Dissertation

Information-optimal Decoding and Demodulation on Sparse Graphs

ausgeführt zum Zwecke der Erlangung des akademischen Grades eines
Doktors der technischen Wissenschaften

unter der Leitung von
AO.UNIV.-PROF. DIPL.-ING. DR.TECHN. GERALD MATZ
Institute of Telecommunications

eingereicht an der Technischen Universität Wien
Fakultät für Elektrotechnik und Informationstechnik

von
DIPL.-ING. MICHAEL MEIDLINGER
Matr. Nr.: 0728169

Wien, Februar 2018

Die Begutachtung dieser Arbeit erfolgte durch:

1. Prof. Dipl.-Ing. Dr.sc.techn. Andreas Burg
   Telecommunications Circuits Laboratory
   Éole polytechnique fédérale de Lausanne

2. Univ.-Prof. Dipl.-Ing. Dr.-Ing. Norbert Görtz
   Institute of Telecommunications
   Technische Universität Wien

Abstract

In this thesis, we consider information-optimal quantization of Gaussian random variables and derive discrete information-optimal decoders for low-density parity-check (LDPC) codes. Moreover, a novel joint decoding and demodulation approach for transmission over continuous input additive white noise channels is proposed.

After a brief revision of related concepts, we begin by discussing information-optimal quantization of Gaussian random vectors. Here, information-optimal means that the quantization preserves information on a second variable that is correlated with the quantizer input. We study the rate information tradeoff for this setting which characterizes the highest amount of information that can be retained for any given quantization rate. Furthermore, we establish a fundamental connection between information-optimal quantization and linearly preprocessed mean-square error (MSE)-optimal rate distortion quantization based on concepts related to Wiener filtering. We then use this connection to obtain information-optimal quantizer designs from well-known MSE-optimal designs.

Next, we use the principle of information-optimality to design low-resolution discrete message passing LDPC decoders based on look-up tables (LUTs). We show that there is a connection between LUT decoding and belief propagation and use this to derive low-complexity hybrid decoding approaches. Special attention is paid to LUT decoding for irregular LDPC codes, for which we derive jointly optimal LUT designs and propose a strategy to optimize the degree distributions of irregular codes for LUT decoding. The so obtained decoders outperform conventional min-sum decoders at floating point precision at LUT resolutions as low as 3 bit for regular and 4 bit for irregular codes.

Subsequently, we introduce superposition modulated low-density parity-check (SMLDPC) codes — a new class of codes for high-rate transmission over continuous input channels. SMLDPC codes are obtained by a concatenation of LDPC coding and superposition modulation (SM) and allow for parallel decoding and demodulation on a joint sparse graph without the need of deinterleaving or an explicit demodulation step. By using different edge coefficients for SM, a wide variety of modulation schemes can be adopted by SMLDPC codes, including the well-known regular QAM constellations. We show that SMLDPC codes exhibit the same concentration and thresholding phenomenon as LDPC codes, where the thresholds can be computed exactly for belief propagation decoding. This gives rise to a code optimization approach based on ensembles, i.e., we propose to optimize the distributions that characterize

the codes rather than the codes themselves. A particular code is then obtained by drawing a random sample from the ensemble and the concentration theorem states that for long block lengths, any code obtained that way will perform as predicted by the ensemble threshold.

Kurzfassung

In der vorliegenden Dissertation beschäftigen wir uns mit informations-optimaler Kodierung, angewandt zur Quantisierung gaußscher Zufallsvariablen sowie zur Konzeptionierung diskreter, informations-optimaler Dekodierverfahren. Darüber hinaus schlagen wir ein neuartiges Verfahren für die vorwärtsfehlerkorrigierte Datenübertragung über additive Rauschkanäle vor, welches zeitgleiche Demodulierung und Dekodierung mit linearer Komplexität ermöglicht.

Nach einem kurzen Überblick über die relevante Literatur beginnen wir mit einer Diskussion über die informations-optimale Quantisierung gaußscher Zufallsvektoren. "Informations-optimal" bedeutet in diesem Zusammenhang, das Quantisierungsverfahren so zu gestalten, dass die Transinformation zwischen der quantisierten, sowie einer zweiten, mit dem Quantisierereingang korrelierten Zufallsvariable, maximal wird. Für diesen Fall betrachten wir den sogenannten Informations-Raten-Abgleich, d.h. die maximal mögliche Transinformation unter der Vorgabe einer höchstzulässigen Kompressionsrate. Wir leiten eine fundamentale Verbindung zwischen informations-optimaler und linear gefilterter, MSE[1] -optimaler Quantisierung her, die eng mit den Konzepten der Wiener-Filterung im Zusammenhang steht. Anschließend nutzen wir diese Verbindung, um aus bewährten, MSE-optimalen Quantisierungsverfahren informations-optimale Verfahren abzuleiten.

Im Weiteren benutzen wir das Prinzip der Informations-Optimalität zur Konstruktion diskreter, LUT[2]-basierter LDPC[3]-Dekoder mit geringer Bitbreite. Wir leiten einen Zusammenhang zwischen LUT-Dekodierung und dem *belief propagation*- Dekodierverfahren her, welchen wir für das Design hybrider Dekodierverfahren mit geringer Komplexität und hoher Fehlertoleranz heranziehen. Besondere Aufmerksamkeit schenken wir dabei der LUT-Dekodierung irregulärer LDPC-Codes, für die wir gemeinsam optimierte LUT-Designs ableiten und ein Verfahren zur Optimierung der irregulären Codegraphenstruktur unter der Berücksichtigung von LUTs vorschlagen. Die so erzeugten Decoder liefern niedrigere Fehlerraten als konventionelle *min-sum*-Decoder mit Gleitkommapräzision bei einer LUT-Auflösung von nur 3 Bit bei regulären und 4 Bit bei irregulären Codes.

Letztlich führen wir SMLDPC[4]-Codes neu ein. Dabei handelt es sich um eine neuarti-

[1] *engl.: mean square error*, mittlerer, quadratischer Fehler

[2] *engl.: look-up table*, Umsetzungstabelle

[3] *engl.: low-density parity-check code*, siehe [32, 79].

[4] *engl.: superposition modulated low-density parity-check*

ge Klasse von Vorwärtsfehlerkorrektur-Codes mit hoher Rate für Rauschkanalübertragungen. SMLDPC-Codes ergeben sich durch eine Kombination aus LDPC-Codes und Superpositions Modulation (SM) und erlauben zeitgleiche Datendemodulierung und –dekodierung mithilfe spärlich besetzter Graphen. Indem verschiedene Gewichte für die SM herangezogen werden, decken SMLDPC-Codes ein weites Spektrum unterschiedlichster Modulationsarten ab – darunter die weitverbreitete Quadraturamplitudenmodulation (QAM). Wir liefern einen mathematisch exakten Beweis, dass SMLDPC-Codes das gleiche Konzentrations– und Grenzwertverhalten wie konventionelle LPDC-Codes aufweisen, d.h. dass für den Grenzfall unendlich langer Codes alle zufällig erzeugten Codes aus dem gleichen Ensemble dasselbe Fehlerkorrekturverhalten aufweisen, und sich dieses durch den Mittelwert zyklenfreier Codegraphen beschreiben lässt. Darüber hinaus zeigen wir, wie sich dieser Mittelwert für *belief propagation*-Decodierung exakt berechnen lässt. Daraus ergibt sich ein Verfahren zur Codeoptimierung, bei dem wir nicht einzelne Codes, sondern die Verteilung des gesamten Code-Ensembles optimieren. Das Konzentrationstheorem besagt dann, dass sich ein zufällig aus einem Ensemble ausgewählter Code ebenso verhält wie der Ensemble-Mittelwert.

Acknowledgements / Danksagung

First of all, I would like to extend my gratitude to my advisor Gerald Matz, for trusting in my abilities and granting me the freedom to pursue own ideas, as well as for his continuous support and encouragement.

I would also like to thank Andreas Burg for giving me the opportunity for a research visit at the telecommunications circuits lab (TCL) at EPFL and for acting as an examinor for this thesis. At this point, I would also like to thank Alex and Reza from the TCL for the good cooperation on our joint work on quantized LDPC decoders.

Special thanks go to my colleagues at the Institute of Telecommunications at TU Wien, for making my time there a worthwhile experience. In particular to Martin, for being the friend you are, to Georg, for being a great TA colleague and to Stefan, for always having an open ear.

Meinen Eltern Gerlinde und Josef danke ich vom ganzem Herzen für die Liebe und Unterstützung die sie mir zeit meines Lebens entgegengebrach haben.

Ganz besonderer Dank gebührt Dir, liebe Pia!

Contents

Chapter 1

Introduction

1.1 Motivation and Scope of Work

It has now been almost exactly 70 years since Claude E. Shannon established the information-theoretic concept of channel capacity, i.e., the highest possible rate for which data can be transmitted reliably over a given communication channel [87]. While Shannon proved the existence of codes capable of achieving capacity, the search for practical coding schemes that come close to that theoretical limit has been an ongoing engineering problem ever since. A major breakthrough was made in the early 1990s when the invention of turbo codes [8] demonstrated that approaching channel capacity was possible using linear time, iterative decoding techniques. This ignited major interest in iterative decoding, eventually culminating into the discovery of irregular low-density parity-check (LDPC) codes [63, 77] — a class of codes that has been shown to come extremely close to capacity [18] and that we will therefore pay special attention to in this thesis. It was soon recognized that most iterative decoding and demodulation algorithms can be formulated in a general framework based on sparse, bipartite factor graphs [56, 117], where the sparsity of the underlying graphs is the reason why decoding with complexity linear in codeword length is possible. At the same time, the optimal decoding procedure in the sense of minimum bit error rate was identified as the sum-product (also known as belief propagation (BP)) algorithm[1]. For binary codes, BP decoding amounts to an iterative exchange of continuous messages through the code's factor graph, where the soft messages represent log-likelihood ratios (LLRs) that capture the probabilities that a certain code bit was either 0 or 1.

