless love, unconditional support, and encouraging me in every decision I make. Last but certainly not least, I am grateful to my beloved sister Zeynep Pepeoğlu for never forgetting me among all her responsibilities and making me feel all her love. I hope we can spend more time together in the future.

Munich, June 2023 Ömer Pepeoğlu

Contents

List of Abbreviations

AWGN	additive white Gaussian noise.
BCH	Bose-Chaudhuri-Hocquenghem.
BER	bit error rate.
BP	belief propagation.
bpcu	bits per channel use.
BPGD	belief propagation guided decimation.
CCSI	channel coding with side information.
DPC	dirty paper coding.
FER	frame error rate.
i.i.d.	independent and identically distributed.
LDGM	low-density generator matrix.
LDPC	low-density parity-check.
LLR	log-likelihood ratio.
MPA	message passing algorithm.
OOK	on-off keying.
SC	spatial coupling.
SCSI	source coding with side information.
SNR	signal-to-noise ratio.
TP	truthiness propagation.

Abstract

The compound LDGM/LDPC codes are shown to attain the information-theoretic limits of the Gelfand-Pinsker and Wyner-Ziv problems when optimal encoding and decoding are employed. The sparse and graphical structure of this code motivates it to be implemented with message passing algorithms. In this thesis, an encoding and decoding message passing algorithm is proposed for the compound code by inspiring the belief propagation and truthiness propagation algorithms. Initial simulations show that the encoding algorithm can correctly encode half of the frames, and the decoding algorithm can successfully decode the correctly encoded frames. The drawbacks of the encoding algorithm are compensated by concatenating an outer code. With the coding scheme provided in the literature for the compound code, it is shown that a uniformly distributed message input is shaped into a non-uniformly distributed channel input, which motivates probabilistic shaping applications. Although it is widely claimed in the literature that this scheme should deliver a good performance, the scheme is not competitive compared to other schemes, e.g., to polar codes.

1 Introduction

Claude Shannon's [Sha48] seminal work on information theory laid the foundation for modern communications systems. Information theory provides us with a profound understanding of how information is encoded, transmitted, and decoded. It enables to design efficient and reliable communications systems. With the contributions of Shannon and subsequent researchers, information theory continues to inspire innovations and drive advancements in the field of communications, making it an indispensable discipline in our increasingly interconnected world.

Shannon's channel capacity refers to the maximum achievable data rate over a communication channel while maintaining reliable transmission in the presence of noise and interference. This capacity can be achieved through the utilization of certain codes such as low-density parity-check (LDPC) codes [Gal62] and polar codes [Ari09]. For uniformly distributed message bits, these codes usually generate codewords that are also uniformly distributed. However, Shannon's findings revealed that the optimal input distribution for achieving the maximum information rate for an asymmetric channel is typically non-uniform. Therefore, it is of interest to shape the probability distributions of the channel inputs to increase spectral efficiency while making use of these kinds of codes. The concept is generally referred to as probabilistic shaping and is the main motivation of this thesis.

On the other hand, multi-user information theory investigates the fundamental limits and efficient strategies when multiple users concurrently access and share information over a common communication channel. Two well-known problems in this domain are the Gelfand-Pinsker [GP80] and Wyner-Ziv [WZ76] problems. The Gelfand-Pinsker problem considers a scenario where the encoder has channel side information while the decoder has not, and its special case enables probabilistic shaping. The Gelfand-Pinsker problem for additive white Gaussian noise (AWGN) channels is also known as dirty paper coding (DPC) [Cos83] and finds wide interest in the literature [EtB05]. The Wyner-Ziv problem is dual to the Gelfand-Pinsker problem and pertains to the challenge of compressing data while the decoder has a side information that is correlated with the source.

Wainwright and Martinian [WM09] proposed a code structure by combining low-density generator matrix (LDGM) and LDPC codes. They showed that such a code can attain the information-theoretic limits of the Gelfand-Pinsker and Wyner-Ziv problems under optimal encoding and decoding. Since implementing the optimal encoding and decoding methods is impractical, finding a practical way is of interest for this structure. The sparse and graphical structure of this code holds the potential to implement message passing algorithms (MPAs) and presents an open research question. There are very few studies that attempt to address this issue. [WMM10] proposes the belief propagation guided decimation (BPGD) while [KVNP14] combines the BPGD with spatial coupling (SC). [KT08, WH09] approach the problem in a way that adapts the DPC concepts.

In this thesis, we propose two MPAs to the LDGM/LDPC compound code for both encoding and decoding. We discuss the performance of our MPAs and come up with solutions to their drawbacks. Thanks to the coding scheme provided with the compound code [WM09], we show how the channel input distribution is non-uniformly shaped for a uniformly distributed message input, and the average symbol energy is decreased for on-off keying (OOK) modulated symbols. The rest of the thesis is organized as follows.

- Chapter 2 provides the preliminaries, including source and channel coding, low-density codes, MPAs, and probabilistic shaping. It also presents the main notation used throughout the thesis.
- Chapter 3 explains channel coding with side information (CCSI) (Gelfand-Pinsker problem), lossy source coding with side information (SCSI) (Wyner-Ziv problem), and relates to probabilistic shaping.
- Chapter 4 introduces the LDGM/LDPC compound code, discusses the belief propagation (BP) and truthiness propagation (TP) algorithms, and presents our proposed algorithms.
- Chapter 5 provides the simulation results of our algorithms and discusses the solutions to their observed shortcomings.
- Chapter 6 concludes the thesis and points to open questions for future work.

2 Preliminaries

This chapter discusses preliminaries for the thesis and defines the notation. First, we will examine the basics of source and channel coding from a theoretical perspective and define the concepts that will be encountered throughout the thesis. Then, we will discuss low-density codes and introduce Tanner graphs which will be used in the main sections. Next, we will explain message passing algorithms which are commonly performed on those graphs. Finally, we will discuss probabilistic shaping as a motivating application of this thesis.

2.1 Source Coding

Source coding tries to represent information with as few bits as possible. For this, Shannon defined the entropy of an information source [Sha48].

Definition 2.1.1 (Entropy)**.** *Let X be a discrete random variable with alphabet $\mathcal{X}$. The entropy in bits of X is defined by*

$$H(X) = -\sum_{x \in \mathcal{X}} P_X(x) \log_2 P_X(x), \tag{2.1}$$

where $P_X(x)$ refers to the probability mass function of the distribution of X.

The entropy of an information source gives an idea about the uncertainty that the random variable has. A simple example can be given to make it more understandable.

Example 2.1.1 (Coin Toss)**.** *Consider a fair coin toss experiment. This experiment has only two possible outcomes: heads and tails. The sample space for this can be represented as $\Omega = \{H, T\}$. Let X be a discrete random variable that maps from Ω into the alphabet $\mathcal{X} = \{0, 1\}$ with $X(H) = 0$ and $X(T) = 1$. Then, $P_X(0) = P_X(1) = 1/2$ as we have a fair coin toss. From (2.1), the entropy for a fair coin toss experiment is $H(X) = 1\,\mathrm{bit}$.*

By generalizing Example 2.1.1, we define the binary entropy function.

Definition 2.1.2 (Binary Entropy Function)**.** *Let X be a Bernoulli distributed random variable with parameter p, i.e., $X \sim \mathrm{Bern}(p)$, $P_X(0) = 1 - p$, and $P_X(1) = p$. The binary entropy function $H_2(p)$ is defined as*

$$H_2(p) = H(X) = -p \log_2 p - (1-p)\log_2(1-p). \tag{2.2}$$

The binary entropy function is shown in Figure 2.1. Note that the entropy is maximal when $p = 0.5$. In other words, this is the case when the uncertainty is highest, just like the fair coin toss experiment in Example 2.1.1.

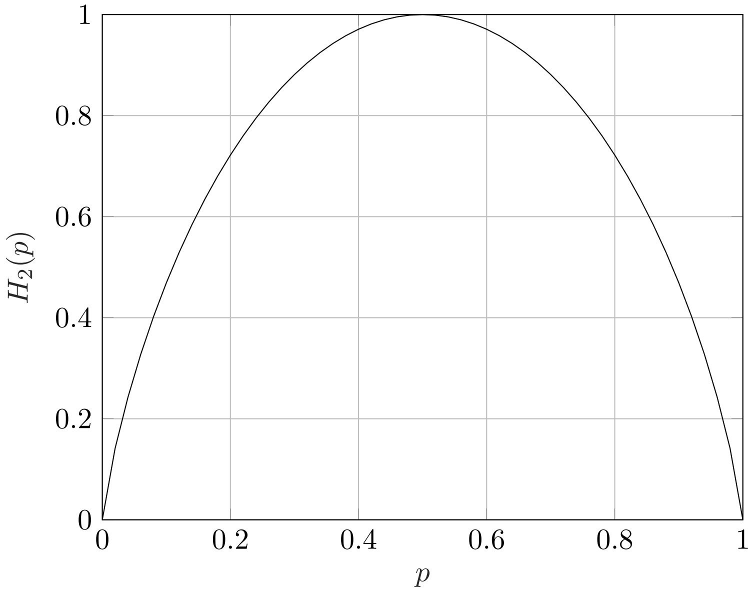


Figure 2.1: The binary entropy function.

Definition 2.1.3 (Conditional Entropy [CT06])**.** *Let X and Y be two discrete random variables with alphabet $\mathcal{X}$ and $\mathcal{Y}$, respectively. The conditional entropy of Y given X is defined by*

$$\begin{aligned} H(Y|X) &= \sum_{x \in \mathcal{X}} P_X(x) H(Y|X=x) \\ &= -\sum_{x \in \mathcal{X}} P_X(x) \sum_{y \in \mathcal{Y}} P_{Y|X}(y|x) \log_2 P_{Y|X}(y|x) \\ &= -\sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} P_{XY}(x,y) \log_2 P_{Y|X}(y|x), \end{aligned} \tag{2.3}$$

where $P_{Y|X}(\cdot)$ and $P_{XY}(\cdot)$ denote the conditional and joint probabilities, respectively.

Definition 2.1.4 (Mutual Information [CT06])**.** *Let X and Y be two discrete random variables with alphabet $\mathcal{X}$ and $\mathcal{Y}$, respectively. The mutual information between X and Y is defined by*

$$I(X;Y) = \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} P_{XY}(x,y) \log_2 \frac{P_{XY}(x,y)}{P_X(x)P_Y(y)}. \tag{2.4}$$

It can also be shown that $I(X;Y) = H(X) - H(X|Y) = H(Y) - H(Y|X)$. We can now relate the entropy to source coding.

Lossless Source Coding

Lossless source coding, also known as entropy coding, aims to reduce the redundancy in representing data without any loss of information. This technique plays a crucial role in various applications such as data compression and efficient storage and transmission of digital information.

Shannon's theorem provides a mathematical foundation for lossless source coding [Sha48]. It states that for any discrete memoryless source with a finite alphabet of symbols and a known probability distribution, one can construct a code that can represent the source's output with an average codeword length arbitrarily close to the source's entropy. The entropy of a source quantifies the average amount of information contained in each symbol and serves as an upper bound on the average codeword length required to represent the source optimally. The theorem establishes a theoretical limit for lossless source coding, known as the entropy bound, which states that no code can achieve an average codeword length lower than the entropy of the source. In other words, it is impossible to compress a source beyond its inherent information content.

Several practical lossless source coding techniques have been developed based on Shannon's theorem. One prominent example is the Huffman coding algorithm, which constructs variable-length prefix codes by assigning shorter codewords to more frequently occurring symbols and longer codewords to less frequent symbols [Huf52]. Huffman coding is widely used in various applications, including image, audio, and video compression, as well as in file compression formats such as ZIP.

Lossy Source Coding

For this work, lossy source coding is more relevant. It is an approach that aims to compress data while allowing the loss of some information, thus achieving higher compression ratios than lossless compression. Shannon's rate-distortion theorem provides a theoretical foundation for lossy source coding by establishing a trade-off between the rate of compression and the distortion introduced in the reconstructed data [Sha59].

The block diagram of the rate-distortion problem is depicted in Figure 2.2. A discrete memoryless source $P_X(\cdot)$ produces a sequence X^n observed by an encoder, where $X^n = (X_1, X_2, \ldots, X_n)$. There, X^n is quantized to a sequence represented by an index W. Then, the decoder reconstructs the sequence $\hat{X}^n$ based on the received index W. The average distortion is required to be at most D, i.e., $\mathbb{E}[d^n(X^n, \hat{X}^n)] \leq D$, where $\mathbb{E}[\cdot]$ denotes the expectation operator, $d^n(X^n, \hat{X}^n) = \frac{1}{n}\sum_{i=1}^{n} d(X_i, \hat{X}_i)$ for some distortion function $d(X, \hat{X})$, and D is some specified value [Sha59].

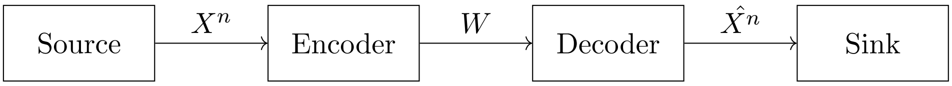


Figure 2.2: Block diagram of the rate-distortion problem.

Definition 2.1.5 (Hamming Distortion)**.** *The Hamming distortion is given by*

$$d(x, \hat{x}) = \begin{cases} 0 & \text{if } x = \hat{x} \\ 1 & \text{if } x \neq \hat{x} \end{cases}. \tag{2.5}$$

The Hamming distortion is the distortion used hereinafter unless stated otherwise.

Definition 2.1.6 (Rate-Distortion Function)**.** *The rate-distortion function of an independent and identically distributed (i.i.d.) source X is given by*

$$R(D) = \min_{P_{\hat{X}|X}(\hat{x}|x):\mathbb{E}[d(X,\hat{X})]\leq D} I(X; \hat{X}). \tag{2.6}$$

The rate-distortion function is equal to the minimum rate required to achieve a distortion level D or lower.

Example 2.1.2 (Rate-Distortion Function of a Bernoulli Source)**.** *The rate-distortion function of a Bernoulli source $X \sim \mathrm{Bern}(p)$ can be expressed as*

$$R(D) = H_2(p) - H_2(D). \tag{2.7}$$

The proof of (2.7) can be found in [CT06, Chapter 10]. If we consider a uniformly distributed source, i.e., $X \sim \mathrm{Bern}(0.5)$, the resulting rate-distortion function will be $R(D) = 1 - H_2(D)$ and it is plotted in Figure 2.3. The curve in the figure will be of *interest for later evaluations of coding schemes. Unfortunately, the bound is not perfectly achievable in practice due to factors such as finite blocklengths, computational complexity in implementation, and other constraints imposed by the scenario. They lead to deviations from the bound and require additional considerations and trade-offs. Therefore, we will aim to get as close to the bound as possible while considering the other aspects of our problems in the following chapters.*

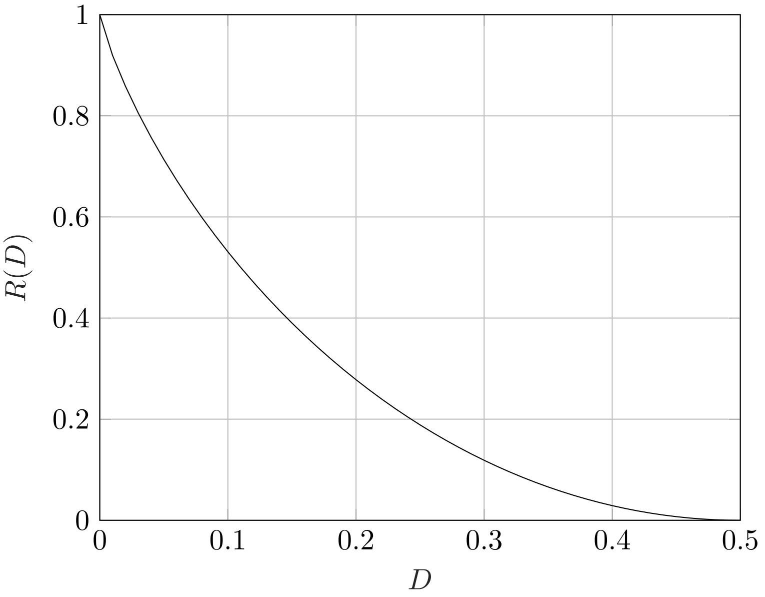


Figure 2.3: The rate-distortion function for a source $X \sim \text{Bern}(0.5)$.