Clearly, if such a decoding scheme is to be implemented in a real world system, continuous LLR values must be approximated by numbers of finite resolution. This issue has become all the more pressing due to the success and resulting wide adoption of iterative soft information based decoding into practical systems. Since the soft messages carry information on the transmit data, it appears natural to search for an *information-optimal* representation. That is, the quantization should be conducted in a way that preserves as much information on the transmit data as possible.

[1] Strictly speaking, BP is optimal only on factor graphs without cycles

Going one step further, we can consider algorithms that are specifically designed to work with discrete messages. The reasoning here is that rather than using a decoding algorithm that is originally based on continuous LLRs and use it on quantized soft messages, one could select an algorithm that works with discrete messages in the first place. The principle of information-optimality can be applied to design such algorithms by requiring that relevant information is preserved throughout the entire iterative decoding process.

Furthermore, from an implementation perspective, there is an incentive to transition to a discrete, low resolution representation of the receive signal even before decoding, ideally already at the demodulation step — arriving at at an entirely discrete iterative receiver that takes quantized receive values as input and produces an estimate of the transmit data sequence by iterative processing of discrete messages. However, such a comprehensive approach is complicated by the prevalence of the bit-interleaved coded modulation (BICM) paradigm [12, 30] which clearly separates decoding and demodulation.

In light of the above, this thesis focuses on following aspects in particular.

Fundamentals of Information-Optimal Quantization

Historically, the fundamental theory of quantization is Shannon's rate distortion theory [86], which characterizes the ultimate tradeoff between compression and resulting signal quality loss. Information-optimal quantization takes a different approach in that it replaces signal distortion as fidelity measure with the mutual information between the compressed source and a relevance variable, an approach that has first been formulated in [99] in the context of the information bottleneck method (IBM). The IBM has been successfully applied to various problems in machine learning [92], computer vision [36], biomedical signal processing [84], and communications [110, 119]. It is also inherently better suited than rate distortion quantization for channel output compression in a communication system [110, 119].

In this thesis, we are interested in the optimal rate information tradeoff for continuous sources, i.e., the highest amount of information that can be preserved at a given compression rate when transitioning from a continuous to a discrete signal representation. We mainly focus on Gaussian random variables, because for that case, analytic expressions can be obtained. Furthermore, the Gaussian case is relevant in the context of receive signal quantization, since it has been shown that in order to achieve capacity over a Gaussian channel, the channel outputs must follow a Gaussian distribution. While the tradeoff between quantization accuracy and rate for that case is well described by the Gaussian information bottleneck (GIB) [16], the design of information-optimal quantizers is still an open problem. Due to the existence of a wide variety of MSE-optimal quantization schemes, we are particularly interested in the connection between information-optimal and conventional MSE-optimal quantization approaches, since existing MSE-optimal approaches can potentially be adapted to obtain information-optimal quantizers.

Discrete Iterative Decoding of LDPC Codes

Irregular LDPC codes [63, 77] outperform the best known turbo codes and approach channel capacity using iterative, linear-time message passing decoding algorithms. In view of ever increasing data rates of modern communication systems, LDPC codes are becoming increasingly attractive due to their superior error floor performance and the possibility for fully parallel, ultra high speed decoding. Consequently, many practical systems, ranging from flash storage [106, 127], to communication standards such as DVB-S2 [29], WiMAX [50], 802.11n and 802.11ac WiFi [51, 52], as well as 10GBASE-T Ethernet [53] rely on LDPC codes for error correction.

In the light of the vast spectrum of applications, efforts have been made to devise optimized decoding algorithms that are well suited for practical hardware implementations [17, 126], most of which are variants or approximations of the well known BP algorithm. These algorithms rely on processing of continuous LLRs which are encoded as fixed point numbers with 5 to 7 bit resolution in most hardware implementations. Low resolutions are desirable to reduce decoder complexity but can deteriorate error rate performance [122, 123, 125, 126].

Recently, there has been significant interest in *finite alphabet* LDPC decoding [25, 59–61, 80], often also referred to as *look-up table (LUT)* decoding. Rather than performing the conventional BP arithmetics on quantized message representations, discrete LUT mappings directly map incoming to outgoing integer messages. In LUT decoding, algebraic structure and the probabilistic LLR interpretation of messages is sacrificed in exchange for very low message resolutions. The main advantage of this approach is a drastic reduction in message bit-width, while at the same time maintaining and in some instances even improving error rate performance of traditional message passing schemes.

In this thesis, we consider mainly information-optimal LUT updates [59, 80, 81]. Furthermore, we are interested in how LUT based decoding algorithms relate to the optimal BP algorithm. While conventional BP is theoretically well understood, LUT decoding has been mainly studied from an application point of view and lacks important theoretical concepts such as symmetry, message consistency and asymptotic stability considerations. Moreover, a coherent theory of LUT and BP decoding allows for a combination of techniques from both approaches, i.e., algorithms that combine the algebraic updates from BP-type algorithms with LUT mappings. In order to seize the full potential of LDPC codes, we pay special attention to the study of LUT decoders for irregular codes.

Challenging the BICM Paradigm — Joint Decoding and Demodulation on Sparse Graphs

Breaking up large problems into several smaller, more tractable ones is generally one of the key approaches in technical science and engineering. The BICM paradigm follows that approach by promoting the separation of decoding and demodulation as two tasks to be considered individually. However, there may be aspects of the overall problem that we miss when taking such an approach. In particular, with the separation between coding and demodulation in place, we found it difficult to extend the concept of information-optimal LUT based message updates to also include iterative demodulation using discrete signal representations.

Furthermore, in view of Shannon's coding theorem for Gaussian channels, it appears that classical, widely used higher order modulation schemes such as QAM are a somewhat arbitrary component to the overall channel code. Clearly, there are practical reasons why these schemes are popular; however, from a theoretical perspective where we enjoy the freedom to select any arbitrary scheme in order to approach capacity, the restriction to those classical schemes does not appear to be justified.

Both of the two above points are addressed by low-density hybrid-check (LDHC) codes [114, 115]: First, SM [46] is used as a modulation scheme, offering a wider range of constellation shapes including constellations that approach a Gaussian distribution and therefore, are potentially capacity achieving. Second, LDHC formulate decoding and demodulation as a joint problem on a single sparse graph. However, LDHC still adhere to the BICM paradigm, since the joint graph structure is merely a way of interleaver design to properly match the demodulation and decoding subgraphs [46, 113]. Thus, while [114, 115] optimize the decoding subgraph given the demodulation subgraph and vice versa, the full potential of this approach seems to be obscured by the lack of a joint design for both components.

We therefore propose SMLDPC codes in this thesis — a novel class of channel codes with a unified approach to iterative decoding and demodulation. SMLDPC codes are an extension to both LDPC and LDHC codes, and much like LDHC, are obtained by a concatenation of binary LDPC coding and SM. In that sense, SMLDPC codes directly map discrete messages to real or complex channel input sequences, with a corresponding inverse operation at the receiver, i.e., there is no separate demodulation step. In terms of decoding, the underlying sparse graph structure of SMLDPC codes allows for parallel decoding and demodulation based on message passing with complexity linear in codeword length. The symmetrization between demodulation and decoding that is guaranteed by interleaving in case of BICM is achieved by using a random construction of SM coefficients for the modulation part of SMLDPC codes. Thus, in many ways, the random construction of SMLDPC codes resembles the construction of Shannon's random codebooks [87]. Due to the freedom of choice for the modulation coefficients, SMLDPC codes bear the potential to approach the capacity of the additive white Gaussian noise (AWGN) channel in the high-rate regime, while at the same time, the underlying sparse structure allows for low complexity iterative decoding.

1.2 Outline and Contributions

Chapter 2: Preliminaries

In Chapter 2, we provide a brief overview of the related literature. The goal here is to familiarize the reader with the most important concepts used throughout the thesis and provide a common point of reference for those concepts.

Chapter 3: Rate Information Coding for Gaussian Signals

In Chapter 3, we study the rate information tradeoff for the case of jointly Gaussian vectors. More specifically, we show that the Gaussian information bottleneck (GIB) [16], which achieves the optimal tradeoff, is closely related to minimum mean square error (MMSE) estimation. By formulating the GIB in a communication context with relevance and observation related via a linear channel plus additive noise model, we demonstrate that the GIB solution is a function of the eigenvalues of the associated Wiener filter [108] for estimating the transformed channel input from the channel output. Furthermore, we show that the optimal GIB trade-off can also be accomplished by linear filtering followed by MSE-optimal source coding. Somewhat surprisingly, the optimal linear filter here is given by the square root of the Wiener filter. This is in contrast to the result of Sakrison [82], who showed that for noisy Gaussian source coding problems with MSE distortion the optimal filter is a Wiener filter. Our results also explain why direct MSE-optimal source coding (i.e., without filtering) in general does not achieve the optimal rate information tradeoff. We then give operational meaning to the information-theoretic tradeoff problems by stating a coding theorem for a sequence of Gaussian vectors. In conjunction with the previous results regarding the relation of MSE- and information-optimal quantization, this proves that information-optimal quantizers obtained from MSE-optimal quantizers do indeed exist. However, unlike the result of [82], this does not imply that square-root-Wiener-filtering followed by MSE-optimal quantization is always information-optimal. Thus, we conduct numerical simulations investigating the rate information performance for various comparable filters prior to MSE-optimal quantization and confirm that the square root Wiener filter indeed yields the pest performance. Subsequently, we also consider extensions from the vector case to stationary ergodic processes. Finally, we also investigate whether the information rate tradeoff characterizes the capacity under channel output quantization but find that this is not the case.

Parts of this chapter have been originally published in [70].

Chapter 4: Quantized Message Passing LDPC Decoding

In this chapter, we develop look-up table (LUT) decoders for irregular LDPC codes and show that the capacity approaching performance of irregular LDPC codes can be achieved with very low resolution LUT decoders. Furthermore, we establish a fundamental relation between LUT decoding and BP decoding that gives insight into the way LUT decoding works on a general level, irrespective of the particular LUT design.

We begin our derivations by establishing a symmetry concept for discrete messages. Exploiting this symmetry, we derive a density evolution based duality between discrete messages as processed by LUT decoders and real LLR values used in BP. We then use this duality to develop a simple algebraic structure on the discrete messages and show how LUT decoding is essentially a variant of BP, where outgoing messages are projected onto a finite set after each message update.

Next, we use the symmetry and algebraic structure to derive low-complexity LUT decoders based on the quantizer design algorithm from [58]. Due to the symmetry requirements, we manage to drastically reduce the implementation and design complexity for the resulting LUTs. Furthermore, we propose the hybrid min-LUT decoding algorithm, that partly relies on generic LUT updates but also takes advantage of the algebraic structure on the discrete messages. Furthermore, we introduce LUT trees, i.e., a hierarchical structure of LUTs that serves to reduce implementation complexity. We find that maximum-width binary trees are optimal both in terms of performance and from an implementation point of view.

Building on those concepts, we develop a novel extension of LUT decoding to irregular LDPC codes. Exploiting the analogy to BP, we derive an asymptotic stability condition for LUT decoding in the spirit of [77]. The stability condition explains why most LDPC degree distributions (DDs) in the literature are not suited for LUT decoding: they are optimized for the BP algorithm and violate the LUT stability criterion. Thus, we propose to optimize DDs for LUT decoding and demonstrate that irregular LDPC codes generated from those DDs perform better than regular LDPC codes under LUT decoding. We also consider further measures to reduce decoder complexity such as LUT reuse and LUT down-sizing and investigate the impact on decoder performance. Lastly, we present an extensive numerical analysis and provide the first comparison between different LUT-based decoders in terms of error rate and density evolution threshold simulations. As a consequence, a performance-complexity tradeoff is established, which can be used to tailor a specific decoder to error rate requirements and implementation constraints.

The content of this chapter has been previously published in [67–69]. Furthermore, decoder implementations following our LUT design have been published in [6, 33].

Chapter 5: SMLDPC Codes

In Chapter 5, we propose superposition modulated low-density parity-check (SMLDPC) codes, which constitute an entirely new approach to joint decoding and demodulation. First, we introduce the general construction of SMLDPC codes which is based on a concatenation of binary LDPC coding followed by superposition modulation (SM). This gives rise to a joint sparse graph structure that can be exploited for fully parallel decoding and demodulation with complexity linear in codeword length.

Unlike previous work on SM [46, 64], we proceed to take a more general approach towards SM by allowing for a wider range of modulation coefficients, even covering the case of multiple-input and multiple-output (MIMO) transmission. This results in very granular control over the extrinsic information transfer (EXIT) function of the demodulation component which serves to prevent rate loss and enables an accurate matching between the decoding and demodulation component codes.

Subsequently, we illustrate how decoding of SMLDPC codes can be performed using message passing on the joint sparse graph. In particular, we derive the BP decoding algorithm for SMLDPC codes, where we pay special attention to the demodulation part of the algorithm — the so called symbol node (SN) updates. For joint modulation of multiple channel input symbols, we derive novel low complexity SN updates which resemble linear MMSE estimators with a priori information [103].

Next, we derive the fundamental asymptotic theory of SMLDPC codes. Our main result states that for any particular random realization of SMLDPC code graph, modulation coefficients, transmit codeword and channel realization, the number of bit errors under message passing decoding converges to the average of the cycle-free case exponentially fast in the length of the codeword. Furthermore, we prove that under BP decoding, monotonicity regarding the channel quality and the average fraction of bit errors holds. Combining these two results, we can conclude that long SMLDPC codes exhibit the same thresholding phenomenon as LDPC codes [78], where the channels for which reliable transmission is possible can be clearly separated from those where it is not.

We continue our analysis by investigating the cycle-free case. We show that under fairly general symmetry assumptions that hold for the case of BP decoding, the densities of the messages are symmetric and independent and identically distributed (iid). Unfortunately, unlike for LDPC codes, we cannot explicitly obtain those densities due to the complicated structure of the SN updates. For BP however, we show that message densities are LLR-consistent [102] throughout decoding. It follows then that we can determine the associated threshold even without explicit calculation of densities based on EXIT analysis. Furthermore, using LLR-consistency and message symmetry, we obtain very good approximations of the EXIT function of the SN updates.

Subsequently, we introduce irregular SMLDPC codes and discuss their potential. In contrast to LDHC codes [114, 115], we consider overall joint degree distributions of the global graph structure rather than focusing on the marginal distributions of decoding and demodulation subgraphs individually. Furthermore, our notion of irregularity includes irregular SN degrees and channel dimensions. On top of the irregularity of the SMLDPC graph structure, we also define irregularity regarding the SN type. This concept can be used to mix different modulation coefficients or model the performance of codes where different SN updates are used, enabling extremely fine control over the EXIT characteristic of the SNs.

Lastly, we demonstrate how the decoding performance of SMLDPC codes can be analyzed by means of EXIT evolution. Based on this concept, we formulate optimization problems for optimizing the degree distribution of SMLDPC codes in a joint fashion, and present an example of an irregular code design with corresponding threshold and error rate performance simulations.

1.3 Notation

$\mathbb{N}$	natural numbers
$\mathbb{R}$	real numbers
$\mathbb{R}_+$	non-negative real numbers
$\mathbb{C}$	complex numbers
GF(2)	Galois field of size 2
$\lvert\mathcal{A}\rvert$	cardinality of the set $\mathcal{A}$
x	deterministic scalar
x	scalar random variable
$\boldsymbol{x}$	deterministic column vector
$\boldsymbol{\mathsf{x}}$	random vector
$\boldsymbol{X}$	deterministic matrix
$\boldsymbol{\mathsf{X}}$	random matrix
j	$\sqrt{-1}$
$\mathrm{Re}\{z\}$	real part of $z \in \mathbb{C}$
z^*	complex conjugate of $z \in \mathbb{C}$
$\mathrm{P}\{\cdot\}$	probability
$\mathbb{E}[\cdot]$	expectation
$\mathrm{cov}[\cdot]$	covariance (matrix)
$p_{\mathsf{x}}(x)$	probability density/mass function of x. Short hand: $p(x)$
$p_{\mathsf{x}\mid\mathsf{y}}(x\mid y)$	conditional probability density/mass function given y. Short hand: $p(x\mid y)$
$D(\cdot\Vert\cdot)$	Kullback–Leibler divergence
$I(\cdot;\cdot)$	mutual information
$\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{C})$	real Gaussian distribution with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{C}$
$\mathcal{CN}(\boldsymbol{\mu}, \boldsymbol{C})$	complex Gaussian distribution with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{C}$
$\boldsymbol{A}_{\sim k}$	matrix $\boldsymbol{A}$ with the kth column removed
$\boldsymbol{A}^{\neg k}$	matrix $\boldsymbol{A}$ with the sign of the kth column inverted
$\boldsymbol{A}^{\mathrm{T}}$	transpose of matrix $\boldsymbol{A}$
$\boldsymbol{A}^{\mathrm{H}}$	conjugate transpose of matrix $\boldsymbol{A}$
$\boldsymbol{A}[m_1 : m_2, :]$	submatrix of $\boldsymbol{A}$ containing only the rows from m_1 to m_2
$\mathrm{tr}\{\boldsymbol{A}\}$	trace of matrix $\boldsymbol{A}$
$\boldsymbol{I}$	identity matrix
$\mathbf{1}$	all-one column vector
$\mathsf{0}$	all-zero column vector
$\mathrm{diag}\{a_n\}_{n=1}^{N}$	$N \times N$ diagonal matrix with diagonal elements a_n
$\lVert\boldsymbol{x}\rVert$	Euclidean norm
$\boldsymbol{A} \otimes \boldsymbol{B}$	kronecker product of matrices
$p_{\mathsf{x}} \otimes p_{\mathsf{y}}$	convolution of probability densities

$p_{\mathsf{x}}^{\otimes i}$	$i$fold self convolution
$\odot$	element-wise multiplication
$\succcurlyeq$	element-wise inequality
$\delta(x)$	Dirac delta
Δ_y	$\delta(x-y)$
δ_x	Kronecker delta
$\mathbb{1}(x)$	logical indicator function. $\mathbb{1}(x)=1$ if x is true and 0 otherwise
$[x]^+$	$\max(x,0)$
$\log$	natural logarithm
$\log_b$	logarithm to base b
$\log_2^+(x)$	$\log_2 \max(x,1)$
$\operatorname{par}\boldsymbol{x}$	parity $\prod_i x_i$ of a vector $\boldsymbol{x}$