2.2 Channel Coding

Channel coding is a vital aspect of communication systems. There, the objective is to transmit data reliably over noisy channels. When data is transmitted, it is susceptible to various sources of interference and noise that can corrupt the signal. Channel coding introduces redundancy, enabling the receiver to detect and correct errors. In this respect, it can be thought of as a dual to source coding.

Figure 2.4 depicts the block diagram of the channel coding problem. A source message W is mapped to a channel input X^n by an encoder. A discrete memoryless channel is represented as the conditional probability distribution $P_{Y|X}(\cdot)$, and Y^n is the channel output. For the decoder output $\hat{W}$, the objective is to find the maximum rate, which is called the capacity, such that the probability of $W \neq \hat{W}$ is arbitrarily close to zero.

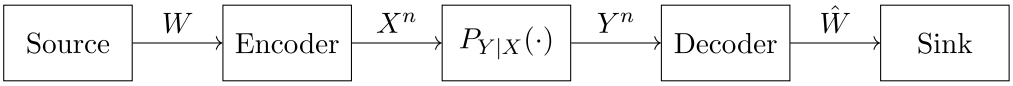


Figure 2.4: Block diagram of the channel coding problem.

Shannon's channel capacity sets the foundation for understanding the importance of channel coding and quantifies the maximum achievable data transmission rate over a noisy communication channel [Sha48]. It is determined by the mutual information between the transmitted and received signals, representing the amount of information that can be reliably transmitted through the channel.

Definition 2.2.1 (Channel Capacity [CT06])**.** *The channel capacity is defined as*

$$C = \max_{P_X(x)} I(X;Y), \tag{2.8}$$

where X and Y represent the transmitted and received signals, respectively.

As an example, the AWGN channel is shown in Figure 2.5. There, $N \sim \mathcal{N}(0, \sigma^2)$ represents the channel noise and $\mathcal{N}(0, \sigma^2)$ denotes the Gaussian distribution with mean 0 and variance σ^2.

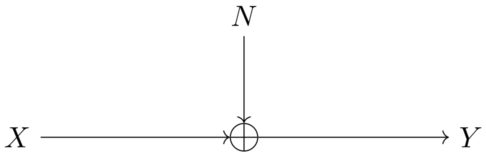


Figure 2.5: AWGN channel.

Definition 2.2.2 (Signal-to-Noise Ratio)**.** *The signal-to-noise ratio (SNR) represents the ratio of the signal power to the noise power and is given by*

$$SNR = \frac{P_{signal}}{P_{noise}} \tag{2.9}$$

$$= \frac{\mathbb{E}[|X|^2]}{\mathbb{E}[|N|^2]}, \tag{2.10}$$

where (2.10) is relevant when the signal and noise are shown by the random variables X and N, respectively.

The SNR influences the channel capacity and characterizes the quality of the communication channel, with higher SNR values indicating a better signal quality relative to the noise. As the SNR increases, the channel capacity also increases, allowing for higher data rates. This relationship can be captured by Shannon's capacity formula for the AWGN [Sha48] and shown by $C = \frac{1}{2}\log_2(1+\text{SNR})$ in bits per channel use (bpcu), see Figure 2.6, where dB stands for decibel and $\text{SNR}_{\text{dB}} = 10\log_{10}(\text{SNR})$. Intuitively, when the SNR is low, the channel capacity decreases, and the data rate must be kept low to maintain reliable communication. As the SNR improves, the capacity increases, allowing for higher data rates without compromising the error performance.

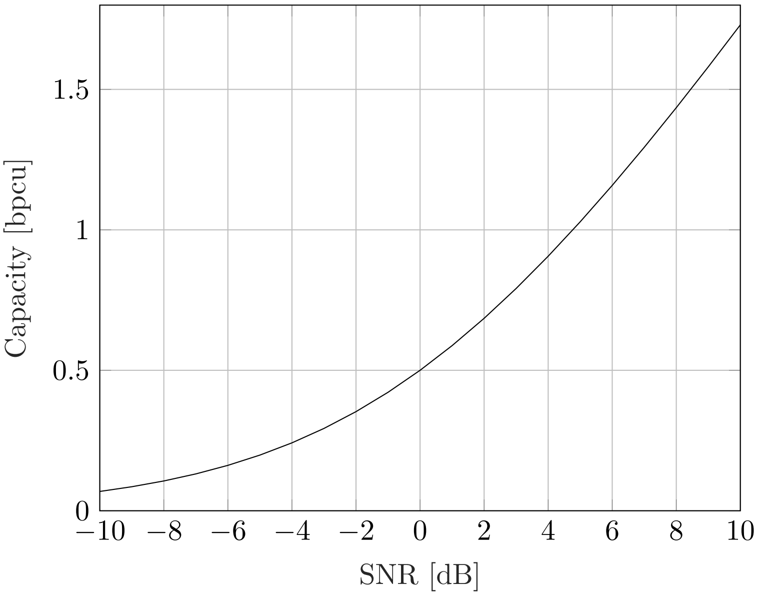


Figure 2.6: Shannon's channel capacity for AWGN with Gaussian inputs.

The channel capacity serves as a benchmark for evaluating the performance of various coding schemes, allowing for the optimization of communication systems in practical scenarios. While it provides an essential measure of the maximum data rate, ensuring reliable transmission requires additional techniques such as error correction coding. By adding redundancy to the original message, error correction codes provide a means to detect and potentially recover from errors at the receiver. Linear codes are a commonly used class of error correction codes.

Definition 2.2.3 (Linear Code)**.** *A linear code $\mathcal{C}$ is defined as a k-dimensional subspace of $\mathbb{F}_q^n$, where $\mathbb{F}_q^n$ is a finite field with n elements of alphabet size q.*

In this work, we will be working with the binary numbers, i.e., $\mathbb{F} = \mathbb{F}_2$.

Linear codes have two essential characterizations: the generator and parity-check matrices.

Definition 2.2.4 (Generator Matrix)**.** *The generator matrix* $\mathbf{G}$ *of a linear code* $\mathcal{C}$ *is a* $k \times n$ *matrix over* $\mathbb{F}$ *whose row space generates the codewords of* $\mathcal{C}$*, i.e.,* $\mathbf{G} \in \mathbb{F}^{k \times n}$ *and* $c^n \in \mathcal{C} \Leftrightarrow \exists u^k \in \mathbb{F}^k : c^n = u^k \cdot \mathbf{G}$*. If* $\mathbf{G}$ *has the form* $(\mathbf{I}_k \mid \mathbf{A})$*, where* $\mathbf{I}_k$ *is the* $k \times k$ *identity matrix, it is called a systematic generator matrix.*

Definition 2.2.5 (Parity-Check Matrix)**.** *The parity-check matrix* $\mathbf{H}$ *of a linear code* $\mathcal{C}$ *is an* $(n-k) \times n$ *matrix over* $\mathbb{F}$ *and the product of any codeword of* $\mathcal{C}$ *and the transpose of* $\mathbf{H}$ *yields the zero vector, i.e.,* $\mathbf{H} \in \mathbb{F}^{(n-k)\times n}$ *and* $c^n \in \mathcal{C} \Leftrightarrow c^n \cdot \mathbf{H}^T = 0^{n-k}$*.*

It can also be shown that $\mathbf{G} \cdot \mathbf{H}^T = \mathbf{0}$. If the product of the received codeword and the parity-check matrix is nonzero, it indicates the presence of errors. This is called syndrome and provides information to correct the errors during decoding.

Definition 2.2.6 (Syndrome)**.** *Let* $\mathbf{H}$ *be a parity-check matrix for the code* $\mathcal{C}$ *and* $r^n \in \mathbb{F}^n$*. The syndrome of* r^n *is defined as* $s^{n-k} = r^n \cdot \mathbf{H}^T$*. Note that* $s^{n-k} = 0^{n-k} \Leftrightarrow r^n \in \mathcal{C}$*.*

Example 2.2.1 (Repetition Code)**.** *Let* $\mathcal{C}$ *be the linear code corresponding to the generator matrix*

$$\mathbf{G}_{RP} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \end{pmatrix}. \tag{2.11}$$

It means repeating an information vector of length $k = 1$ *to form a codeword of length* $n = 5$*. Then the corresponding parity-check matrix* $\mathbf{H} \in \mathbb{F}^{4\times 5}$ *can be shown as*

$$\mathbf{H}_{RP} = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{pmatrix}. \tag{2.12}$$

The rate of this code is given by $R = k/n = 1/5$*. Note again that any received vector other than* $r^5 = (0, 0, \dots, 0)$ *and* $r^5 = (1, 1, \dots, 1)$ *gives* $s^4 \neq 0^4$*, which indicates an error.*

Remark. It can be easily seen in Example 2.2.1 how simple error correction can be made. If $d^n(c^n, r^n) < 1/2$, the majority bit will give the information even if erroneous bits occur. A comprehensive study of various error control coding techniques, including syndrome decoding, can be found in [LC01].

2.3 Low-Density Codes and Tanner Graphs

LDPC codes are a class of error correcting codes that have attracted significant attention in the field of communication systems. They were first introduced by Gallager [Gal62] in the 1960s and have emerged as a powerful tool for achieving reliable communication over noisy channels. Gallager's seminal work laid the foundation for LDPC codes and their subsequent advancements. Inspired by his ideas, researchers such as MacKay and Neal made significant contributions to the theory and practical aspects of LDPC codes [MN97].

Construction of LDPC codes involves designing a parity-check matrix. There are several methods for constructing LDPC codes, including random constructions [Gal62] and protograph-based constructions [Tho03]. Protograph-based constructions provide a systematic approach to constructing LDPC codes. A protograph is a small, regular graph that represents the structure of the LDPC code. By repeating and connecting protographs, larger LDPC codes are formed. For in-depth discussions on protograph-based constructions, we refer to [Tho03]. The choice of LDPC code construction method depends on various factors such as error correction performance and complexity constraints. Each construction technique has its advantages and trade-offs. Further details on the code construction are beyond the scope of this thesis.

The sparsity property plays a crucial role in LDPC codes, particularly in the context of decoding algorithms. LDPC codes are designed with a sparse parity-check matrix, where only a small fraction of the elements are nonzero. This sparsity is vital because LDPC codes are typically decoded using BP algorithms, see Section 2.4, rather than the computationally expensive maximum likelihood decoding. In dense matrices with a high density of nonzero elements, the cycles are encountered, and BP becomes computationally infeasible due to the extensive message passing required. Therefore, the sparsity property of LDPC codes is essential for enabling the practical use of BP decoding, providing an efficient and feasible approach for error correction. An amazing animation of how LDPC codes work can be watched in [Art18].

Tanner Graphs

Tanner graphs [Tan81], named after Stephen Tanner, are graphical representations that provide valuable insights about LDPC codes. They allow for an understanding of the code structure, decoding algorithms, and performance limits.

Example 2.3.1 (Tanner Graph of a Parity-Check Matrix)**.** *Let $\mathcal{C}$ be the linear code* *corresponding to the systematic parity-check matrix*

$$\mathbf{H} = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 \end{pmatrix}. \tag{2.13}$$

From Definition 2.2.5, $n-k=3$ and $n=7$. Thus, the Tanner graph of the parity-check matrix $\mathbf{H}$ will have $n-k=3$ check nodes and $n=7$ variable nodes. Each row of $\mathbf{H}$ represents a check node, while each column of $\mathbf{H}$ represents a variable node. For every nonzero entry in $\mathbf{H}$, an edge is created between the corresponding check and variable nodes, see Figure 2.7. Note that all check nodes sum to zero if and only if $z^7 \in \mathcal{C}$, e.g., the node f_1 "checks" if $z_1 \oplus z_5 \oplus z_6 \oplus z_7 = 0$, where $\oplus$ denotes the modulo 2 addition.

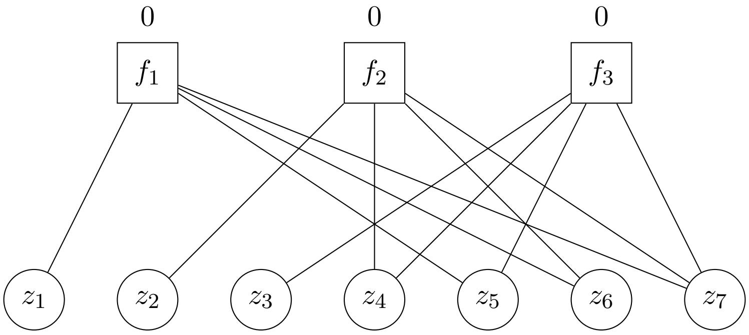
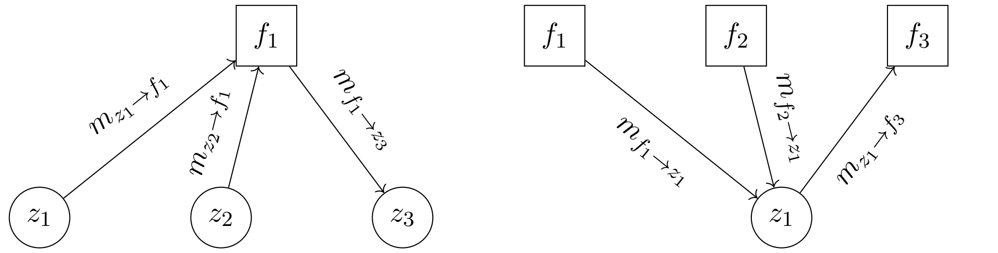


Figure 2.7: The Tanner graph of the parity-check matrix given by (2.13).

2.4 Message Passing Algorithms

MPAs play a crucial role in decoding LDPC codes. They leverage the graphical representation of LDPC codes to iteratively exchange messages between variable nodes and check nodes. In this section, we discuss them verbally. In the following chapters, we will delve into the specific examples of MPAs and examine them algorithmically.

The BP algorithm, also known as sum-product algorithm [KFL01], is a fundamental MPA used for the decoding of LDPC codes. It operates on the Tanner graph. There, each variable node sends a message to its connected check nodes. This message represents the belief or probability distribution of the variable's value based on the received channel information. The check nodes then process the received messages and compute their own messages, representing the satisfaction of the associated parity-checks. These check node messages are then sent back to the variable nodes, see Figure 2.8. Upon receiving check node messages, variable nodes update their beliefs by combining the information from the received messages. This iterative process continues until a certain convergence criterion is met or a maximum number of iterations is reached. The BP algorithm in LDPC decoding is based on the assumption that the code is constructed to have a low-density structure, meaning that only a small number of variables participate in each parity-check equation, see Section 2.3.



(a) From the check node to variable node step, e.g., the check node f_1 receives the messages from the variable nodes z_1 and z_2, then generates the message $m_{f_1 \to z_3}$ to be sent to the variable node z_3.

(b) From the variable node to check node step, e.g., the variable node z_1 receives the messages from the check nodes f_1 and f_2, then generates the message $m_{z_1 \to f_3}$ to be sent to the check node f_3.

Figure 2.8: Message passing steps in BP algorithm.