1.4 Acronyms

APP a posteriori probability

AWGN additive white Gaussian noise

BER bit error rate

BI-AWGN binary input additive white Gaussian noise

BICM bit-interleaved coded modulation

BP belief propagation

CN check node

DD degree distribution

EXIT extrinsic information transfer

FER frame error rate

GIB Gaussian information bottleneck

IB information bottleneck

IBM information bottleneck method

iid independent and identically distributed

KLD Kullback-Leibler divergence

LDHC low-density hybrid-check

LDPC low-density parity-check

LLR log-likelihood ratio

LUT look-up table

MAP maximum a posteriori probability

MIMO multiple-input and multiple-output

MS min-sum

MSE mean-square error

PAM pulse amplitude modulation

pmf probability mass function

PSM phase shifted superposition modulation

QAM quadrature amplitude modulation

SM superposition modulation

SM-EPA superposition modulation using equal power allocation

SM-GPA superposition modulation using group power allocation

SM-UPA superposition modulation using unequal power allocation

SMLDPC superposition modulated low-density parity-check

SN symbol node

SNR signal to noise ratio

VN variable node

Chapter 2

Preliminaries

This thesis covers a wide variety of topics, ranging from source- to channel coding and modulation, as well as techniques to design and analyze iteratively decoded and demodulated systems. For that purpose, this chapter introduces the reader to the basic concepts and relevant literature for each of those areas and lays the groundwork for the derivations in later chapters.

In Section 2.1, we introduce the basic theory of lossy source coding as well various optimization problems centering around information-optimal quantization. Material presented in this section is referred to from Chapter 3, where we examine information-optimal source coding for Gaussian signals, as well as from Chapter 4, where the concept of information-optimal quantization is used to design efficient, low resolution LDPC decoders.

Section 2.2 summarizes definitions and major results on LDPC codes. We revisit the material presented here both in Chapter 4 and Chapter 5.

Finally, Section 2.3 introduces EXIT charts as a method of analyzing and designing iterative receivers. The tools presented here are used in Chapter 5 to analyse the convergence behaviour of iterative decoders and find good degree distributions for irregular SMLDPC codes.

2.1 Information-Optimal Quantization

Quantization is the division of a quantity into a discrete number of small parts, often assumed to be integral multiples of a common quantity [40]. In communications, quantization is usually understood as the process of mapping continuous wave form signals into a sequence of bits. In this context, the process of quantization is also referred to as lossy source coding, because the transition from continuous source signals to the discrete domain naturally entails a loss of signal quality/accuracy.

To quantify this loss, rate distortion theory [7, 86] defines a distortion measure to assess the fidelity of the quantized signal as compared to the source. Based on this concept, rate distortion theory characterizes the asymptotic tradeoff between the lowest possible rate that a signal can be quantized with while still maintaining a prescribed accuracy in terms of the distortion measure. For classic applications, the choice of the distortion measure often follows from the application in a straightforward manner: E.g., in audio coding, we wish that the signal amplitude of the quantized signal matches the source amplitude as closely as possible, giving rise to the MSE distortion measure.

The general formulation and strong theoretic foundation of rate distortion theory over the years led to it being applied to problems it was not originally developed for. That is, concepts of rate distortion theory have been applied to more abstract quantization problems occurring in machine learning applications such as speech compression, language processing, bio informatics, neural coding etc., cf. [99]. For those application, we face the problem that while rate distortion theory requires the definition of a distortion measure, the "correct" choice of distortion measure is virtually impossible to make, i.e., coming up with an adequate distortion measure that characterizes the relevant features of the source signals for those problems is a difficult task by itself.

In communications, a set of scenarios where the applicability of classic distortion measures such as MSE is limited are those concerned with received signal quantization. To illustrate the problem, consider a communication system transmitting iid data symbols x_i over a memoryless channel $p(y|x)$, producing outputs y_i. We now wish to quantize the outputs y_i to finite or lower resolution samples $\mathsf{z}_i = Q(\mathsf{y}_i)$ for further processing. The question arises on how we should assess the fidelity of the quantization, i.e., what distortion measure to chose. Clearly, we are interested in recovering the data x_i and not so much in accurately reproducing the received symbols y_i. Hence, choosing the quantization scheme in a way that the MSE $\mathbb{E}\left[\|\mathsf{y}_i - \mathsf{z}_i\|^2\right]$ is minimized does not necessarily bring us closer to the goal of recovering x_i. Towards this end, a better fidelity criterion is the overall capacity of the channel including the quantizer, that is, we want to maximize the mutual information $I(\mathsf{x};\mathsf{z})$. We refer to this approach as *information-optimal quantization* of a random variable y_i with respect to some *relevance variable* x_i, where z_i is the quantizer output. There are various instances of such quantization problems, e.g., channel output quantization [110, 120], quantization of decoder input LLRs [23, 109] or quantizing LLR messages within iterative message passing decoders [60, 67, 68, 81], to name some of them.

In what follows, we will first present a brief introduction to lossy source coding, followed by an examination of several variants of information-optimal quantization problems.

2.1.1 Fundamentals of Lossy Source Coding

In this section, we give a brief review of some aspects of rate distortion theory, as many concepts of information-optimal quantization are extensions and variations of thereof. The material presented here largely follows [21, Section 10.2].

We consider quantization of a discrete, iid source with distribution $p(y)$, emitting a sequence $\mathsf{y}_1, \mathsf{y}_2, \ldots, \mathsf{y}_n$, compactly denoted as $\mathsf{y}^n \in \mathcal{Y}^n$. Realizations y^n of this sequence are encoded to discrete integer messages m by means of an *encoding function*

$$f_n : \mathcal{Y}^n \to \{1, 2, \ldots, 2^{nR}\}, \tag{2.1}$$

and reconstructed from those integers to sequences $z^n \in \mathcal{Z}^n$ by a *decoding function*

$$g_n : \{1, 2, \ldots, 2^{nR}\} \to \mathcal{Z}^n, \tag{2.2}$$

where R is the *code rate* in bits per source symbol. The set of n-tupels $g(1)$, $g(2)$, …, $g(2^{nR})$ constitutes the *codebook*, and $f^{-1}(1)$, $f^{-1}(2)$, …, $f^{-1}(2^{nR})$ are the associated *assignment regions*. The distortion associated with a source sequence and the corresponding codeword is defined as

$$\rho_n(y^n, x^n) \triangleq \frac{1}{n} \sum_{i=1}^{n} \rho(y_i, z_i), \tag{2.3}$$

where ρ is the bounded *single letter distortion measure*

$$\rho : \mathcal{Y} \times \mathcal{Z} \to \mathbb{R}_+ \setminus \infty. \tag{2.4}$$

The *average distortion of the code* is then given by

$$D = \mathbb{E}\Big[\rho_n\Big(\mathrm{y}^n, g_n(f_n(\mathrm{y}^n))\Big)\Big], \tag{2.5}$$

with the expectation being taken over $\mathrm{y}^n \sim p(y^n) = \prod_{i=1}^{n} p(y_i)$.

Based on this setting, a pair (R, D) is said to be *achievable* if there exists a sequence of rate R codes (f_n, g_n) such that

$$\lim_{n\to\infty} \mathbb{E}\Big[\rho_n\Big(\mathrm{y}^n, g_n(f_n(\mathrm{y}^n))\Big)\Big] \leq D. \tag{2.6}$$

The *operational rate distortion function* then assigns to any distortion value D the lowest possible rate R such that (R, D) is still achievable[1]. In [86], Shannon derived a *coding theorem* stating that for this setup, the tremendously complex problem of finding the operational rate distortion function can be reduced to finding the *information rate distortion function*:

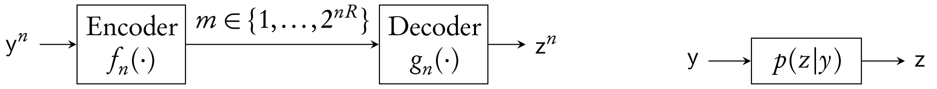
Definition 1. *For a given single letter distortion measure ρ and source distribution $p(y)$, the information rate distortion function $R : \mathbb{R}_+ \to \mathbb{R}_+$ is defined as*

$$R(D) \triangleq \min_{p(z|y)} I(\mathrm{y}; \mathrm{z}) \qquad \textit{such that} \qquad \mathbb{E}[\rho(\mathrm{y}, \mathrm{z})] \leq D. \tag{2.7}$$

Theorem 1. *The operational rate distortion function for an iid source* y *with distribution $p(y)$ and bounded distortion measure $\rho(y, z)$ is equal to the associated information rate distortion function.*

Due to Theorem 1, both operational and information rate distortion function are equivalent and are simply referred to by the same term in what follows. The rate distortion function characterizes the ultimate tradeoff between compression rate of the code and the accuracy of

[1] In this section, we mainly focus on the rate distortion function $R(D)$. All statements can be equivalently formulated using its inverse, the distortion rate function $D(R)$.



(a) Operational Model **(b)** Information theoretic Model

Figure 2.1: Rate distortion coding setup (a) and information-theoretic simplification (b)

the reconstructed sequences z^n, if infinitely long and complex codes were to be used. Both the operational setup and the simplified, information-theoretic setup are depicted in Figure 2.1.