Generalizing from BP, there are many variants of MPAs that follow a similar framework. These algorithms include min-sum algorithm [ZZB05], generalized BP [YFW$^+$03], and TP [Reg09], among others. They differ in the way messages are computed and updated, but they share the underlying idea of exchanging information between variable nodes and check nodes iteratively. MPAs have been widely adopted in various applications such as wireless communications and storage systems, due to their effectiveness and low complexity compared to alternative decoding methods.

2.5 Probabilistic Shaping

Probabilistic shaping is a technique that aims to improve the performance and efficiency of data transmission in communications systems. It involves designing the probability distribution of the transmitted symbols to maximize the achievable information rate and minimize the error probability. In traditional modulation schemes with uniform signaling, all symbols are transmitted with equal probability. However, the performance of the system can be significantly improved by assigning higher probabilities to certain symbols.

Definition 2.5.1 (On-Off Keying Modulation). *OOK is a digital modulation scheme* *where the information is encoded by using two levels of amplitude. Let $m(t)$ be the binary message signal. The OOK modulated signal is given by*

$$s(t) = \begin{cases} 0 & \textit{if } m(t) = 0 \\ A & \textit{if } m(t) = 1 \end{cases}, \tag{2.14}$$

where A is the amplitude of the carrier signal.

We will assume that the amplitude of the carrier signal $A = 1$ hereinafter unless stated otherwise.

Example 2.5.1 (Probabilistic Shaping with OOK Modulation)**.** *Let $X \sim \mathrm{Bern}(0.5)$ represent the symbols in an OOK scheme. Then the average symbol energy spent for the transmission can be found by*

$$E_s = \mathbb{E}[|X|^2] = \frac{1}{2} \cdot \left(0^2 + 1^2\right) = \frac{1}{2}. \tag{2.15}$$

If we can allocate a higher probability to the zero symbol for the same message input, e.g., $P_X(0) = 3/4$, the average symbol energy spent for the transmission will decrease to

$$E_s = \mathbb{E}[|X|^2] = \frac{3}{4} \cdot 0^2 + \frac{1}{4} \cdot 1^2 = \frac{1}{4}. \tag{2.16}$$

Probabilistic shaping finds applications in various communications systems. In fiber-optic communication, probabilistic shaping has been shown to increase the achievable data rates over long-haul optical links [BSS17]. In satellite communications, it enables higher throughput and improved spectral efficiency, which is crucial for delivering high-bandwidth services to remote areas [EA20]. One area of interest is the investigation of different shaping techniques and their performance characteristics. Researchers are exploring various shaping distributions and optimization algorithms to find the best probabilistic shaping schemes for different channel conditions [RFY$^+$17]. Ongoing research aims to explore different shaping techniques, integrate shaping with advanced modulation schemes, and develop low-complexity algorithms for practical implementations.

3 Coding with Side Information

In this chapter, we introduce two information theory concepts that are of interest to us for this thesis and relate to probabilistic shaping. We will start with discussing channel coding with side information, which explores strategies to enhance communication over noisy channels by exploiting side information available to the encoder. Then, we introduce source coding with side information, which investigates efficient compression techniques when additional information is available to the decoder.

3.1 Channel Coding with Side Information

Channel coding with side information (CCSI), also known as the Gelfand-Pinsker problem, was initially introduced by Israel Gelfand and Mark Pinsker [GP80] in 1980. In their seminal work, they addressed the scenario where the transmitter has the side information with the message to be sent, while the receiver lacks access to this side information during decoding. They investigated the fundamental limits and optimal coding strategies for reliable communication under these circumstances. CCSI over AWGN with Gaussian interference is known as dirty paper coding (DPC), and it enables efficient transmission over channels corrupted by interference or noise by leveraging knowledge about the interference at the transmitter.

The binary CCSI problem plays a major role for us since we will be dealing with this problem in the following chapters. By quantizing the side information with the encoder and transmitting the quantization error with OOK modulated symbols, the channel input distribution can be shaped, and the average symbol energy is decreased, which motivates probabilistic shaping discussed in Section 2.5.

In the Gelfand-Pinsker problem, the side information is available only at the encoder. This poses a challenge as the receiver must decode the message solely based on the received signal without any direct knowledge of the side information. Figure 3.1 depicts the block diagram of the binary Gelfand-Pinsker problem discussed in this section [WM09]. Let $\mathcal{C} \subset \mathbb{F}^n$ be a linear code of rate $R = k/n$. We encode a message $M^k \in \mathbb{F}^k$ into a codeword

$X^n \in \mathcal{C}$. There is side information $Z^n \in \mathbb{F}^n$ only available to the encoder but not the decoder. The channel noise is shown by $W^n \in \mathbb{F}^n$ with each $W_i \sim \text{Bern}(\delta)$. Then the output at the decoder will be $Y^n = X^n \oplus Z^n \oplus W^n$. We also have a constraint on the average weight of the codeword such that $\frac{1}{n}\sum_{i=1}^{n} X_i \leq p$ and $\delta < p \leq 1/2$.

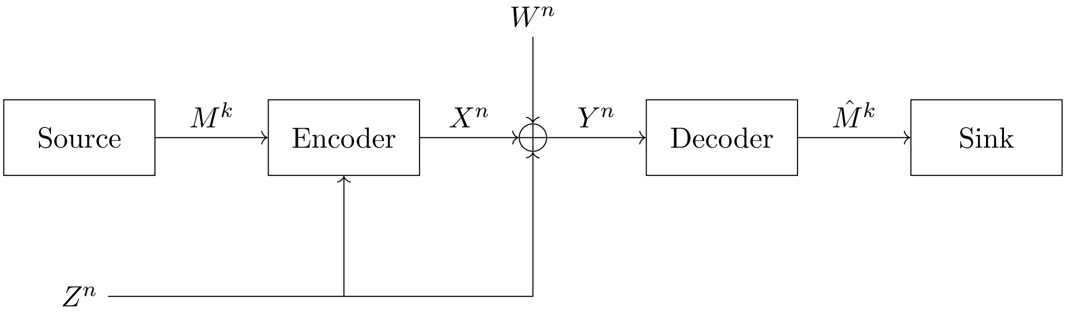


Figure 3.1: Block diagram of the binary CCSI (Gelfand-Pinsker) problem.

Definition 3.1.1 (Capacity of the Gelfand-Pinsker Problem [GP80])**.** *The capacity of the binary Gelfand-Pinsker problem is expressed by*

$$R_{GP}(p, \delta) = H_2(p) - H_2(\delta). \tag{3.1}$$

3.2 Source Coding with Side Information

Lossy source coding with side information (SCSI), also known as the Wyner-Ziv problem, addresses the problem of compressing data when the decoder has access to side information that is correlated with the source. In classical source coding, the encoder has full knowledge of the source data and aims to minimize the number of bits required to represent it accurately at the decoder. However, there may be scenarios where the decoder has some additional information about the source. Then it is possible to exploit this side information to achieve better compression performance.

The SCSI problem was formulated by Aaron Wyner and Jacob Ziv [WZ76] in the 1970s. The encoder's objective is to generate a compressed representation of the source that, together with the side information available at the decoder, enables the reconstruction of the source with limited distortion.

Figure 3.2 depicts a block diagram of the binary Wyner-Ziv problem discussed in this section [WM09]. $X^n \in \mathbb{F}^n$ is a binary i.i.d. source sequence with each $X_i \sim \text{Bern}(0.5)$. We also have additional information $Z^n \in \mathbb{F}^n$ only available to the decoder but not the encoder. It is given by $Z^n = X^n \oplus W^n$, where $W^n \in \mathbb{F}^n$ with each $W_i \sim \text{Bern}(\delta)$.

The decoder reconstructs the sequence $\hat{X}^n$ based on the received index U and the side information Z^n.

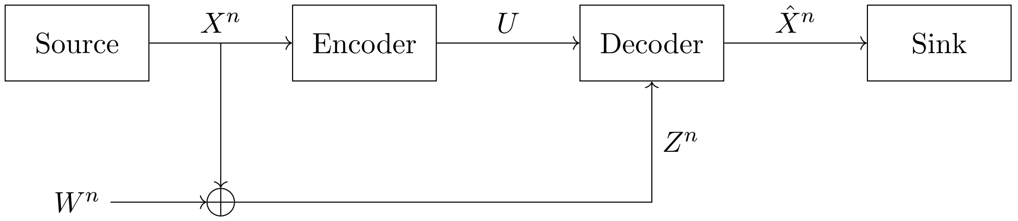


Figure 3.2: Block diagram of the binary SCSI (Wyner-Ziv) problem.

Definition 3.2.1 (Rate-Distortion Function of the Wyner-Ziv Problem [WZ76]). *The rate-distortion function of the Wyner-Ziv problem for binary symmetric source with binary symmetric channel side information is the lower convex envelope of the function shown by*

$$g_{\delta}^{WZ}(D) = \begin{cases} H_2(D * \delta) - H_2(D) & \textit{if } 0 \leq D < \delta \\ 0 & \textit{if } D = \delta \end{cases}, \tag{3.2}$$

where $D * \delta = D(1-\delta) + \delta(1-D)$.

As discussed in Example 2.1.2, when performing lossy compression of a source $X \sim \text{Bern}(0.5)$, the minimum rate required to attain a distortion D or lower is given by $R(D) = 1 - H_2(D)$. Note that when $\delta = 1/2$, the side information becomes meaningless and we will have $\text{l.c.e}\{g_{1/2}^{WZ}(D)\} = R(D) = 1 - H_2(D)$, where l.c.e denotes the lower convex envelope.

SCSI framework provides an approach that can capture complex dependencies and correlations between the source data and side information. This enables the development of coding schemes that achieve efficient compression while accommodating real-world constraints and requirements. From this perspective, SCSI finds applications in diverse fields such as multimedia communication, image and video coding, sensor networks, and beyond.

4 LDGM/LDPC Compound Code and Message Passing Algorithms

In this chapter, we discuss a code construction to the CCSI problem and our contributions to that. First, we will introduce the construction by Wainwright and Martinian [WM09], which combines an LDGM code and an LDPC code as a solution to the Gelfand-Pinsker and Wyner-Ziv problems. Then, we will examine two MPAs, the belief propagation (BP) and truthiness propagation (TP) [Reg09] algorithms, as good candidates for the LDPC and LDGM codes, respectively. Finally, we will propose an MPA for the LDGM/LDPC compound codes with binary CCSI, inspired by the BP and TP algorithms.

4.1 Wainwright and Martinian's Construction

Martin Wainwright and Emin Martinian provided a code construction [WM09] for binning and coding with side information. They showed that their code achieves the theoretical rate regions of the Gelfand-Pinsker and Wyner-Ziv problems when optimal encoding and decoding are employed. The construction involves combining a low-density generator matrix (LDGM) code and a low-density parity-check (LDPC) code, thus providing a sparse and graphical structure. As discussed in Sections 2.3 and 2.4, this structure holds a potential for utilizing the MPAs since the optimal encoding and decoding are infeasible to implement in practice.

LDGM codes are linear codes with a sparse generator matrix. The dual of an LDGM code is an LDPC code.

Definition 4.1.1 (Dual Code)**.** *Let* $\mathbf{G} \in \mathbb{F}^{k \times n}$ *be a generator matrix for the code* $\mathcal{C}$*. The dual code* $\mathcal{C}^{\perp}$*is defined by* $c^n \in \mathcal{C}^{\perp} \Leftrightarrow c^n \cdot \mathbf{G}^T = 0^k$*.*

Remark. It is important to state here that the LDGM and LDPC codes that we use to build the compound code in this section will not necessarily be the duals of each other.

LDGM codes are shown to achieve the rate-distortion bound for lossy source coding problems when the optimal encoding is performed [WMM10]. Examples 4.1.1 and 4.1.2 highlight the details on the Tanner graphs of an LDGM code and an LDPC code, respectively, for the following discussions.

Example 4.1.1 (Tanner Graph of an LDGM Code [WM09]). *Let* $\mathbf{G} \in \mathbb{F}^{n\times m}$ *be the generator matrix of an LDGM code. The Tanner graph of such a code is depicted in Figure 4.1. There,* $x^n \in \mathbb{F}^n$ *and* $z^m \in \mathbb{F}^m$ *represent the codeword and information sequence, respectively. We also have* $x^n = z^m \cdot \mathbf{G}^T$*. For a lossy source coding problem, the encoder will be given a sequence* x^n *and will be looking for a codeword* $\hat{x}^n$*, i.e.,* $\exists z^m \in \mathbb{F}^m : \hat{x}^n = z^m \cdot \mathbf{G}^T$*, with* $\mathbb{E}[d^n(X^n, \hat{X}^n)] \leq D$*. Note that for the example,* $n = 12$*,* $m = 9$*, and we have an LDGM code of rate* $R = m/n = 3/4$*.*

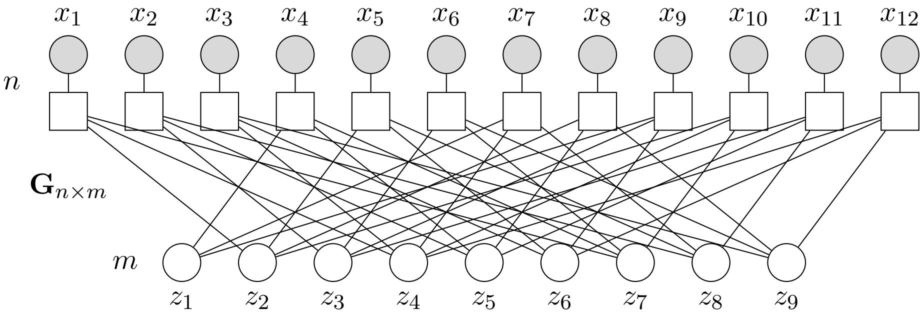


Figure 4.1: The Tanner graph of an LDGM code.

Example 4.1.2 (Tanner Graph of an LDPC Code [WM09]). *Let* $\mathbf{H} \in \mathbb{F}^{m\times n}$ *be the parity-check matrix of an LDPC code* $\mathcal{C}$*. The Tanner graph of such a code is depicted in Figure 4.2. There,* $y^n \in \mathbb{F}^n$ *represents the received sequence in a channel coding scenario. Any syndrome* $y^n \cdot \mathbf{H}^T \neq 0^m$ *will indicate an error and* $y^n \cdot \mathbf{H}^T = 0^m$ *if and only if* $y^n \in \mathcal{C}$*. Note that for the example,* $n = 12$*,* $m = n - k = 6$*, and we have an LDPC code of rate* $R = k/n = 1/2$*.*

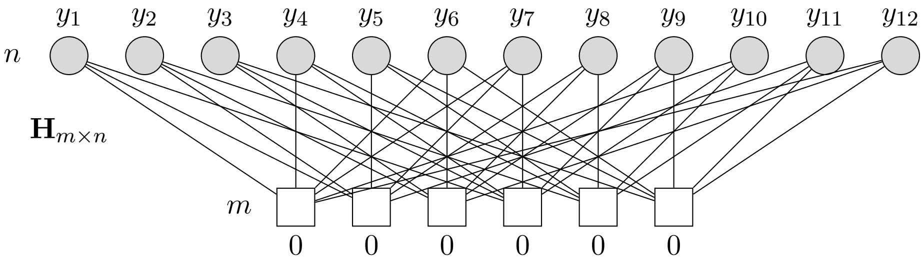


Figure 4.2: The Tanner graph of an LDPC code.

LDGM/LDPC Compound Code

Wainwright and Martinian's LDGM/LDPC compound code [WM09] is illustrated in Figure 4.3. The upper part of the Tanner graph represents the LDGM code, while the lower part represents the LDPC code. The compound code is a good source code and good channel code at the same time, thanks to the LDGM and LDPC parts, respectively.

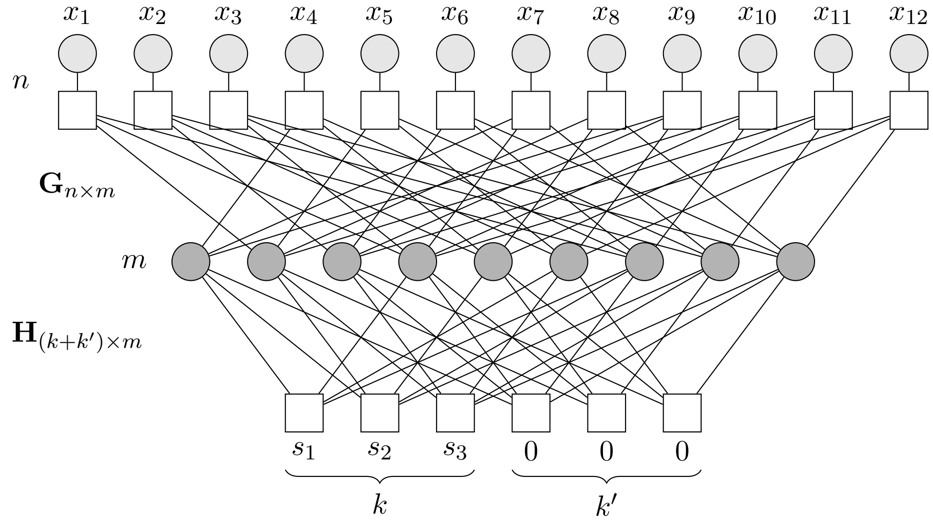


Figure 4.3: The Tanner graph of the LDGM/LDPC compound code.

The rate region of the Wyner-Ziv problem can be achieved when optimal processing is performed with the compound code [WM09]. We will not go into further detail about this and focus on the Gelfand-Pinsker problem, which is more relevant to this thesis.

Unlike what we have introduced so far, the LDPC part of the compound code has k check nodes which are not necessarily even checks, i.e., the check nodes now "check" if the variable nodes are summed not just to zero, but also one, see Figure 4.3.

Let $\mathcal{C}$ be the linear code for the compound structure when $k = 0$, i.e., k check nodes and the connected edges are removed from the graph. Then $\mathcal{C}$ has rate $R = R_{\text{LDGM}} \times R_{\text{LDPC}} = \frac{m}{n} \times \frac{m-k'}{m} = \frac{m-k'}{n}$. With $k > 0$, i.e., new check nodes and the connected edges are added back to the graph, we increase the number of parity-checks. Only a subset $\mathcal{C}(s^k) \subset \mathcal{C}$ satisfies the new checks. This $\mathcal{C}(s^k)$ is a coset of $\mathcal{C}$, where all checks $s^k \in \mathbb{F}^k$ are satisfied in addition to k' checks from $\mathcal{C}$. It can be shown that the code $\mathcal{C}$ is the composition of these cosets, i.e., $\mathcal{C} = \bigcup_{s^k \in \mathbb{F}^k} \mathcal{C}(s^k)$. Note that $\mathcal{C}(s^k)$ is a code of rate $R = \frac{m-(k+k')}{n}$ and a linear code if and only if $s^k = 0^k$.

We now explain how the compound code can be used to attain the rate region of the

Gelfand-Pinsker problem. Design $\mathcal{C}$ to be a good channel code of rate $R = \frac{m-k'}{n}$ such that $\mathcal{C}(s^k)$ are good source codes of rate $R = \frac{m-(k+k')}{n}$. If we consider the binary CCSI problem from Section 3.1, for a given message $s^k \in \mathbb{F}^k$, the side information $z^n \in \mathbb{F}^n$ will be compressed to $\hat{z}^n \in \mathcal{C}(s^k)$. The encoder will transmit the quantization error, which is the codeword of this scheme, $x^n = z^n \oplus \hat{z}^n$. Then the output at the decoder will be $y^n = x^n \oplus z^n \oplus w^n = \hat{z}^n \oplus w^n$. Since $\mathcal{C}(s^k)$ is a good source code, the constraint on the average weight of the codeword $\frac{1}{n}\sum_{i=1}^{n} x_i = \frac{1}{n}\sum_{i=1}^{n} z_i \oplus \hat{z}_i = d^n(z^n, \hat{z}^n) \leq p$ is satisfied. Since $\mathcal{C}$ is a good channel code, $\hat{z}^n$ can be obtained successfully from the received y^n. As $\hat{z}^n$ is a member of a particular coset $\mathcal{C}(s^k)$, it is possible to decode the message s^k, meaning that $R < H_2(p) - H_2(\delta)$ can be attained [KVNP14].

4.2 Belief Propagation for LDPC Codes over Binary-Input Channels

The standard BP algorithms can be used for LDPC codes, as we discussed in Section 2.4. In this section, we present the BP algorithm for LDPC codes over binary-input channels and will be utilizing it as a part of the MPA proposed for the LDGM/LDPC compound code in Section 4.4.

Algorithm 1 describes the BP algorithm in the context of channel coding. For the parity-check matrix $\mathbf{H} \in \mathbb{F}^{m\times n}$ and the received codeword $y^n \in \mathbb{F}^n$, it decodes to $\hat{c}^n \in \mathbb{F}^n$. There, $F(i)$ represents the set containing the indices j of check nodes f_j to which the variable node z_i is connected. $V(j)$ is the set containing the indices i of variable nodes z_i to which the check node f_j is connected. $L_{z_i \to f_j}$ shows the message sent from the variable node z_i to the check node f_j. We denote the sent messages by L since we are switching to the log-likelihood ratio (LLR) domain with line 1 instead of the probability domain. LLR is a measure of the reliability associated with the received information about a particular bit. It represents the logarithm of the ratio between the likelihood of the received bit being a zero and the likelihood of it being a one. X in line 1 corresponds to the transmitted code bit. We output the final LLRs Λ^n for future use.

4.3 Truthiness Propagation for LDGM Codes with Binary Sources

The sparse and graphical structure motivates LDGM codes as good candidates for encoding with MPAs, see Sections 2.3 and 2.4. Unfortunately, the standard BP algorithms

Algorithm 1: Belief propagation algorithm.

```
Data: Received sequence $y^n \in \mathbb{F}^n$, parity-check matrix $\mathbf{H} \in \mathbb{F}^{m \times n}$, number of
      iterations $W$
Result: Decoded codeword $\hat{c}^n \in \mathbb{F}^n$, final LLRs $\Lambda^n$
for $i \leftarrow 1$ to $n$ do $L_i \leftarrow \ln \frac{P_{Y|X}(y_i|0)}{P_{Y|X}(y_i|1)}$;
for $j \leftarrow 1$ to $m$ do
    foreach $i \in V(j)$ do $L_{f_j \to z_i} \leftarrow 0$;
end
for $w \leftarrow 1$ to $W$ do
    for $i \leftarrow 1$ to $n$ do
        foreach $j \in F(i)$ do $L_{z_i \to f_j} \leftarrow L_i + \sum_{\substack{k \in F(i) \\ k \neq j}} L_{f_k \to z_i}$;
    end
    for $j \leftarrow 1$ to $m$ do
        foreach $i \in V(j)$ do $L_{f_j \to z_i} \leftarrow 2 \tanh^{-1} \left( \prod_{\substack{k \in V(j) \\ k \neq i}} \tanh \left( \frac{L_{z_k \to f_j}}{2} \right) \right)$;
    end
end
for $i \leftarrow 1$ to $n$ do
    $\Lambda_i \leftarrow L_i + \sum_{k \in F(i)} L_{f_k \to z_i}$;
    if $\Lambda_i < 0$ then $\hat{c}_i \leftarrow 1$ else $\hat{c}_i \leftarrow 0$;
end
return $\hat{c}^n, \Lambda^n$;
```

are not suited to run over the LDGM code graphs as the iterations do not converge to something meaningful [Reg09]. Therefore, more advanced MPAs have to be developed. There are several MPAs provided such as TP [Reg09], survey propagation [TMZ06] and BPGD [WMM10]. In this work, we focus on Regalia's TP algorithm for two main reasons. First, it is much easier compared to the others, meaning that it provides a lower computational complexity. Second, it gives more flexibility in terms of composability for the MPA we will propose for the LDGM/LDPC compound code in Section 4.4.

In addition to the standard BP algorithm, Regalia [Reg09] proposed to add another node to the Tanner graph to create a new message update step. A representative part of the new Tanner graph is depicted in Figure 4.4. There, q_j is the new node, and $m_{z_i \to f_j}$ shows the message sent from the node z_i to the node f_j. We extend this notation to distinguish bits, e.g., $m_{z_i \to f_j}(1)$ is the probability that the message sent from the node z_i to the node f_j is one. It is always satisfied that $m_{z_i \to f_j}(0) + m_{z_i \to f_j}(1) = 1$. We have

$m_{f_j \to x_j} = m_{x_j \to q_j}$ and $m_{x_j \to f_j} = m_{q_j \to x_j} = \alpha x_j + (1-\alpha) m_{x_j \to q_j}$, where $\alpha = 1 - 2D$ with distortion D. Note that nothing has changed in the LDGM code itself, and the rest of the Tanner graph is as in Figure 4.1.

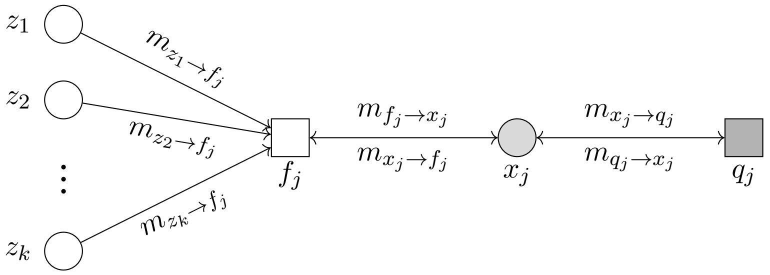


Figure 4.4: The new update node added for the TP algorithm.

Algorithm 2: Truthiness propagation algorithm. Adapted from [Reg09].

Data: Source sequence $x^n \in \mathbb{F}^n$, generator matrix $\mathbf{G} \in \mathbb{F}^{n \times m}$, algorithm constant α, dither amplitude d, number of iterations W

Result: Reconstructed source sequence $\hat{x}^n \in \mathbb{F}^n$, beliefs one $b^m(1)$, beliefs zero $b^m(0)$

```
for $i \leftarrow 1$ to $m$ do
    foreach $j \in F(i)$ do $N_{ij} \leftarrow$ sample from $\mathcal{N}(0,1)$, $m_{z_i \to f_j}(1) \leftarrow 0.5 + dN_{ij}$;
end
for $w \leftarrow 1$ to $W$ do
    for $j \leftarrow 1$ to $n$ do
        $m_{f_j \to x_j}(1) \leftarrow \frac{1}{2}\{1 - \prod_{k \in V(j)} \left(1 - 2m_{z_k \to f_j}(1)\right)\}$;
        $m_{x_j \to f_j}(1) \leftarrow \alpha x_j + (1-\alpha) m_{f_j \to x_j}(1)$;
        foreach $i \in V(j)$ do
            $m_{f_j \to z_i}(z_i) \leftarrow \frac{1}{2}\{1 + (-1)^{z_i} \left(1 - 2m_{x_j \to f_j}(1)\right) \times \prod_{k \in V(j), k \neq i} \left(1 - 2m_{z_k \to f_j}(1)\right)\}$;
        end
        for $i \leftarrow 1$ to $m$ do
            foreach $j \in F(i)$ do $m_{z_i \to f_j}(z_i) \leftarrow \zeta_{ij} \prod_{k \in F(i), k \neq j} m_{f_k \to z_i}(z_i)$ ;
        end
end
for $i \leftarrow 1$ to $m$ do
    $b_i(z_i) \leftarrow \zeta_i \prod_{k \in F(i)} m_{f_k \to z_i}(z_i)$;
    if $b_i(1) > b_i(0)$ then $z_i \leftarrow 1$ else $z_i \leftarrow 0$;
end
return $\hat{x}^n \leftarrow z^m \cdot \mathbf{G}^T, b^m(1), b^m(0)$;
```

Algorithm 2 presents the TP algorithm for a lossy source coding problem. It quantizes a source sequence $x^n \in \mathbb{F}^n$ to the reconstructed sequence $\hat{x}^n \in \mathbb{F}^n$ for a given LDGM code with the generator matrix $\mathbf{G} \in \mathbb{F}^{n\times m}$. There, we use f^n, z^m, $F(\cdot)$, and $V(\cdot)$ in the same sense as in Algorithm 1. The factor ζ ensures that the sum of $m_{(\cdot)}(1)$ and $m_{(\cdot)}(0)$ is one at each step, e.g., $m_{z_i\to f_j}(0)+m_{z_i\to f_j}(1) = 1$. We output the beliefs $b^m(\cdot)$ for future use.

Figure 4.5 shows the rate-distortion performance of Algorithm 2 as the empirical average of 4,000 runs, with 200 different frames for each 20 different generator matrices. The parameters are $n = 300$, $m = \{60, 100, 120, 150, 200\}$, each $x_j \sim \text{Bern}(0.5)$, $\alpha = 1-2D$ with $R(D) = 1 - H_2(D) = R_{\text{LDGM}} = m/n$, $d = 0.001$, and $W = 50$. The average distortion is $d^n(x^n, \hat{x}^n) = \frac{1}{n}\sum_{j=1}^{n} d(x_j, \hat{x}_j)$. The generator matrices are randomly generated with 2 ones in each row and $2n/m$ ones in each column, as in [Reg09].

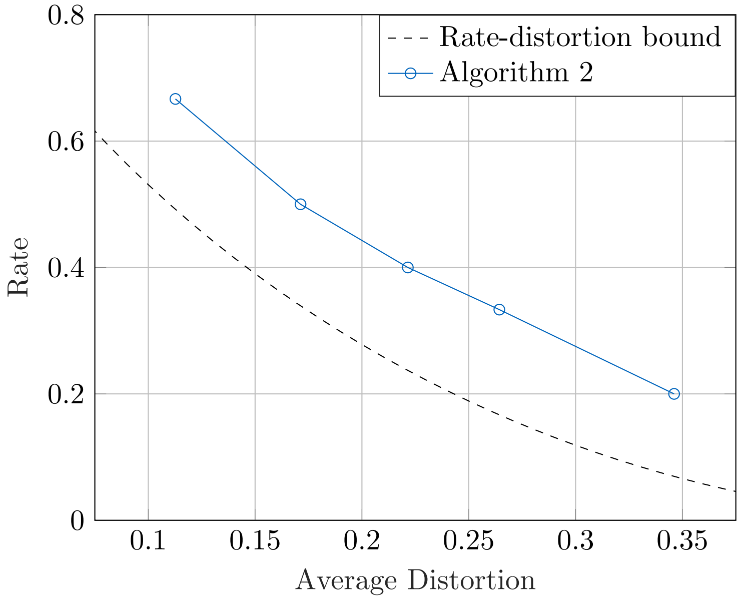


Figure 4.5: The rate-distortion performance of Algorithm 2 with $n = 300$.

Despite the gap to the rate-distortion bound in Figure 4.5, Algorithm 2 offers reasonable performance given the low implementation complexity. It has been shown that the gap to the bound can be reduced by generating irregular generator matrices [Reg09], which we will not discuss in this work.