2.1.2 Information Bottleneck Method

The IBM has first been introduced in [99] as a way to eliminate the distortion measure of rate distortion theory in favour of an information-theoretic relevance criterion. In its original formulation, the method assumes a given joint distribution $p(x,y)$ over the discrete probability space $\mathcal{X} \times \mathcal{Y}$, where x is the hidden *relevance variable* and y is the *observation variable* (also referred to as quantizer input in what follows). The IBM then seeks to find a probabilistic quantizer $p(z|y)$ that preserves as much information on the relevance variable as possible while at the same time striving for a maximally compact representation $z \in \mathcal{Z}$. Due to the restriction that the quantizer is independent of the relevance itself, the triple $x - y - z$ forms a Markov chain.

Figuratively speaking, the information that y provides on x is squeezed through a bottleneck formed by the compact representation z. Formally, this tradeoff is characterized by the variational problem

$$\min_{p(z|y)} I(y;z) - \beta I(x;z). \tag{2.8}$$

In this context, the quantity $I(x;z)$ is referred to as *relevant information*, while $I(y;z)$ is referred to as the *(compression) rate*. The Lagrange parameter $\beta > 0$ trades off rate against relevant information. For $\beta \to \infty$, the focus lies entirely on preserving relevant information, while for small β, minimizing the rate becomes increasingly important. Due to the data processing inequality, we have

$$I(y;z) \geq I(x;z) \quad \Longrightarrow \quad I(y;z) - \beta I(x;z) \geq (1-\beta) I(x;z), \tag{2.9}$$

an thus,

$$\beta \leq 1 \quad \Longrightarrow \quad \text{(2.8) is minimized by setting } I(x;z) = I(y;z) = 0, \tag{2.10}$$

i.e., the solution to (2.8) degenerates.

Unfortunately, an explicit solution to (2.8) can not be given; however, it is shown in [99, Theorem 4], that the optimal $p(z|y)$ minimizing (2.8) satisfies

$$p(z|y) = \frac{p(z)\exp\big(-\beta d(y,z)\big)}{\sum_{z' \in \mathcal{Z}} p(z')\exp\big(-\beta d(y,z')\big)}, \tag{2.11}$$

where

$$d(y,z) \triangleq \sum_{x\in\mathcal{X}} p(x|y)\log\frac{p(x|y)}{p(x|z)} = D\big(p(x|y)\|p(x|z)\big) \tag{2.12}$$

is the *Kullback-Leibler divergence (KLD)* [21, Section 2.3] between the two probability mass functions (pmfs) $p(x|y)$ and $p(x|z)$. The pmfs $p(z)$ and $p(x|z)$ occurring in (2.11) can be obtained from the known joint pmf $p(x,y)$ and the quantizer mapping $p(z|y)$ via

$$p(z) = \sum_{y\in\mathcal{Y}} p(z|y)p(y), \tag{2.13}$$

$$p(x|z) = \frac{1}{p(z)}\sum_{y\in\mathcal{Y}} p(x,y)p(z|y). \tag{2.14}$$

The stationarity condition (2.11) gives rise to an iterative algorithm based on alternating projections [21, Section 10.8]: Starting with some initial quantizer $p^{(0)}(z|y)$, we compute $p^{(0)}(z)$ and $p^{(0)}(x|z)$ according to (2.13) and (2.14). The quantizers for subsequent iterations then follow by plugging $p^{(0)}(z)$ and $p^{(0)}(x|z)$ into (2.11) and proceeding recursively. This algorithm is known as the iterative information bottleneck (IB) algorithm [99] and is summarized in Algorithm 1. It is similar to the famous *Blahut-Arimoto* algorithm [2, 9] that is used to solve the optimization problem (2.7) for finding the rate distortion function of a discrete iid source. However, contrary to (2.7), (2.8) is not convex, so Algorithm 1 can not be expected to converge to the global minimum. Therefore, it must be run with several different initializations to obtain a good solution.

Algorithm 1 Iterative IB algorithm

```
Input: $\mathcal{X}$, $\mathcal{Y}$, $\mathcal{Z}$, $p(x,y)$, $\beta > 0$, $p^{(0)}(z|y)$, $I$,$\epsilon$
$i \leftarrow 1$
$\epsilon^{(0)} \leftarrow \infty$
while $i \le I$ and $\epsilon^{(i-1)} \ge \epsilon$ do
    $p^{(i)}(z) \leftarrow \sum_{y\in\mathcal{Y}} p^{(i-1)}(z|y)p(y)$
    $p^{(i)}(x|z) \leftarrow \frac{1}{p^{(i)}(z)}\sum_{y\in\mathcal{Y}} p(x,y)p^{(i-1)}(z|y)$
    $d^{(i)}(y,z) \leftarrow D\big(p(x|y)\|p^{(i)}(x|z)\big)$
    $p^{(i+1)}(z|y) \leftarrow \frac{p^{(i)}(z)\exp\big(-\beta d^{(i)}(y,z)\big)}{\sum_{z'\in\mathcal{Z}} p^{(i)}(z)\exp\big(-\beta d^{(i)}(y,z')\big)}$
    $\epsilon^{(i)} \leftarrow \sum_{y\in\mathcal{Y}} p(y)D\big(p^{(i+1)}(z|y)\|p^{(i)}(z|y)\big)$
end while
Output: Locally optimal mapping $p(z|y)$
```

2.1.3 Rate Information Coding and Relation to Rate Distortion Theory

In this section, we will study the relation between rate distortion optimal compression and rate information compression as characterized by the IBM. In contrast to the IBM, the notion of relevance is determined by the choice of the distortion measure in rate distortion theory.

In what follows, we will show that the IB variational problem (2.8) takes the form of a rate distortion tradeoff with an information-theoretic distortion measure. To this end, we note that due to the Markovity of the IB variables $\mathsf{x} - \mathsf{y} - \mathsf{z}$, we have

$$I(\mathsf{x};\mathsf{z}) = I(\mathsf{x};\mathsf{y}) - I(\mathsf{x};\mathsf{y}|\mathsf{z}) = I(\mathsf{x};\mathsf{y}) - \mathbb{E}[d(\mathsf{y},\mathsf{z})], \tag{2.15}$$

with $I(\mathsf{x};\mathsf{y})$ being independent of $p(z|y)$ and d as defined in (2.12). Consequently, the optimization problem (2.8) can be equivalently reformulated as

$$\min_{p(z|y)} I(\mathsf{y};\mathsf{z}) + \beta \mathbb{E}[d(\mathsf{y},\mathsf{z})]. \tag{2.16}$$

The constrained minimization problem in (2.7) takes the same form as (2.16), if the constraint $\mathbb{E}[\rho(\mathsf{y},\mathsf{z})] \leq D$ is taken into account by introducing a Lagrange parameter $\beta > 0$,

$$\min_{p(z|y)} I(\mathsf{y};\mathsf{z}) + \beta \mathbb{E}[\rho(\mathsf{y},\mathsf{z})]. \tag{2.17}$$

Thus, the IB approach can be viewed as a rate distortion tradeoff with the information-theoretic distortion measure (2.12). There is however a significant difference between conventional distortion measures (2.4) and the IB distortion measure (2.12) in that d depends on the conditional $p(z|y)$, thereby rendering the equivalent problems (2.8) and (2.16) non convex. Furthermore, d is also dependent on the joint distribution $p(x,y)$. In this way, the joint distribution $p(x,y)$ determines the distortion measure and in turn what features of y are deemed relevant about the hidden variable x.

There is another fundamental difference between rate distortion theory and the IBM when it comes to the motivation for solving the information-theoretic tradeoff problems (2.17) and (2.8). As explained in Section 2.1.1, rate distortion theory is interested in solving (2.17) due to its relevance for the operational coding problem, cf. Figure 2.1a. Because of this, rate distortion theory focuses on the optimal tradeoff and not so much on the argument $p(z|y)$ achieving this tradeoff. This is in stark contrast to the IBM, which was originally introduced in [99] without any operational meaning and seeks to utilize the optimal $p(z|y)$ as a (soft) quantizer. A coding theorem for the IB approach similar to Theorem 1 was only introduced later in [34]. Equivalent theorems have been proven in other contexts in [1, 20, 43, 107]. In [34], the fidelity criterion (2.5) of a code (f_n, g_n) is replaced by the *average relevant information* of the code,

$$I^{\mathrm{RI}} \triangleq \frac{1}{n}\sum_{i=1}^{n} I\Big((\mathsf{x}^n)_i; g_n\big(f_n(\mathsf{y}^n)\big)_i\Big). \tag{2.18}$$

Note that the component-wise definition of (2.18) can be replaced by the mutual information rate between the involved sequences,

$$I^{\mathrm{RI}} \triangleq \frac{1}{n} I\Big(\mathsf{x}^n; g_n\big(f_n(\mathsf{y}^n)\big)\Big), \tag{2.19}$$

and the resulting achievable regions can be shown to be identical, cf. [20, 73].