4.4 Proposed Algorithm for LDGM/LDPC Compound Codes with Binary CCSI

It is of interest to develop appropriate MPAs for the LDGM/LDPC compound code. The challenge of this structure is to find an MPA that satisfies both LDGM and LDPC codes' check nodes. There are very few studies addressing this issue. [WH09, KT08] approach the problem from a DPC perspective. [WH09] focuses on optimizing the degree distributions of the codes. Although [KT08] does not directly use the LDGM/LDPC compound code, they make use of an LDPC code for quantization purposes. The BPGD algorithm [WMM10] was shown to be very successful in LDGM codes, thus making it a potential candidate for the compound structure. However, [KVNP14] showed that it was always incapable of satisfying the check nodes. [KVNP14] suggested to use SC as the only solution that makes the BPGD algorithm work. But SC greatly increases the computational complexity. In this work, we focus on schemes without SC and propose new MPAs for both the encoding and decoding, inspired by the BP and TP algorithms. The basis of our choice and proposal is the composability of these two algorithms and their relatively lower complexity.

Encoding Algorithm

Algorithm 3 summarizes the encoding algorithm. The input notation is as in Figure 4.3. Two functions are defined by calling Algorithms 1 and 2. The parity-check matrix $\mathbf{H}$ and the BP algorithm have been extended, see lines 1 and 22-23, respectively. Note that unlike the all-even check nodes assumption in Algorithm 1, now the odd check nodes are also possible. $\mathbf{I}_k$ is the $k \times k$ identity matrix and $\mathbf{0}$ is an all-zero matrix. $\Lambda^{m+k}[1:m]$ denotes the the first m elements of Λ^{m+k} and Inf means the infinity.

Decoding Algorithm

In the context of classical channel coding, a received sequence y^n is decoded to the codeword $\hat{x}^n$ by running MPAs such as Algorithm 1, e.g., if a systematic generator matrix is used, the message s^k can be easily decoded such that $\hat{s}^k = \hat{x}^n[1:k]$. But for the compound code, the message s^k should be recovered from the decoded codeword $\hat{x}^n$, meaning that it would not be sufficient to just have some $\hat{x}^n$ by running a classical MPA. Therefore, a decoding algorithm concerning the compound structure also needs to be developed.

Figure 4.6 depicts the Tanner graph of the proposed decoding structure. The graph

Algorithm 3: Proposed encoding algorithm for the compound code.

```
Data: Message $s^k \in \mathbb{F}^k$, side information $z^n \in \mathbb{F}^n$, generator matrix $\mathbf{G} \in \mathbb{F}^{n\times m}$,
      parity-check matrix $\mathbf{H} \in \mathbb{F}^{(k+k')\times m}$, algorithm constant $\alpha$, dither amplitude
      $d$, number of iterations $W$, number of TP iterations $W_{\mathrm{TP}}$, number of BP
      iterations $W_{\mathrm{BP}}$
Result: Codeword $x^n \in \mathbb{F}^n$
1  $\mathbf{H}' \leftarrow \begin{pmatrix} \mathbf{H} & \begin{matrix} \mathbf{I}_k \\ \mathbf{0} \end{matrix} \end{pmatrix}$;
2  $\beta^m(1) \leftarrow 0^m$;
3  for $i \leftarrow 1$ to $W$ do
4      $b^m(1), b^m(0) \leftarrow$ EncoderTruthinessPropagation($z^n, \mathbf{G}, \alpha, d, W_{TP}, i, \beta^m(1)$);
5      $v^{m+k}$, $\Lambda^{m+k} \leftarrow$ EncoderBeliefPropagation($\mathbf{H}', W_{BP}, b^m(1), b^m(0), s^k$);
6      $\xi^m \leftarrow \Lambda^{m+k}[1:m]$, $\beta^m(1) \leftarrow 1/(1+e^{\xi^m})$;
7  end
8  $u^m \leftarrow v^{m+k}[1:m]$, $\hat{z}^n \leftarrow u^m \cdot \mathbf{G}^T$;
9  return $x^n \leftarrow z^n \oplus \hat{z}^n$;
10 Function EncoderTruthinessPropagation($x^n, \mathbf{G}, \alpha, d, W, \psi, \beta^m(1)$):
       // Replace lines 1-3 in Algorithm 2 with the following lines 11-17
11     for $i \leftarrow 1$ to $m$ do
12         if $\psi = 1$ then
13             foreach $j \in F(i)$ do $N_{ij} \leftarrow$ sample from $\mathcal{N}(0,1)$, $m_{z_i \to f_j}(1) \leftarrow 0.5 + dN_{ij}$;
14         else
15             foreach $j \in F(i)$ do $m_{z_i \to f_j}(1) \leftarrow \beta_i(1)$;
16         end
17     end
18     Call Algorithm 2 with the given input parameters;
19     return $b^m(1), b^m(0)$;
20 end
21 Function EncoderBeliefPropagation($\mathbf{H}, W, b^\mu(1), b^\mu(0), s^\kappa$):
       // Replace line 1 in Algorithm 1 with the following lines 22-23
22     for $i \leftarrow 1$ to $\mu$ do $L_i \leftarrow \ln \frac{b_i(0)}{b_i(1)}$;
23     for $i \leftarrow 1$ to $\kappa$ do $L_{\mu+i} \leftarrow (1-2s_i) \times \mathrm{Inf}$;
24     Call new Algorithm 1 with the given input parameters;
25     return $\hat{c}^n, \Lambda^n$;
26 end
```

contains the same nodes as the graph in Figure 4.3. In order to make the proposed algorithm more understandable, the variable nodes are grouped in a row, and the check nodes are grouped in a row. y^n and $\hat{u}^m$ represent the received and intermediate node sequences, respectively. Note that $\hat{s}^k$ shows the nodes corresponding to the message. Therefore, these check nodes cannot be used for decoding at the beginning.

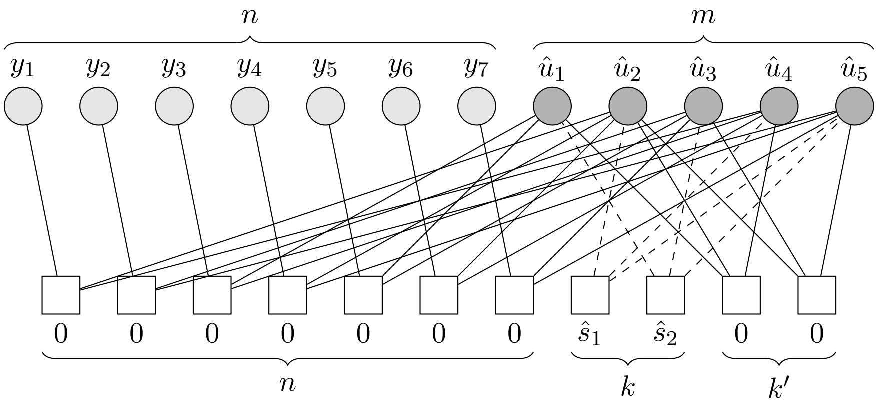


Figure 4.6: The Tanner graph of proposed decoding structure.

The proposed decoding approach is summarized by Algorithm 4. It is a modified version of Algorithm 1. Since there is no initial information for the intermediate nodes $\hat{u}^m$, a small dither noise is applied to those nodes inspired by the TP algorithm. $\mathbf{H_1}$ is excluded from the graph as $\hat{s}^k$ is unknown at first. Note again that the parity-check matrix $\mathbf{H}$ and the BP algorithm have been extended, see lines 2 and 8-9, respectively. After W iterations, a resulting sequence $\hat{u}^m$ is obtained. Finally, the decoded message $\hat{s}^k$ is found by using $\mathbf{H_1}$. In the notation, $\mathbf{H}[k+1 : \text{end}, :]$ means the sub-matrix from matrix $\mathbf{H}$ starting from the row with index $k+1$ to the last row, including all columns.

Algorithm 4: Proposed decoding algorithm for the compound code.

```
Data: Received sequence $y^n \in \mathbb{F}^n$, generator matrix $\mathbf{G} \in \mathbb{F}^{n \times m}$, parity-check matrix
      $\mathbf{H} \in \mathbb{F}^{(k+k') \times m}$, dither amplitude $d$, number of iterations $W$
Result: Decoded message $\hat{s}^k \in \mathbb{F}^k$
1  $\mathbf{H_1} \leftarrow \mathbf{H}[1:k,:], \mathbf{H_2} \leftarrow \mathbf{H}[k+1:\text{end},:]$;
2  $\mathbf{H}' \leftarrow \begin{pmatrix} \mathbf{G} & \mathbf{I}_n \\ \mathbf{H_2} & \mathbf{0} \end{pmatrix}$;
3  for $i \leftarrow 1$ to $m$ do $N_i \leftarrow$ sample from $\mathcal{N}(0,1)$, $b_{\hat{u}_i} \leftarrow 0.5 + dN_i$;
4  $v^{m+n} \leftarrow$ DecoderBeliefPropagation($y^n, \mathbf{H}', W, b_{\hat{u}}^m$);
5  $\hat{u}^m \leftarrow v^{m+n}[1:m]$;
6  return $\hat{s}^k \leftarrow \hat{u}^m \cdot \mathbf{H_1}^T$;
7  Function DecoderBeliefPropagation($y^\nu, \mathbf{H}, W, b_{\hat{u}}^\mu$):
       // Replace line 1 in Algorithm 1 with the following lines 8-9
8      for $i \leftarrow 1$ to $\mu$ do $L_i \leftarrow \ln \frac{1-b_{\hat{u}_i}}{b_{\hat{u}_i}}$;
9      for $i \leftarrow 1$ to $\nu$ do $L_{\mu+i} \leftarrow \ln \frac{P_{Y|X}(y_i|0)}{P_{Y|X}(y_i|1)}$;
10     Call new Algorithm 1 with the given input parameters;
11     return $\hat{\mathbf{c}}$;
12 end
```

5 Results and Discussion

In this chapter, we present and discuss the performance of the proposed encoding and decoding MPAs for the LDGM/LDPC compound code in the previous chapter. First, we will describe the channel model. Then, we will show the performance of Algorithm 3 at the transmitter and Algorithm 4 at the receiver after the AWGN channel. Next, based on the shortcomings of the initial results, we will propose a solution by concatenating an outer code with the compound code. Finally, we will present the performance of the final schemes with our comments.

5.1 Channel Model

We present the channel model we will consider in the following sections. Figure 5.1 depicts the end-to-end block diagram of the binary CCSI over AWGN channel. There, each $\Theta_i \sim \mathcal{N}(0, \sigma^2)$ and the other notation is the same as in the previous chapter. For the rest of the thesis, we always use OOK as the modulation type of the transmission, i.e., we send unit energy for each $X_i = 1$ and zero energy for each $X_i = 0$. Therefore, X^n and modulated symbols are equal to each other, and we do not use separate letter for the modulated symbols. It should be noted here that the average codeword weight $\frac{1}{n}\sum_{i=1}^{n} X_i$ is equal to the average symbol energy E_s in the case of OOK. This relates to our motivation with probabilistic shaping. Section 2.5 can be referred to for the discussion. We have $Y^n = X^n \oplus Z^n + \Theta^n = \hat{Z}^n + \Theta^n$, meaning that $Y^n \notin \mathbb{F}^n$ anymore.

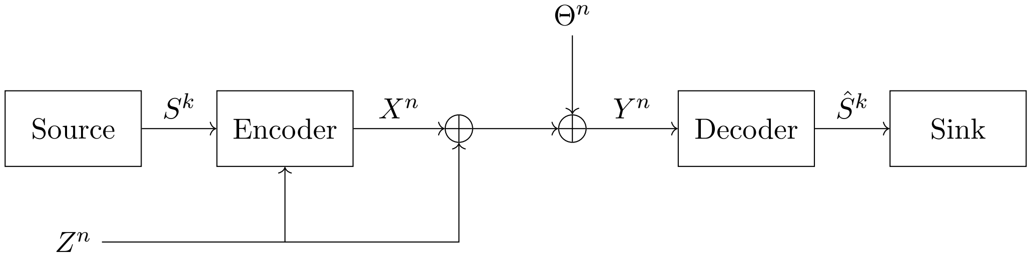
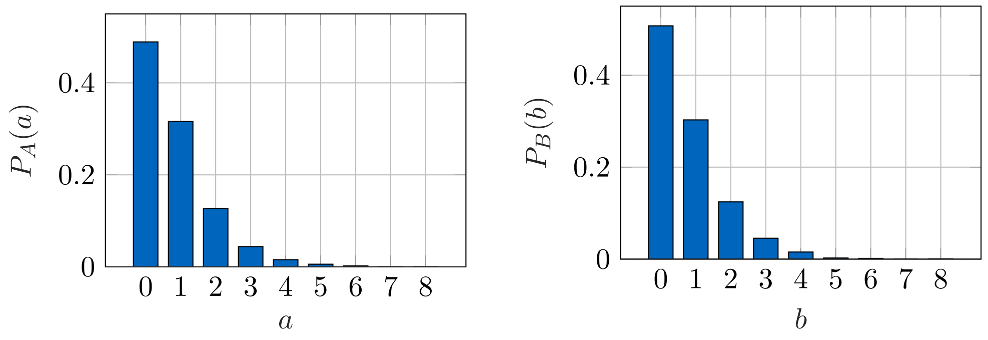


Figure 5.1: Block diagram of the binary CCSI over AWGN channel.

5.2 Performance of Encoding

As highlighted in Section 4.4, the challenge with the MPAs for the compound code is finding an encoding algorithm that satisfies all check nodes of the code. Therefore, we start by examining whether our encoding algorithm can produce a valid codeword, i.e., if $u^m \cdot \mathbf{H_1}^T = s^k$ and $u^m \cdot \mathbf{H_2}^T = 0^{k'}$ are satisfied. u^m represents the intermediate variable nodes at the transmitter side. Here, we use $\mathbf{H_1}$ and $\mathbf{H_2}$ in the same sense as in Algorithm 4, such that $\mathbf{H_1} = \mathbf{H}[1:k,:]$ and $\mathbf{H_2} = \mathbf{H}[k+1:\text{end},:]$.

Let A and B be two discrete random variables that show the number of unsatisfied check nodes of $\mathbf{H_1}$ and $\mathbf{H_2}$, respectively, after Algorithm 3 is performed for the following parameters. The empirical probability mass functions $P_A(a)$ and $P_B(b)$ were obtained as in Figure 5.2 after 4,000 runs, with 200 different frames for each 20 different generator and parity-check matrices. The parameters are $n = 300$, $m = 200$, $k = k' = 50$. In the rest of the thesis, we will use that each $\{s_i, z_i\} \sim \text{Bern}(0.5)$, $\alpha = 1 - 2D$ with $m/n = 1 - H_2(D)$, $d = 0.001$, $W = 1$, and $W_{\text{TP}} = W_{\text{BP}} = 50$. Here, W was chosen as such because the performance starts to deteriorate when $W > 1$. The generator matrices are randomly generated with 2 ones in each row and $2n/m$ ones in each column [Reg09]. We randomly generated the parity-check matrices such that $\mathbf{H} = (\mathbf{I}_{k+k'} \mid \mathbf{P})$, where $\mathbf{P}$ has $\frac{2}{(m-(k+k'))/m} - 1$ ones in each row and 2 ones in each column. Figure 5.3 depicts an example from the sparsity pattern of the generated matrices $\mathbf{G}$ and $\mathbf{H}$, where each dot in the plot represents the nonzero elements in the matrix.

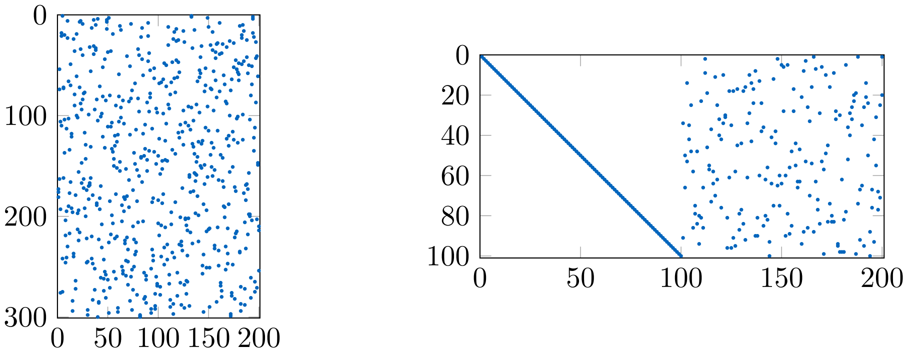


(a) The distribution of unsatisfied check nodes number of $\mathbf{H_1}$ with $k = 50$.

(b) The distribution of unsatisfied check nodes number of $\mathbf{H_2}$ with $k' = 50$.

Figure 5.2: The performance of the encoding algorithm to generate valid codewords for $n = 300$, $m = 200$, and $k + k' = 100$.

As can be seen in Figure 5.2, the distributions of A and B are almost the same. Therefore, we can conclude that Algorithm 3 performs equally for all check nodes of the LDPC code. Since we will aim to decode the message s^k by using $\mathbf{H_1}$ at the receiver, we continue with commenting on Figure 5.2a. $P_A(0) = 0.489$ shows that our encoding algorithm can correctly encode almost half of the frames. On the other hand, we have $P_A(a > 0) = 0.511$, so at least 1 of 50 check nodes is not satisfied for the other half of the frames. We also find $\mathbb{E}[A] = \sum_a P_A(a) \cdot a = 0.809$, and it will help us in interpreting the results that follow.



(a) An example of the generator matrix $\mathbf{G}$ with 600 nonzero elements.

(b) An example of the parity-check matrix $\mathbf{H}$ with 300 nonzero elements.

Figure 5.3: The sparsity pattern of the generated matrices with $n = 300$, $m = 200$, and $k + k' = 100$.

While discussing the encoding performance for the LDPC part, of course, the performance of the LDGM part is also important since the quantization task is another part of the Gelfand-Pinsker problem. As discussed in the previous chapters, we can comment on the quantization performance by considering the gap to the rate-distortion bound. Figure 5.4 illustrates the rate-distortion performance of Algorithm 3 compared to the theoretical bound and Algorithm 2. It is the empirical average of 4,000 different runs with $n = 300$, $m = 200$, and $k = k' = \{50, 60, 70\}$. Note that the average distortion $d^n(z^n, \hat{z}^n)$ is also the average weight of the codeword such that $\frac{1}{n}\sum_{i=1}^{n} z_i \oplus \hat{z}_i = \frac{1}{n}\sum_{i=1}^{n} x_i$ and the average symbol energy in the case of OOK, e.g., $E_s \approx 0.34$ for the compound code of rate $R = 1/3$ with our encoding algorithm. The rate is $R = R_{\text{LDGM}} \times R_{\text{LDPC}} = \frac{m-(k+k')}{n}$ for Algorithm 3, while $R = R_{\text{LDGM}} = \frac{m=\{60,100,120\}}{n}$ for Algorithm 2. Since it is not easy to randomly generate the matrices $\mathbf{G}$ and $\mathbf{H}$ that satisfy the aforementioned conditions, we had to limit the number of different rates to three samples in the simulation. It can be seen in the figure that our encoding algorithm yields an average distortion that is approximately 0.09 more than only the TP algorithm introduces. We can interpret this as the cost of the BP iterations trying to generate valid codewords for the LDPC code.

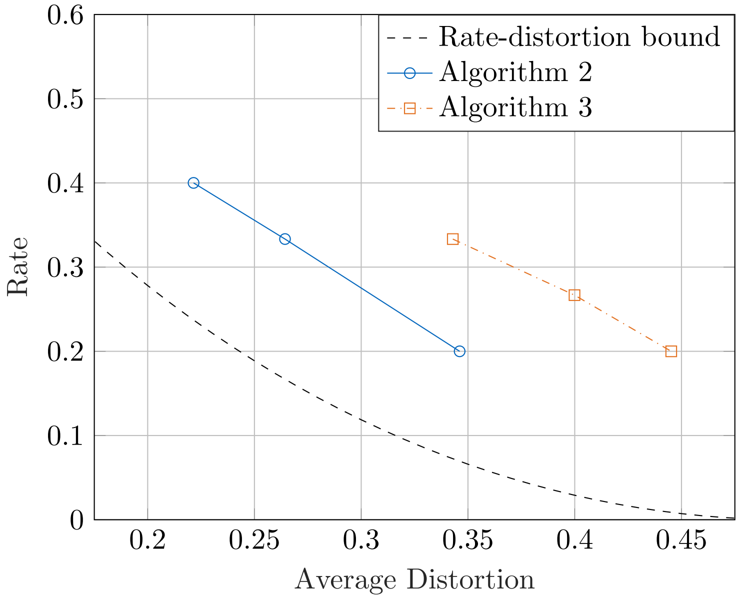


Figure 5.4: The rate-distortion performance of Algorithm 3 with $n = 300$, $m = 200$, and $k = k' = \{50, 60, 70\}$.

5.3 Performance of Decoding

We have presented the initial results of the encoding algorithm at the transmitter side. We now investigate how our decoding algorithm behaves at the receiver side, given the performance of the encoder and the AWGN channel.

We need to find the LLRs for the OOK modulated symbols to be able to run the decoding algorithm given by Algorithm 4, see line 9. For the AWGN channel, we find

$$L_i = \ln \frac{P_{Y|X}(y_i|0)}{P_{Y|X}(y_i|1)} = \ln \frac{P_{Y|X}(y_i|x_i = 0)P_X(0)}{P_{Y|X}(y_i|x_i = 1)P_X(1)} \tag{5.1}$$

$$= \ln \frac{e^{-y_i^2/2\sigma^2} P_X(0)}{e^{-(y_i-1)^2/2\sigma^2} P_X(1)} \tag{5.2}$$

$$= \frac{1 - 2y_i}{2\sigma^2} + \ln \frac{P_X(0)}{P_X(1)}, \tag{5.3}$$

where we are calculating the conditional probabilities for all x_i by (5.1). Also note that the average symbol energy $E_s = \mathbb{E}[|X|^2] = P_X(0) \cdot 0^2 + P_X(1) \cdot 1^2 = P_X(1)$. Finally, we distinguish the rate of compound code and overall rate by $R = \frac{m-(k+k')}{n}$ and $R' = k/n$, respectively. We have provided all the explanations needed to present the performance of Algorithm 4.

Figure 5.5 demonstrates the bit error rate (BER) and frame error rate (FER) performance of our initial algorithms for the compound code of rate $R = 1/3$ and $R' = 1/6$ with $n = 300$, $m = 200$, and $k = k' = 50$. We have $E_s \approx 0.34$ from Figure 5.4 and define $\mathrm{SNR} = 0.34/\sigma^2$. BER is the empirical mean of the average message distortion after running φ frames, i.e., $\frac{1}{\varphi}\sum_{i=1}^{\varphi} d^k(s^k, \hat{s}^k)$. We count a frame error if $d^k(s^k, \hat{s}^k) > 0$. The simulation was run until at least 50 frame errors were obtained for each SNR and providing $\varphi \geq 1,000$.

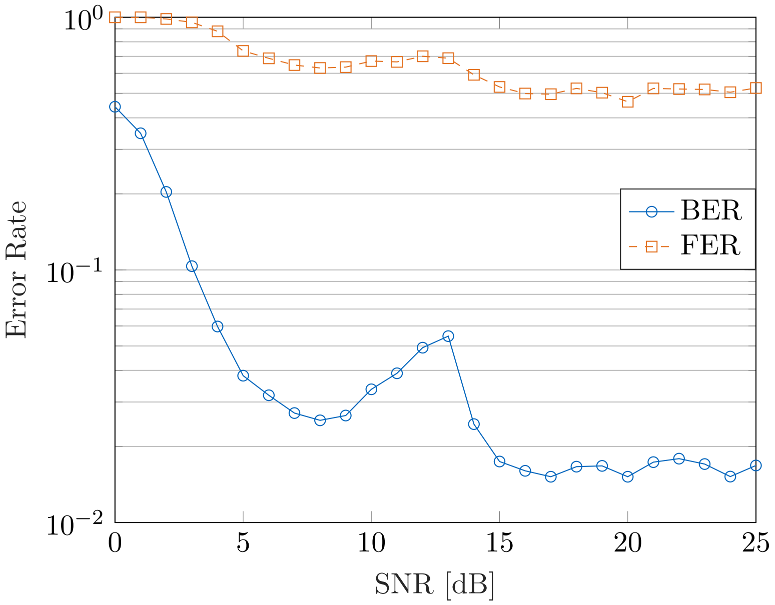


Figure 5.5: The error rate performance of the compound code with $n = 300$, $m = 200$, $k = k' = 50$, $R = 1/3$, and $R' = 1/6$.

Based on the results in Figure 5.5, two observations are in order. First, there is an unexpected jump between 9 dB and 14 dB. This is caused by the formal LLR expression given by (5.3). The dither noise we use in our encoding and decoding algorithms and the effects that may arise from the compound code itself make (5.3) inefficient for that SNR region. This problem can be avoided by informing the receiver in advance and modifying the (5.3) for the problematic SNR region to provide a better result. We ignore this problem for now, but we will tackle it in the sections that follow. Second, we have two BER and FER floors from 15 dB. This is expected and consistent with what we have discussed with Figure 5.2a. There, A was the random variable that shows the number of unsatisfied check nodes of $\mathbf{H_1}$ when $k = 50$, i.e., the number of wrongly encoded bits. We have found $\mathbb{E}[A] = 0.809$, meaning that every 0.809 bits of a message frame of length 50 will be encoded erroneously. It can be seen that the BER floor is at $\mathbb{E}[A]/50 \approx 0.016$.

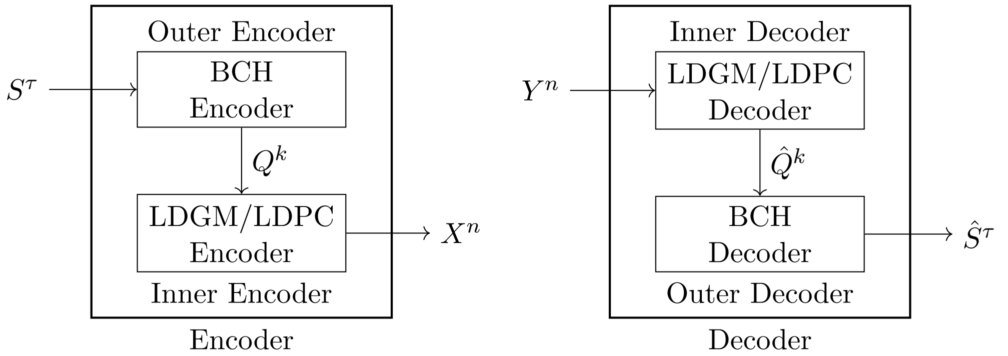
With the same sense, the FER floor is at $P_A(a > 0) \approx 0.5$. In the next section, we come up with a solution to these shortcomings.

5.4 Proposed Algorithm with Concatenated Outer Code

We make use of code concatenation as the solution to the erroneously encoded frames discussed in the previous section. Concatenated codes are constructed by combining multiple layers of error correction codes, each layer providing unique error correction capabilities. Among the various options available for constructing concatenated codes, the Bose-Chaudhuri-Hocquenghem (BCH) codes [BRC60, Hoc59] stand out as a widely adopted choice.

BCH codes are a class of cyclic error correction codes. How the BCH codes are generated is beyond the scope of this work, but [CJC13] can be referred to for the definitions. In the concatenated code configuration, BCH codes are utilized as the outer code, which serves as the first layer of error correction. Thus, we can preserve the original message from the errors caused by our encoding algorithm. On the other hand, note that each concatenated code will reduce the overall rate R'. Therefore, trying to keep the rate R' as high as possible while keeping the error correction capability of the BCH code reasonably high is the main trade-off of this approach and will be discussed in this section.

Figure 5.6 depicts the new blocks for the encoder and decoder with the concatenated outer BCH code. The rest of the diagram is as in Figure 5.1. This structure has an overall rate of $R' = \tau/n$.



(a) The new encoder encodes the source message S^τ to the codeword X^n.

(b) The new decoder decodes the received sequence Y^n to the message $\hat{S}^\tau$.

Figure 5.6: The encoder and decoder blocks with the concatenated outer BCH code.

By their definitions, BCH codes cannot be utilized for information sequences or codewords

Table 5.1: The available lengths for BCH codes and error correction capabilities.

k	...	15	15	31	31	31	31	31	63	63	63	63	63	63
τ	...	7	5	26	21	16	11	6	57	51	45	39	36	30
ϱ	...	2	3	1	2	3	5	7	1	2	3	4	5	6

k	63	63	63	63	63	127	127	127	127	127	127	127	127	...
τ	24	18	16	10	7	120	113	106	99	92	85	78	71	...
ϱ	7	10	11	13	15	1	2	3	4	5	6	7	9	...

of any length, see [CJC13, Chapter 2]. We also use narrow-sense BCH codes as they provide efficient decoding algorithms. Thus, we restrict ourselves to choosing τ and k from a finite set. Table 5.1 provides a part of that set. There, ϱ denotes the error correction capability of the corresponding code. We denote a BCH code with BCH(k, τ), e.g., BCH$(63, 24)$ encodes a message of length 24 to the codeword of length 63 and it ensures that errors up to 7 of 63 codeword bits can be corrected by the decoder.

We now discuss which BCH code to choose to concatenate with our encoder. Figure 5.7 is the same as Figure 5.2a and plotted again for the discussion. We consider the following criteria.

- The error correction capability of the code should be as high as possible, e.g., if all encoding errors in Figure 5.7 are to be corrected, the BCH code should have an error correction capability rate such that $\varrho/k \geq 8/50$. This would result in the set of appropriate codes being $\{\text{BCH}(15,5), \text{BCH}(31,11), \text{BCH}(31,6), \ldots\}$.

- For a fixed n, the overall rate $R' = \tau/n$ should be as high as possible since we want to send as many bits as possible at once. BCH codes with a larger τ are therefore preferred.

- We must take into account the constraints of the compound code, e.g., k is also a parameter for the parity-check matrix $\mathbf{H}$, and a parity-check matrix that satisfies the aforementioned conditions may not be easily generated. Another example, for a fixed m, choosing a $k \geq m$ would not make sense.

Based on the trade-off considerations above, we propose to concatenate one BCH encoder with multiple LDGM/LDPC encoders. Figure 5.8 demonstrates the resulting block for the encoder. There, the message S^τ is encoded to Q^κ by the BCH encoder and the BCH codeword is partitioned to the subframes such that

$$Q^\kappa = \left({}^{(1)}Q^{\kappa/\lambda} \quad {}^{(2)}Q^{\kappa/\lambda} \quad \ldots \quad {}^{(\lambda)}Q^{\kappa/\lambda} \right),$$

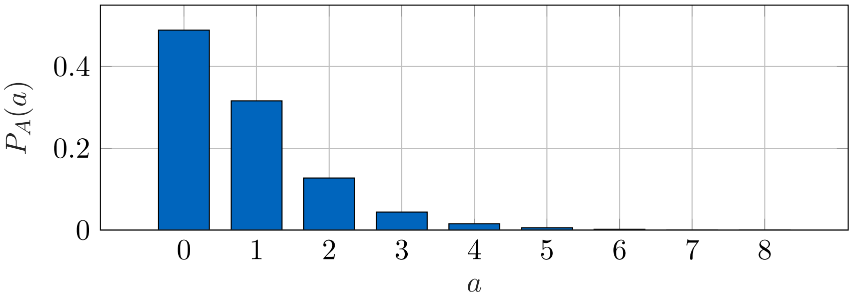


Figure 5.7: The distribution of encoding bit errors for frame length 50.

where λ is the number of subframes. Note that we denote the BCH codeword length by κ instead of k to distinguish them for this case. If $\lambda = 1$, we do not make use of the multiple encoders, so $\kappa = k$. Finally, the LDGM/LDPC encoders generate the codewords X^n as usual. We can use the same approach at the decoder, i.e., every λ frames will be decoded to ${}^{(\cdot)}\hat{Q}^{\kappa/\lambda}$ and $\hat{S}^\tau$, respectively. This scheme has an overall rate of $R' = \tau/\lambda n$.