The operational setting for rate information coding is as follows: We consider an iid sequence of pairs $(\mathsf{x}_i, \mathsf{y}_i) \sim p(x,y)$, $i = 1, \ldots, n$. Out of the resulting two sequences x^n and y^n, only y^n is observed and subject to encoding and decoding. A *rate information code* (f_n, g_n) then produces a compressed version $\mathsf{z}^n = g_n\big(f_n(\mathsf{y}^n)\big)$ such that z^n is maximally informative about the hidden sequence x^n, corresponding to a large average relevant information (2.18). In (2.18), the mutual information of the ith element is calculated with respect to the distribution defined by the code for this coordinate,

$$p_i(y,z) = \sum_{y^n \in \mathcal{Y}^n :\, (y^n)_i = y \,\wedge\, g_n(f_n(y^n))_i = z} p(y^n), \tag{2.20}$$

$$p_i(x,z) = \sum_{y \in \mathcal{Y}} p(x,y) p_i(y,z). \tag{2.21}$$

In this context, a rate information pair (R, I) is said to be achievable, if there exists a sequence of rate R codes (f_n, g_n) such that for some n, $I^{\mathrm{RI}} \geq I$. The *rate information function* $R(I)$ then assigns to any I the minimal possible R such that (R, I) is still achievable. Similarly, the *information rate function* $I(R)$ assigns to any code rate R the maximum average relevant information I such that (R, I) is still achievable. In [20, 34] it is shown that those complex operational tradeoffs are given by the following information-theoretic functions:

Definition 2. *Let* x—y—z *be a Markov chain. The rate information function* $R\colon \big[0, I(\mathsf{x};\mathsf{y})\big] \to \mathbb{R}_+$ *is defined as*

$$R(I) \triangleq \min_{p(\mathsf{z}|\mathsf{y})} I(\mathsf{y};\mathsf{z}) \quad \textit{subject to} \quad I(\mathsf{x};\mathsf{z}) \geq I; \tag{2.22}$$

the information rate function $I\colon \mathbb{R}_+ \to \big[0, I(\mathsf{x};\mathsf{y})\big]$ *is defined as*

$$I(R) \triangleq \max_{p(\mathsf{z}|\mathsf{y})} I(\mathsf{x};\mathsf{z}) \quad \textit{subject to} \quad I(\mathsf{y};\mathsf{z}) \leq R. \tag{2.23}$$

Theorem 2. *Let $R(I)$ be convex and continuous. The operational rate information function for an iid sequence of pairs $(\mathsf{x}_i, \mathsf{y}_i)$ with distribution $p(x,y)$ over the finite set $\mathcal{X} \times \mathcal{Y}$ is equal to the rate information function* (2.22)*. Similarly, the operational information rate function is equal to the information rate function* (2.23)

The Markovity in Definition 2 is due to the fact that x is not observed by the decoder and matches the presumption of the IBM. Lastly, we want to remark that information-optimal source coding is related to cost capacity [22, 105] and coding with side information [13, 118], cf. [34].

2.1.4 Deterministic Information-Optimal Quantizers

The solution to the IBM variational problem (2.8) by means of the Algorithm 1 generally produces a soft partition of $\mathcal{Y}$, i.e., any $y \in \mathcal{Y}$ is assigned to an output z with probability $p(z|y)$. Soft partitions can be useful for some applications, e.g., when clustering documents into categories, the same document might fall into multiple categories. For this problem $\mathcal{Y}$ is the set of documents, $\mathcal{Z}$ are the categories and $p(z|y)$ can be regarded as a weighted assignment of documents to categories. Thus, while probabilistic $p(z|y)$ certainly make sense as soft partitions in some cases, their applicability as quantizers is limited. Furthermore, if $p(z|y)$ is used directly as a quantizer, the rate minimization characterized by the minimization of $I(\mathsf{y};\mathsf{z})$ is not meaningful in an operational context, because the rate of a quantizer is determined by the size of the discrete output alphabet $\mathcal{Z}$. Due to these considerations, one is often more interested in solving

$$\max_{p(z|y)} I(\mathsf{x};\mathsf{z}), \tag{2.24}$$

rather than (2.8). Assuming again that $\mathsf{x}-\mathsf{y}-\mathsf{z}$ is a Markov chain, (2.24) is a convex maximization problem over a convex domain, which by sign inversion can be equivalently formulated as concave minimization problem over a convex domain. It is shown in [47, Theorem 1.1] that for this type of problems, the optimum is attained at an extreme point of the convex domain. For the convex set of probability distributions, this means that the optimum of (2.24) is achieved by a *deterministic* assignment $p(z|y)$. Unfortunately, (2.24) is generally *NP-hard* and therefore no efficient algorithm exists for finding the global optimum. In what follows, we will discuss several instances of and solution strategies to (2.24).

IBM Hard Clustering Limit

First, note that (2.24) is equivalent to the IBM variational problem (2.8) for $\beta \to \infty$. We can thus employ the stationarity condition (2.11) for the limit

$$p(z|y) = \lim_{\beta\to\infty} \frac{p(z)\exp\big(-\beta d(y,z)\big)}{\sum_{z'\in\mathcal{Z}} p(z')\exp\big(-\beta d(y,z')\big)} \tag{2.25}$$

$$= \lim_{\beta\to\infty} \left(1 + \sum_{z'\in\mathcal{Z}\setminus z} \frac{p(z')}{p(z)} \exp\Big(-\beta\big(d(y,z') - d(y,z)\big)\Big)\right)^{-1} \tag{2.26}$$

$$= \begin{cases} 1, & z = \arg\min_{z'} d(y,z'), \\ 0, & \text{otherwise} \end{cases}. \tag{2.27}$$

Interestingly, Algorithm 1 then becomes an instance of the famous *LLoyd-Max* algorithm with the distributions $p(x|y)$ taking the role of the vectors to be quantized,

$$p(x|z) = \sum_{y\in\mathcal{Y}} p(x|y)p(y|z) = \frac{\sum_{\mathcal{Y}_z} p(y)p(x|y)}{\sum_{\mathcal{Y}_z} p(y)} \tag{2.28}$$

representing the *mass centroids* (cf., e.g., [26]) of the quantization regions $\mathcal{Y}_z = \{y \mid p(z|y) = 1\}$, and (2.12) being used to measure the distance between elements and centroids. Starting from an initial partition $\mathcal{Y}_z^{(0)}$, any iteration of the algorithm updates the centroids, followed by assigning the elements $p(x|y)$ to the output z with minimum centroid distance according to (2.12) and (2.27), thereby defining the partition for the next iteration, cf. Algorithm 2. This

Algorithm 2 LLoyd-Max infotmation theoretic clustering based on the IBM

```
Input: $\mathcal{X}$, $\mathcal{Y}$, $\mathcal{Z}$, $p(x,y)$, $\mathcal{Y}_z^{(0)}$, $I$,$\epsilon$
i ← 1
$\epsilon^{(0)} \leftarrow \infty$
while $i \leq I$ and $\epsilon^{(i-1)} \geq \epsilon$ do
    $p^{(i)}(x|z) \leftarrow \dfrac{\sum_{\mathcal{Y}_z^{(i)}} p(y)p(x|y)}{\sum_{\mathcal{Y}_z^{(i)}} p(y)}$
    $d^{(i)}(y,z) \leftarrow D\big(p(x|y) \| p^{(i)}(x|z)\big)$
    $\mathcal{Y}_z^{(i+1)} \leftarrow \big\{y \mid d^{(i)}(y,z) \leq d^{(i)}(y,z') \quad \forall z' \in \mathcal{Z} \setminus z\big\}$
    $\epsilon^{(i)} \leftarrow \sum_{z \in \mathcal{Z}} p(z) D\big(p^{(i+1)}(x|z) \| p^{(i)}(x|z)\big)$
end while
Output: Locally optimal hard partition $\mathcal{Y}_z$
```

algorithm was originally published in [91] as *hard clustering limit* of the IBM; slight variations or reformulations thereof later appeared in [14, 57, 60]. Once again, as it was the case for Algorithm 1, Algorithm 2 converges to a locally, though not generally globally optimal solution and hence, running the algorithm for different initializations leads to the best results.

Binary Relevance $\mathcal{X}$

The optimization problem (2.24) can be simplified for the case of binary relevance $\mathcal{X} = \{-1, +1\}$. For this case, based on [11, Theorem 1], it can be shown that there exists an optimum, convex partition $\mathcal{L}_z^\star$ on the set of LLRs

$$\mathcal{L} \triangleq \big\{\mathrm{L}(\mathrm{x}|y) \mid y \in \mathcal{Y}\big\}. \tag{2.29}$$

By Bayes law, the *a posteriori* LLR

$$\mathrm{L}(\mathrm{x}|y) \triangleq \log \frac{p(x=+1|y)}{p(x=-1|y)} = \underbrace{\log \frac{p(x=+1)}{p(x=-1)}}_{\mathrm{L}(\mathrm{x})} + \underbrace{\log \frac{p(y|x=+1)}{p(y|x=-1)}}_{\mathrm{L}(y|\mathrm{x})}. \tag{2.30}$$

is the sum of the *a priori* LLR L(x) that depends only on $p(x)$ and the *channel* LLR $\mathrm{L}(y|\mathrm{x})$ that only depends on $p(y|x)$. If not mentioned otherwise, the elements of $\mathcal{X}$ are assumed to be equally likely, i.e., $\mathrm{L}(\mathrm{x}) = 0$ and $\mathrm{L}(\mathrm{x}|y) = \mathrm{L}(\mathrm{x}|y) = \mathrm{L}(y)$, where we drop the explicit notation of the binary random variable x if it is clear from the context.

Thus, if the elements of $\mathcal{Y}$ are ordered in terms of their LLR[2],

$$\mathrm{L}(y_1) < \mathrm{L}(y_2) < \cdots < \mathrm{L}(y_{|\mathcal{Y}|}), \tag{2.31}$$

this guarantees the existence of a quantizer with contiguous quantization regions on $\mathcal{Y}$, i.e., $p(z|y)$ takes the form[3]

$$p(z|y) = \begin{cases} 1, & a_{z-1} < y \leq a_z, \\ 0, & \text{otherwise,} \end{cases} \tag{2.32}$$

cf. [58]. The strict ordering applies in (2.31) because elements that share the same LLR value can be merged into a single element without changing the mutual information between the observation and relevance $I(\mathrm{x};\mathrm{y})$. More precisely, if $y_1,\ldots, y_l$ are such that $\mathrm{L}(y_1) = \cdots = \mathrm{L}(y_l)$, then we merge these events into a new element y; thus

$$\mathrm{L}(y_1) = \cdots = \mathrm{L}(y_l) \quad \Rightarrow \quad y \triangleq \bigcup_{i=1}^{l} y_i \quad \Rightarrow \quad p(y|x) = \sum_{i=1}^{l} p(y_i|x). \tag{2.33}$$

One can easily verify that the LLR of the merged elements is preserved: $\mathrm{L}(y) = \mathrm{L}(y_1) = \cdots = \mathrm{L}(y_l)$.

For discrete $\mathcal{Y}$, the fact that there is an optimal quantizer with contiguous regions greatly reduces the search space for an optimal mapping and gives rise to the dynamic programming algorithm presented in [58]. The algorithm efficiently evaluates the objective function $I(\mathrm{x};\mathrm{z})$ for all contiguous mappings by noting that it can be written as

$$I(\mathrm{x};\mathrm{z}) = \sum_{z\in\mathcal{Z}} I(a_{z-1} \to a_z), \tag{2.34}$$

where

$$I(a_{z-1} \to a_z) = \sum_{x\in\mathcal{X}} p(x) \left(\sum_{y\in\{a_{z-1}+1,\ldots,a_z\}} p(y|x) \right) \log \frac{\sum\limits_{y\in\{a_{z-1}+1,\ldots,a_z\}} p(y|x)}{\sum\limits_{x'\in\mathcal{X}} p(x') \sum\limits_{y\in\{a_{z-1}+1,\ldots,a_z\}} p(y|x')} \tag{2.35}$$

denotes the *partial mutual information* of an output z with contiguous assignment region $\{a_{z-1}+1,\ldots,a_z\}$. Taking advantage of the fact that there are less than $|\mathcal{Y}|^2$ possible combinations of pairs (a_{z-1}, a_z), the algorithm precomputes all possible values $I(a_{z-1} \to a_z)$ and then finds the optimal set of boundaries

$$0 = a_1^\star < a_2^\star < \cdots < a_{|\mathcal{Z}|}^\star = |\mathcal{Z}| \tag{2.36}$$

[2] Note that $\mathrm{L}(\mathrm{x})$ is independent of y and thus, even for $\mathrm{L}(\mathrm{x}) \neq 0$, the ordering (2.31) does not depend on whether we base it on $\mathrm{L}(\mathrm{x}|y)$ or $\mathrm{L}(\mathrm{x}|y)$

[3] for the formulation of the algorithm, we assume that the sets $\mathcal{Y}$ and $\mathcal{Z}$ are integers, i.e., $\mathcal{Y} = \{1,\ldots,|\mathcal{Y}|\}$, $\mathcal{Z} = \{1,\ldots,|\mathcal{Y}|\}$ and without loss of generality, that the quantizer $\Phi : y \to z$ is monotonic in that $y' > y \Rightarrow \Phi(y') \geq \Phi(y)$.

by a state evolution type algorithm with traceback, cf. Algorithm 3. Contrary to Algorithm 2,

Algorithm 3 Optimal quantizer design for binary relevance [58]

```
Input: $|\mathcal{Y}|$, $|\mathcal{Z}|$, $p(x,y)$

# preparation
Order and merge inputs according to (2.31) and (2.33)
Compute $I(a' \to a)$ for all $a' < a$

# state recursion
$S_0(0) \leftarrow 0$
for $z \in \{1,\dots,|\mathcal{Z}|\}$ do
    for $a \in \{z,\dots,z+|\mathcal{Y}|-|\mathcal{Z}|\}$ do
        $S_z(a) \leftarrow \max_{a' \in \{z-1,\dots,a-1\}} \big(S_{z-1}(a') + I(a' \to a)\big)$
        $h_z(a) \leftarrow \operatorname{arg\,max}_{a' \in \{z-1,\dots,a-1\}} \big(S_{z-1}(a') + I(a' \to a)\big)$
    end for
end for

# state traceback
$a^\star_{|\mathcal{Z}|} \leftarrow |\mathcal{Z}|$
for $z \in \{|\mathcal{Z}|-1,\dots,1\}$ do
    $a^\star_z \leftarrow h_{z+1}(a^\star_{z+1})$
end for
$a^\star_0 \leftarrow 0$
Output: Optimal quantizer boundaries $\boldsymbol{a}^\star$, maximum mutual information $I^\star(\mathsf{x},\mathsf{z}) = S_{|\mathcal{Z}|}\big(|\mathcal{Y}|\big)$
```

Algorithm 3 always finds a global optimum to (2.24). The complexity of Algorithm 3 is dominated by the calculation of the partial mutual information (2.35) and is of the order $O\big(|\mathcal{Y}|^3\big)$.

We also want to note that the existence of an optimal convex partition $\mathcal{L}^\star_z$ can be exploited to simplify Algorithm 2 for the case of binary relevance, cf. [60]. Furthermore, this also implies that there exists a contiguous quantizers for continuous $\mathcal{Y}$, giving rise to algorithms that try to optimize the quantizer boundaries $\boldsymbol{a}$ over a subset of $\mathbb{R}^{|\mathcal{Z}|-1}$, cf. [109].

2.1.5 Gaussian Information Bottleneck

In this section, we shift our focus back to the problem of solving (2.8), but this time we consider the case where observation and relevance are jointly Gaussian random vectors $\mathbf{x} \in \mathbb{R}^{m_x}$ and $\mathbf{y} \in \mathbb{R}^{m_y}$ with zero mean and covariance matrix $C_{\mathbf{x},\mathbf{y}}$. This setup is referred to as Gaussian information bottleneck (GIB) [16] and is formally characterized by the variational problem

$$\min_{p(z|y)} I(\mathbf{y};\mathbf{z}) - \beta I(\mathbf{x};\mathbf{z}). \tag{2.37}$$

It is shown in [35] that the optimum is attained by a random vector $\mathbf{z}$ that is jointly Gaussian with $\mathbf{y}$. Consequently, the optimal $\mathbf{z}$ can be expressed as a linear transformation

$$\mathbf{z} = A\mathbf{y} + \boldsymbol{\xi}, \tag{2.38}$$

with $\boldsymbol{\xi} \sim \mathcal{N}(\mathbf{0}, C_{\boldsymbol{\xi}})$ and independent of $\mathbf{y}$. Hence, the optimization over $p(z|y)$ in (2.37) can be equivalently formulated as an optimization over the projection matrix A and covariance matrix $C_{\boldsymbol{\xi}}$,

$$\min_{A, C_{\boldsymbol{\xi}}} I(\mathbf{y}; A\mathbf{y} + \boldsymbol{\xi}) - \beta I(\mathbf{x}; A\mathbf{y} + \boldsymbol{\xi}). \tag{2.39}$$

A solution to (2.39) is presented in [16, Theorem 3.1]. There, the authors show that for a prescribed β, the optimum is attained by

$$C_{\boldsymbol{\xi}} = \boldsymbol{I}, \qquad A = \mathrm{diag}\{\alpha_i\}_{i=1}^{n} V^{\mathrm{T}}, \qquad \alpha_i = \sqrt{\frac{[\beta(1-\lambda_i)-1]^{+}}{\lambda_i v_i^{\mathrm{T}} C_{\mathbf{y}} v_i}}, \tag{2.40}$$

where the eigendecomposition

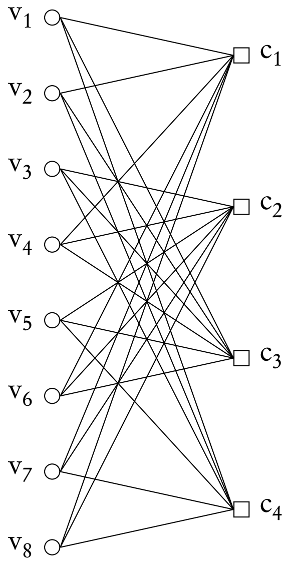
$$C_{\mathbf{y}|\mathbf{x}} C_{\mathbf{y}}^{-1} = V \mathrm{diag}\{\lambda_i\}_{i=1}^{m_y} V^{\mathrm{T}} \quad \text{with} \quad V = [v_1, \ldots, v_{m_y}], \tag{2.41}$$

is used. Here, v_i^{T} and $0 \leq \lambda_i \leq 1$ are the left eigenvectors and corresponding eigenvalues of the matrix $C_{\mathbf{y}|\mathbf{x}} C_{\mathbf{y}}^{-1}$, where we assume that the eigenvalues are in ascending order $\lambda_1 \leq \cdots \leq \lambda_{m_y}$, i.e., the smallest eigenvalues are the first ones to contribute to (2.40). The number of non-zero rows of the projection matrix A is determined by the value of the Lagrange parameter β. For $\beta < \frac{1}{1-\lambda_1}$ the solution degenerates to $A = \mathbf{0}$. As β increases, it transitions through several critical points, at each of which another non-zero row is added to A. In this sense, β can be interpreted as "continuous rank" of the projection, because A changes continuously in β. Once $\beta > \beta_{c,k}$ with

$$\beta_{c,k} \triangleq \frac{1}{1-\lambda_k}, \tag{2.42}$$

the kth mode with eigenvalue λ_k becomes active and starts contributing to the relevant information $I(\mathbf{x};\mathbf{z})$. Denoting the number of active modes as $j(\beta) = \max\{k : \beta_{c,k} \leq \beta\}$ and using Definition 2 for the Markov chain $\mathbf{x} - \mathbf{y} - \mathbf{z}$, an explicit relation between $R = I(\mathbf{y};\mathbf{z})$ and $I(R) = I(\mathbf{x};\mathbf{z})$ is given by

$$I(R) = R - \frac{j}{2} \log_2 \left(\prod_{i=1}^{j} (1-\lambda_i)^{\frac{1}{j}} + e^{\frac{2R}{j}} \prod_{i=1}^{j} \lambda_i^{\frac{1}{j}} \right). \tag{2.43}$$



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

Figure 2.2: Example of a lenght 8, rate 1/2, regular (3,6) LDPC code with $N = 8$ and $M = 4$

2.2 LDPC Codes

2.2.1 General Definitions and Message Passing Decoding

Low-density parity-check (LDPC) codes are a class of linear block codes[4] with sparse parity check matrix $\boldsymbol{H} \in \mathrm{GF}(2)^{M \times N}$, i.e., $\boldsymbol{H}\boldsymbol{c} = \mathbf{0}$ for all codewords $\boldsymbol{c}$ of the LDPC code.