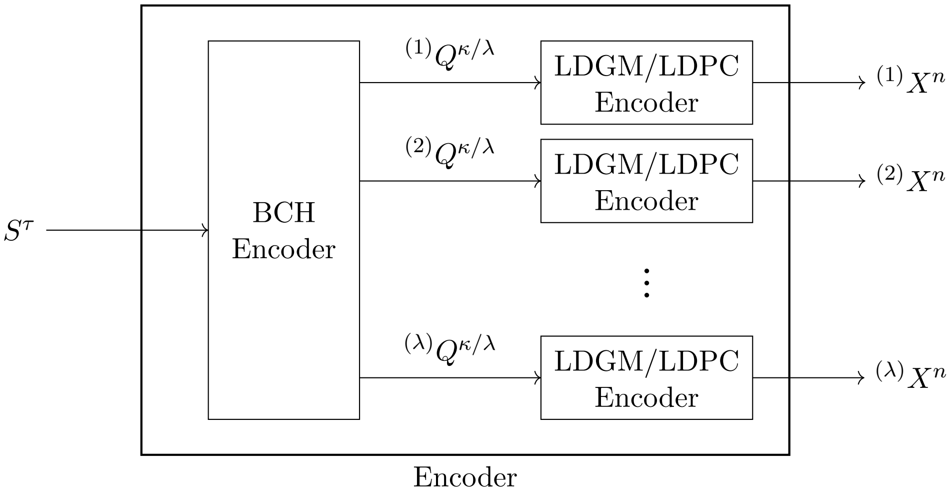
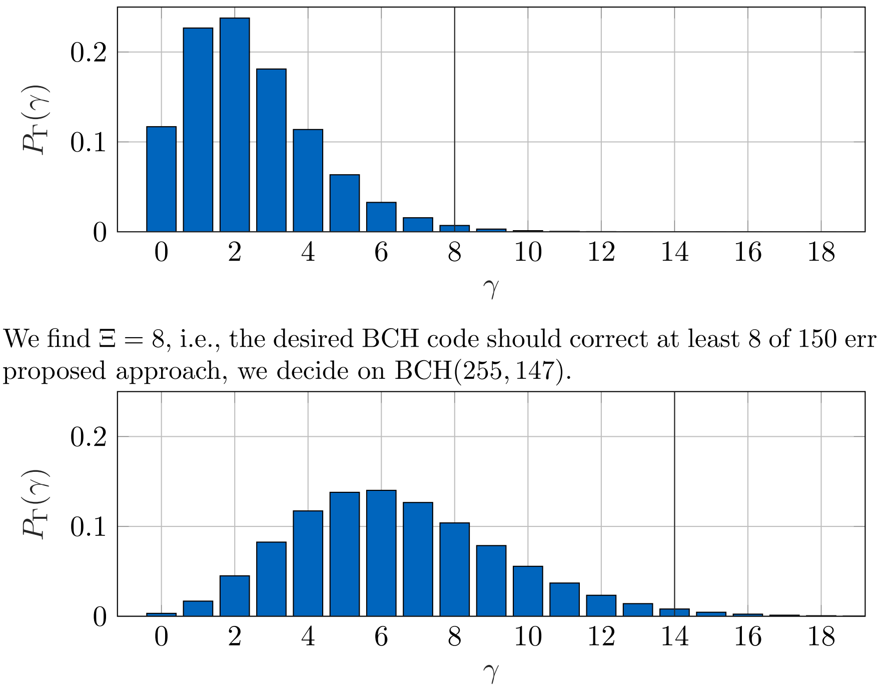


Figure 5.8: The encoder block for the concatenation with multiple encoders.

In order to determine which error correction capability to choose for the BCH encoder, we should first find the probability distribution of encoding bit errors for multiple encoders. We know from the probability theory that the sum of two or more independent random variables follows a probability distribution that can be obtained by convolving their respective individual distributions. Let Γ be the random variable that shows the number of encoding bit errors for the multiple encoder frames of length 50 each. Then we have $\Gamma \sim \sum_{i=1}^{\lambda} A$, where A is the random variable for one frame as in Figure 5.7. Two distributions of Γ for $\lambda = \{3, 8\}$ are shown in Figure 5.9. There, we set a limit for the error correction capability rate of the BCH code, as specifying a ϱ/κ that will correct all errors will greatly reduce the overall rate R'. It does not already make sense to consider all probabilities, e.g. we have only $P_\Gamma(18) = 1.39 \times 10^{-7}$ for $\lambda = 3$. We cover at least 99% of the error probabilities by setting a Ξ, where $\sum_{\gamma=0}^{\Xi} P_\Gamma(\gamma) \geq 0.99 > \sum_{\gamma=0}^{\Xi-1} P_\Gamma(\gamma)$.



(a) $\lambda = 3$. We find $\Xi = 8$, i.e., the desired BCH code should correct at least 8 of 150 errors. Based on the proposed approach, we decide on BCH$(255, 147)$.

(b) $\lambda = 8$. We find $\Xi = 14$, i.e., the desired BCH code should correct at least 14 of 400 errors. Based on the proposed approach, we decide on BCH$(511, 358)$.

Figure 5.9: The distribution of encoding bit errors for multiple encoders with frame length 50 each.

Based on the found Ξ, we refer to Table 5.1 and get a finite subset that provides $\varrho/k \geq \Xi/\lambda 50$. Note that this k will be κ of the BCH encoder. We also want to keep the rate $R' = \tau/\lambda n$ as high as possible, thus prioritizing the appropriate BCH codes with a larger τ. Finally, for the fixed $n = 300$, $m = 200$, and $k + k' = 100$, we decide on the first BCH code which has $\kappa/\lambda < k + k' = 100$, see Figures 5.9a and 5.9b for the examples.

We have discussed which BCH code to choose for which λ. But we also need to bring an approach for how to set λ. Since our ultimate goal is to maximize $R' = \tau/\lambda n$, we examine the rates that this approach provides for each λ and the corresponding BCH code. Figure 5.10 demonstrates an overview of this relation. It can be seen that, for example, setting $\lambda = 11$ makes much more sense than setting $\lambda = 20$ since it provides more rate. This is the result of choosing the parameters of the BCH code from a finite set, as we discussed above. The orange bars in the figure indicate that the decided BCH code provides a κ which can be divided by λ without a remainder. Note that otherwise, we would have to decimate the remaining bits. Therefore, orange λ's with the highest rates are preferred.

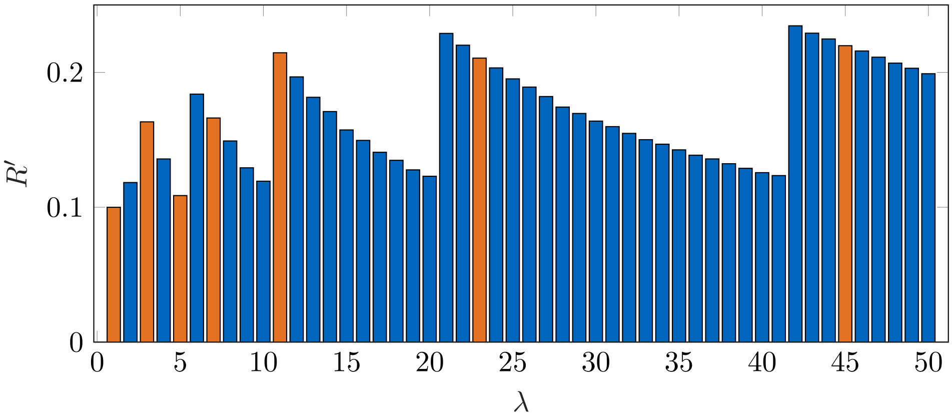
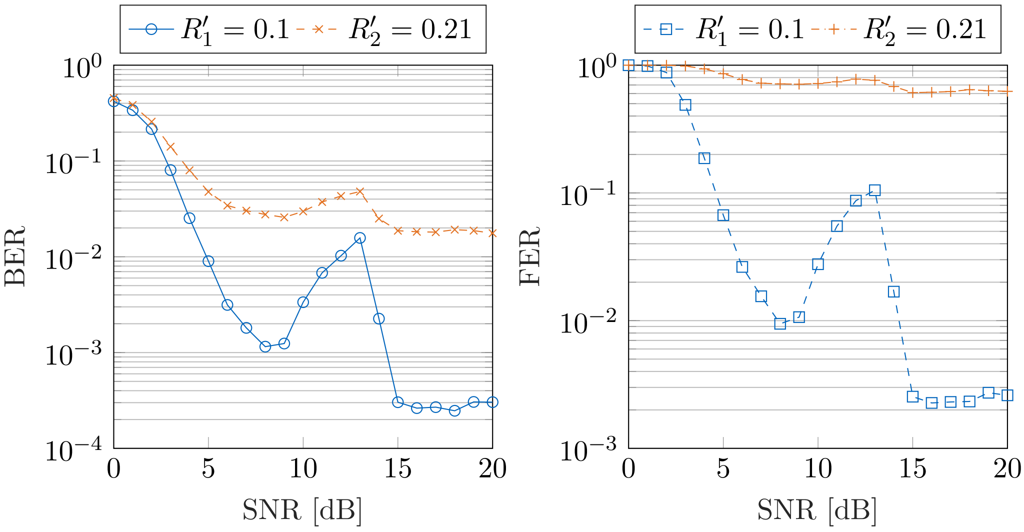


Figure 5.10: The resulting rates for set λ's. The orange bars indicate $\kappa \equiv 0 \pmod{\lambda}$.

5.5 Final Results

In this section, we present the results of what we have discussed so far in this chapter and compare them with each other. As a solution to the error floors encountered in the initial results, we made use of code concatenation in the previous section. Figure 5.11 reveals how the code concatenation can decrease the error floors. There, we have considered two different codes. The first one is the BCH(63,30) concatenated code with $\lambda = 1$ and the rate $R'_1 = 0.1$. The second one is the compound code with the rate $R'_2 = 0.21$. For a fair comparison, both have $k = 63$ and $k' = 37$. For the concatenated code, note that BER is the empirical mean of the average message distortion after running φ frames, i.e., $\frac{1}{\varphi}\sum_{i=1}^{\varphi} d^{\tau}(s^{\tau}, \hat{s}^{\tau})$. We count a frame error if $d^{\tau}(s^{\tau}, \hat{s}^{\tau}) > 0$. Including the rest of the thesis, $n = 300$, $m = 200$, and $k + k' = 100$.

Based on Figure 5.11, two observations are in order. First, we can significantly improve the error performance by making use of code concatenation. With the discussed approach in Section 5.4, we decided on $\Xi = 4$, i.e., the BCH code corrects at least 4 of 50 errors, which results in BCH(63,30) code. Here, the trade-off between the rate R' and the error correction capability appears. Depending on the desired performance, Ξ can be kept high by sacrificing from R', or vice versa. Second, the previously encountered jump between 9 dB and 14 dB in Figure 5.5 remains. We will now take a closer look at this problem.



(a) The comparison of BERs.

(b) The comparison of FERs.

Figure 5.11: The error rate performances of the compound code with and without code concatenation. For both, $n = 300$, $m = 200$, $k = 63$, and $k' = 37$. The BCH(63,30) concatenated code with $\lambda = 1$ has a rate $R_1' = 0.1$. The compound code has a rate $R_2' = 0.21$.

As discussed in Section 5.3, the jump between 9 dB and 14 dB is caused by the LLR expression

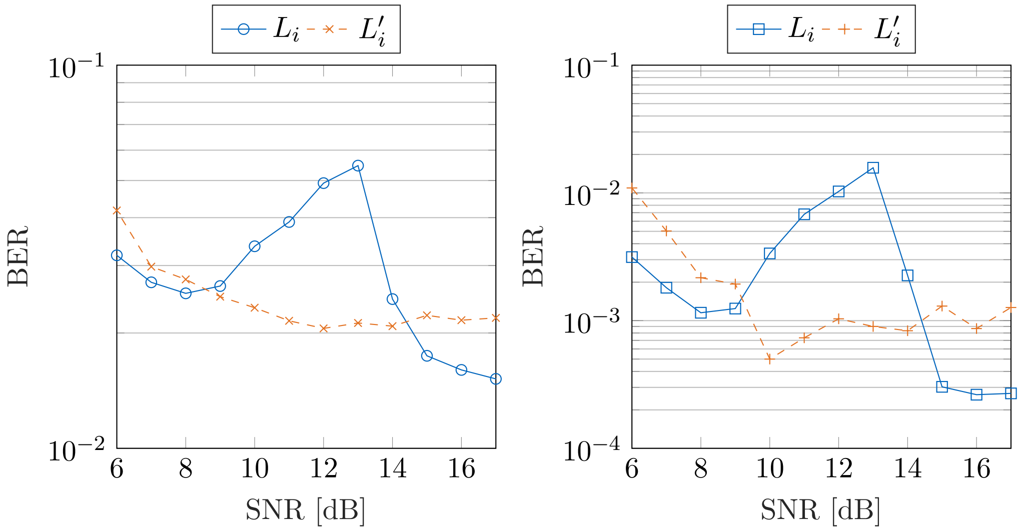
$$L_i = \ln \frac{P_{Y|X}(y_i|0)}{P_{Y|X}(y_i|1)} = \frac{1 - 2y_i}{2\sigma^2} + \ln \frac{P_X(0)}{P_X(1)}. \tag{5.4}$$

The inefficiency of (5.4) in the specified SNR range comes from the utilization of dither noise in our encoding and decoding algorithms, as well as the potential consequences arising from the compound code. To overcome this issue, one can inform the receiver in advance and make necessary modifications to (5.4) specifically for the problematic SNR range. Thereby, a more favorable performance can be achieved. We now replace the LLR expression with

$$L_i' = \frac{1 - 2y_i}{2\sigma} + \ln \frac{P_X(0)}{P_X(1)}, \tag{5.5}$$

where we simply multiply the first term of L_i by σ. Figure 5.12 shows the comparison of these two LLR definitions in terms of BER performance. Figure 5.12a is the compound code with $k = 50$ while Figure 5.12b is the BCH(63,30) concatenated code with $\lambda = 1$. The results clearly show that the unexpected jump is due to SNR mismatch, and better results can be obtained by modifying the LLR expression appropriately, e.g., for each SNR in the figures, the LLR definition which gives the minimum error can be selected.

We have presented only one approach here and showed that this problem can be solved. Further optimization is out of the scope of this thesis, and we will not consider this problem in the remaining results.



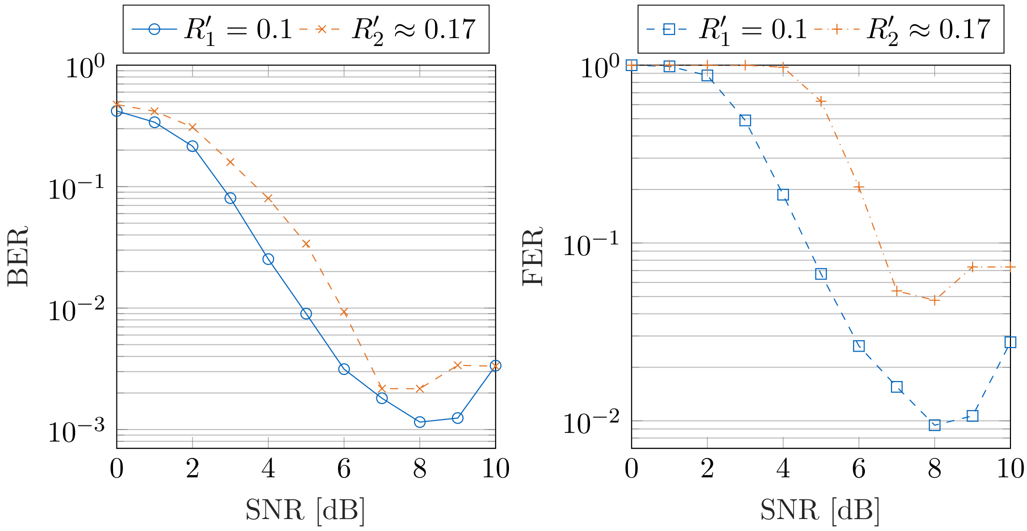
(a) The compound code with $k = 50$ and $R' = 0.17$.

(b) The BCH(63,30) concatenated code with $\lambda = 1$ and $R' = 0.1$.

Figure 5.12: BER performances of the LLR definitions L_i and L'_i, which are given by (5.4) and (5.5), respectively, with $n = 300$, $m = 200$, and $k + k' = 100$.

Based on the trade-off considerations we discussed in Section 5.4, we proposed to concatenate one BCH encoder with multiple LDGM/LDPC encoders. Figure 5.13 compares the performances of the concatenated codes BCH(63,30) with $\lambda = 1$ and BCH(511,349) with $\lambda = 7$. BCH(63,30) provides the rate $R'_1 = 0.1$ while BCH(511,349) has the rate $R'_2 \approx 0.17$. For this example, our proposal with multiple encoders allows us the following. When $k+k'$ is fixed, which we always assume that it is 100, the maximum k we can choose for the BCH encoder with $\lambda = 1$ is 63, see Table 5.1. Ideally, we prefer to choose k as high as possible to increase the rate. Note that BCH(511,349) gives $k = \kappa/\lambda = 73$. For similar error correction capabilities, we got larger τ and increased $R' = \tau/\lambda n$. However, as we expected, the frame errors increased at low SNRs as the frame size increased.

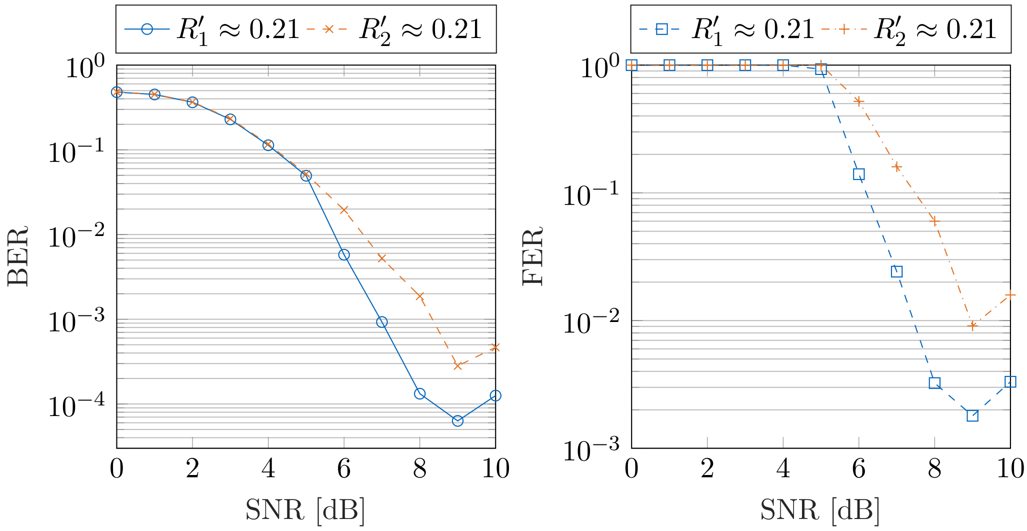
Figure 5.14 shows the performances of the concatenated codes BCH(1023,708) with $\lambda = 11$ and BCH(2047,1453) with $\lambda = 23$. BCH(1023,708) and BCH(2047,1453) almost give similar rates, which are $R'_1 \approx R'_2 \approx 0.21$, respectively. Here, we wanted to see the performance of different λ's which provide similar rates, and it can be seen that setting a larger λ worsened the error performance for the same R'.



(a) The comparison of BERs.

(b) The comparison of FERs.

Figure 5.13: The error rate performances of the concatenated codes BCH(63,30) with $\lambda = 1$ and the rate $R'_1 = 0.1$, and BCH(511,349) with $\lambda = 7$ and the rate $R'_2 \approx 0.17$. For both, $n = 300$, $m = 200$, and $k + k' = 100$.



(a) The comparison of BERs.

(b) The comparison of FERs.

Figure 5.14: The error rate performances of the concatenated codes BCH(1023,708) with $\lambda = 11$ and the rate $R'_1 \approx 0.21$, and BCH(2047,1453) with $\lambda = 23$ and the rate $R'_2 \approx 0.21$. For both, $n = 300$, $m = 200$, and $k + k' = 100$.

Finally, Figure 5.15 compares the BCH concatenated codes with shaped polar codes (see, e.g., [WSSY19]) as a benchmark. There, the polar codes are with lengths 512 and 256, and with the same rates $R'_1 = R'_2 \approx 0.21$, respectively. The concatenated codes are BCH(1023,708) with $\lambda = 11$ and BCH(2047,1453) with $\lambda = 23$, and again with the similar rates $R'_3 \approx R'_4 \approx 0.21$. Looking at the figure, we can conclude that although we have put a lot of effort and improved the performance of the LDGM/LDPC compound codes significantly, there is still much to be developed when compared to the performance of polar codes.

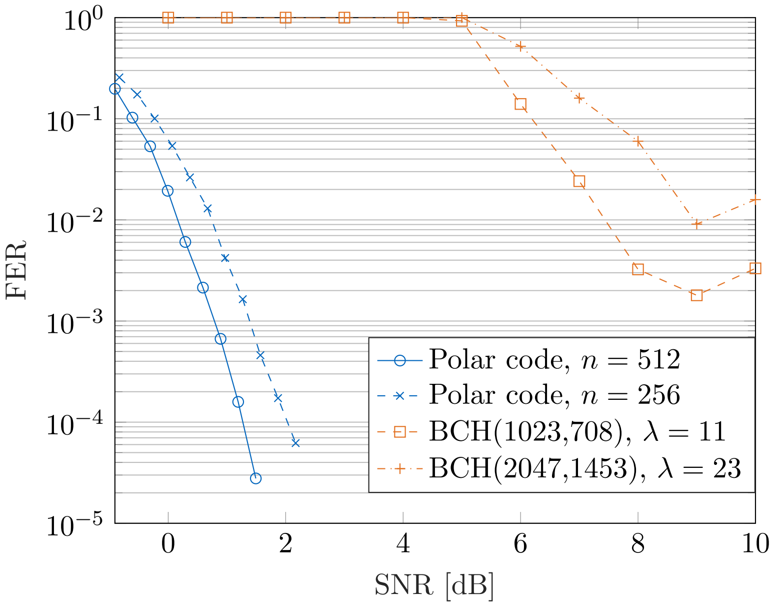


Figure 5.15: The FER performances comparison of the BCH concatenated codes with shaped polar codes. The polar codes are with length 512 and the rate $R'_1 \approx 0.21$, and with length 256 and the rate $R'_2 \approx 0.21$. The concatenated codes are BCH(1023,708) with $\lambda = 11$ and the rate $R'_3 \approx 0.21$, and BCH(2047,1453) with $\lambda = 23$ and the rate $R'_4 \approx 0.21$. For both concatenated codes, $n = 300$, $m = 200$, and $k + k' = 100$.

6 Conclusion

We have presented two MPAs for the LDGM/LDPC compound code, including both encoding and decoding. We showed that half of the frames can be correctly encoded, and the correctly encoded frames can be successfully decoded. To overcome the erroneously encoded frames, we made use of code concatenation and discussed the trade-off between the rate and error correction capability. With the simulation results performed for the binary CCSI over AWGN channel, we can arrive at the following conclusions.

A non-uniformly distributed channel input is obtained for the uniformly distributed message input, e.g., $P_X(1) \approx 0.34$ for the compound code of rate $R = 1/3$, which is also equal to the average symbol energy E_s for OOK modulated symbols. The error floors caused by the encoder can be decreased with the concatenation of the outer BCH code. The unexpected jump that has appeared in the specific SNR range can be avoided by carefully designing the LLR expressions. By concatenating one BCH encoder with multiple LDGM/LDPC encoders, the overall rate R' can be increased. However, a higher number of subframes λ does not necessarily mean a higher rate and better error performance, or vice versa. The trade-off between the rate and the error correction capability can be determined by setting Ξ. Although it is claimed in the literature that the coding scheme with the compound code is expected to perform well, it falls short in comparison to alternative schemes like polar codes.

Further research may address the shortcomings discussed so far, including the following. Of course, the proposed encoding algorithm can be improved in terms of both quantization and valid codewords producing performance, i.e., if $u^m \cdot \mathbf{H_1}^T = s^k$ and $u^m \cdot \mathbf{H_2}^T = 0^{k'}$. The decoder algorithm can be improved to avoid the dither noise used for the intermediate variable nodes, and a more systematic solution to the SNR mismatch issue can be brought. The relationship between the coding algorithm and the outer code can be examined by testing different codes other than narrow-sense BCH codes. The code construction, i.e., the generation of the parity-check and generator matrices, is a research topic itself and should be investigated better for our cases.

Bibliography

[Ari09] E. Arikan, “Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels,” *IEEE Transactions on Information Theory*, vol. 55, no. 7, pp. 3051–3073, 2009.

[Art18] Art of the Problem, “Hamming & low density parity check codes,” YouTube video, 2018, accessed: June 4, 2023. [Online]. Available: https://youtu.be/RWUxtGh-guY

[BRC60] R. C. Bose and D. K. Ray-Chaudhuri, “On a class of error correcting binary group codes,” *Information and Control*, vol. 3, no. 1, pp. 68–79, 1960.

[BSS17] G. Böcherer, F. Steiner, and P. Schulte, “Fast probabilistic shaping implementation for long-haul fiber-optic communication systems,” in *2017 European Conference on Optical Communication (ECOC)*, 2017, pp. 1–3.

[CJC13] G. C. Clark Jr and J. B. Cain, *Error-correction coding for digital communications.* Springer Science & Business Media, 2013.

[Cos83] M. Costa, “Writing on dirty paper (corresp.),” *IEEE Transactions on Information Theory*, vol. 29, no. 3, pp. 439–441, 1983.

[CT06] T. Cover and J. A. Thomas, *Elements of information theory.* Wiley-Interscience, 2006.

[EA20] A. Elzanaty and M.-S. Alouini, “Adaptive coded modulation for IM/DD free-space optical backhauling: A probabilistic shaping approach,” *IEEE Transactions on Communications*, vol. 68, no. 10, pp. 6388–6402, 2020.

[EtB05] U. Erez and S. ten Brink, “A close-to-capacity dirty paper coding scheme,” *IEEE Transactions on Information Theory*, vol. 51, no. 10, pp. 3417–3432, 2005.

[Gal62] R. Gallager, “Low-density parity-check codes,” *IRE Transactions on Information Theory*, vol. 8, no. 1, pp. 21–28, 1962.

[GP80] S. I. Gel'fand and M. S. Pinsker, "Coding for channel with random parameters," *Problems of Control and Information Theory*, vol. 9, no. 1, pp. 19–31, 1980.

[Hoc59] A. Hocquenghem, "Codes correcteurs d'erreurs," *Chiffres*, vol. 2, pp. 147–156, 1959.

[Huf52] D. A. Huffman, "A method for the construction of minimum-redundancy codes," *Proceedings of the IRE*, vol. 40, no. 9, pp. 1098–1101, 1952.

[KFL01] F. Kschischang, B. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 498–519, 2001.

[KT08] F. Kayhan and T. Tanaka, "A practical low-density coding scheme for binary dirty paper channel," in *2008 5th International Symposium on Turbo Codes and Related Topics*, 2008, pp. 396–401.

[KVNP14] S. Kumar, A. Vem, K. Narayanan, and H. D. Pfister, "Spatially-coupled codes for side-information problems," in *2014 IEEE International Symposium on Information Theory*, 2014, pp. 516–520.

[LC01] S. Lin and D. J. Costello, *Error control coding.* Prentice hall Lebanon, IN, 2001, vol. 2, no. 4.

[MN97] D. J. MacKay and R. M. Neal, "Near Shannon limit performance of low density parity check codes," *Electronics Letters*, vol. 33, no. 6, pp. 457–458, 1997.

[Reg09] P. A. Regalia, "A modified belief propagation algorithm for code word quantization," *IEEE Transactions on Communications*, vol. 57, no. 12, pp. 3513–3517, 2009.

[RFY$^+$17] J. Renner, T. Fehenberger, M. P. Yankov, F. Da Ros, S. Forchhammer, G. Böcherer, and N. Hanik, "Experimental comparison of probabilistic shaping methods for unrepeated fiber transmission," *Journal of Lightwave Technology*, vol. 35, no. 22, pp. 4871–4879, 2017.

[Sha48] C. E. Shannon, "A mathematical theory of communication," *The Bell System Technical Journal*, vol. 27, no. 3, pp. 379–423, 1948.

[Sha59] ——, “Coding theorems for a discrete source with a fidelity criterion,” in *IRE International Convention Record*, March 1959, pp. 142–163.

[Tan81] R. Tanner, “A recursive approach to low complexity codes,” *IEEE Transactions on Information Theory*, vol. 27, no. 5, pp. 533–547, 1981.

[Tho03] J. Thorpe, “Low-density parity-check (LDPC) codes constructed from protographs,” *Interplanetary Network Progress Report*, vol. 42, no. 154, pp. 42–154, 2003.

[TMZ06] R. Tu, Y. Mao, and J. Zhao, “On generalized survey propagation: Normal realization and sum-product interpretation,” in *2006 IEEE International Symposium on Information Theory*, 2006, pp. 2042–2046.

[WH09] Q. Wang and C. He, “Practical dirty paper coding with nested binary LDGM-LDPC codes,” in *2009 IEEE International Conference on Communications*, 2009, pp. 1–6.

[WM09] M. J. Wainwright and E. Martinian, “Low-density graph codes that are optimal for binning and coding with side information,” *IEEE Transactions on Information Theory*, vol. 55, no. 3, pp. 1061–1079, 2009.

[WMM10] M. J. Wainwright, E. Maneva, and E. Martinian, “Lossy source compression using low-density generator matrix codes: Analysis and algorithms,” *IEEE Transactions on Information Theory*, vol. 56, no. 3, pp. 1351–1368, 2010.

[WSSY19] T. Wiegart, F. Steiner, P. Schulte, and P. Yuan, “Shaped on–off keying using polar codes,” *IEEE Communications Letters*, vol. 23, no. 11, pp. 1922–1926, 2019.

[WZ76] A. Wyner and J. Ziv, “The rate-distortion function for source coding with side information at the decoder,” *IEEE Transactions on Information Theory*, vol. 22, no. 1, pp. 1–10, 1976.

[YFW$^+$03] J. S. Yedidia, W. T. Freeman, Y. Weiss *et al.*, “Understanding belief propagation and its generalizations,” *Exploring Artificial Intelligence in the New Millenium*, vol. 8, no. 236-239, pp. 0018–9448, 2003.

[ZZB05] J. Zhao, F. Zarkeshvari, and A. Banihashemi, “On implementation of min-sum algorithm and its modifications for decoding low-density parity-check (LDPC) codes,” *IEEE Transactions on Communications*, vol. 53, no. 4, pp. 549–554, 2005.