Regular LDPC codes have originally been introduced by Gallager in [32] and were largely forgotten until their rediscovery in [66]. For a regular (d_v, d_c) LDPC code, each row of the parity check matrix contains exactly d_c non-zero entries and each column contains exactly d_v non-zero entries. Thus, "sparse" refers to the fact that d_c and d_v are fixed and independent of the codeword length, and hence, the number of non-zero elements in $\boldsymbol{H}$ is *linear* in block length N, while for fixed rate, the total number of elements is proportional to N^2. The $M \times N$ parity check matrix of a LDPC code can be represented by a sparse, bipartite factor graph consisting of M check nodes (CNs) and N variable nodes (VNs), where each non-zero entry corresponds to an edge in the graph, cf. Figure 2.2 for an example. Consequently, d_v and d_c are referred to as VN and CN *degree*, respectively. We denote the *ensemble* of all regular, length N (d_v, d_c) LDPC code graphs as $\mathcal{C}^N(d_v, d_c)$.

Later on [62, 63, 77], it was discovered that the performance of LDPC codes can be improved by allowing for an *irregular* degree structure, where the gains come mostly from an irregularity in the VN degrees. Intuitively, irregular codes improve upon regular codes of the same rate due to on a mixture of low degree VNs, whose purpose it is to keep the rate small and high degree VNs which provide redundancy and good error correction performance. To

[4] In this thesis, we restrict our attention to binary codes, although LDPC codes can be defined over arbitrary finite fields [24]. We represent codewords in $\mathrm{GF}(2)^N$ as either $\boldsymbol{c} \in \{0,1\}^N$ or $\boldsymbol{x} = 1 - 2\boldsymbol{c} \in \{+1,-1\}^N$, with identity elements of addition $\oplus$ 0 and $+1$, respectively, and identity elements of multiplication $\odot$ 1 and -1, respectively.

characterize irregular LDPC codes, we use *edge perspective degree distributions* in polynomial form [63],

$$\lambda(z) = \sum_{i\in\mathcal{D}_\mathrm{v}} \lambda_i z^{i-1}, \qquad \rho(z) = \sum_{j\in\mathcal{D}_\mathrm{c}} \rho_j z^{j-1}. \tag{2.44}$$

Here λ_i, $i \in \mathcal{D}_\mathrm{v}$, is the fraction of edges incident to a degree i VN, and ρ_j, $j \in \mathcal{D}_\mathrm{c}$, is the fraction of edges incident to a degree j CN, with $\mathcal{D}_\mathrm{v}$ and $\mathcal{D}_\mathrm{c}$ denoting the VN and CN degree sets, respectively. For a regular (d_v, d_c) LDPC code, all VNs have degree d_v and all CNs have degree d_c, i.e., $\mathcal{D}_\mathrm{v} = \{d_v\}$ and $\mathcal{D}_\mathrm{c} = \{d_c\}$. The edge perspective degree distributions λ and ρ are related to their node perspective counterparts Λ and P via

$$\Lambda_i = \frac{\lambda_i/i}{\sum_{i\in\mathcal{D}_\mathrm{v}} \lambda_i/i}, \qquad P_j = \frac{\rho_j/j}{\sum_{j\in\mathcal{D}_\mathrm{c}} \rho_j/j}, \tag{2.45}$$

$$\lambda_i = \frac{i\Lambda_i}{\sum_{i\in\mathcal{D}_\mathrm{v}} i\Lambda_i}, \qquad \rho_j = \frac{jP_j}{\sum_{j\in\mathcal{D}_\mathrm{c}} jP_j}. \tag{2.46}$$

The number of edges incident to VNs and CNs must be the same,

$$N \sum_{i\in\mathcal{D}_\mathrm{v}} i\Lambda_i = M \sum_{j\in\mathcal{D}_\mathrm{c}} jP_j \quad \Longleftrightarrow \tag{2.47}$$

$$\frac{N}{\sum_{i\in\mathcal{D}_\mathrm{v}} \lambda_i/i} = \frac{M}{\sum_{j\in\mathcal{D}_\mathrm{c}} \rho_j/j}, \tag{2.48}$$

and therefore, the code rate can be expressed solely in terms of the degree distributions,

$$R = 1 - \frac{M}{N} = 1 - \frac{\sum_{i\in\mathcal{D}_\mathrm{v}} i\Lambda_i}{\sum_{j\in\mathcal{D}_\mathrm{c}} jP_j} = 1 - \frac{\sum_{j\in\mathcal{D}_\mathrm{v}} \rho_j/j}{\sum_{i\in\mathcal{D}_\mathrm{c}} \lambda_i/i}. \tag{2.49}$$

Similar to the regular ensemble $\mathcal{C}^N(d_v, d_c)$, we denote the irregular ensemble with degree distributions λ and ρ as $\mathcal{C}^N(\lambda, \rho)$. Lastly, note that the concept of irregular LDPC codes can be even further generalized to so called *multi-edge type* LDPC codes [75, 76, 79].

Message Passing Decoding

LDPC codes are traditionally decoded using message passing algorithms, where messages are exchanged across the edges between VNs and CNs over the course of several decoding iterations. The sparsity of $\boldsymbol{H}$ is paramount for the efficient decoding of LDPC codes, since it ensures that the number of edges is proportional to N. Consequently, for a total of $\mathscr{L}$ message passing iterations, decoding complexity is proportional to $O(N\mathscr{L})$ which means that LDPC codes can be efficiently decoded in linear time.

In what follows, we describe a generic message passing decoding scheme. In iteration ℓ, the message from VN n (with degree i) to an adjacent CN m are computed via a mapping

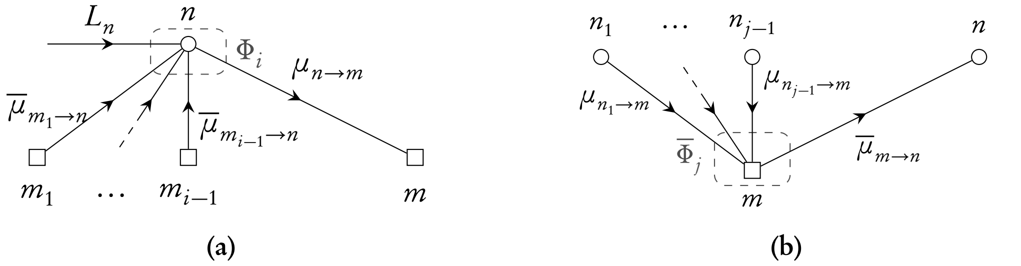


Figure 2.3: VN update (a) and CN update (b) for $\mathcal{N}(n) = \{m, m_1, \ldots, m_{i-1}\}$ and $\mathcal{N}(m) = \{n, n_1, \ldots, n_{j-1}\}$

$\Phi_i^{(\ell)} : \mathcal{L} \times \overline{\mathcal{M}}_\ell^{i-1} \to \mathcal{M}_\ell$ according to

$$\mu_{n\to m} = \Phi_i^{(\ell)}\big(L_n, \overline{\mu}_{n\backslash m}\big). \tag{2.50}$$

Here, $L_n \in \mathcal{L}$ denotes the channel LLR at VN n and $\overline{\mu}_{n\backslash m} \in \overline{\mathcal{M}}_\ell^{i-1}$ is the vector of incoming messages from all adjacent CNs except m ($\overline{\mathcal{M}}_\ell$, $\mathcal{M}_\ell$, and $\mathcal{L}$ are the message and LLR alphabets). For $\ell = 0$, the VN update only depends on the channel message, $\Phi_v^{(0)} : \mathcal{L} \to \mathcal{M}_0$, as there is no extrinsic information from CNs available yet. Similarly, the message in iteration ℓ from CN m (with degree j) to an adjacent VN n is obtained via the mapping $\overline{\Phi}_j^{(\ell)} : \mathcal{M}_\ell^{j-1} \to \overline{\mathcal{M}}_{\ell+1}$ defined by

$$\overline{\mu}_{m\to n} = \overline{\Phi}_j^{(\ell)}(\mu_{m\backslash n}), \tag{2.51}$$

where $\mu_{m\backslash n} \in \mathcal{M}_\ell^{j-1}$ is the vector of messages incoming from the adjacent VNs except for VN n. Figure 2.3 illustrates the message updates in the factor graph.

After a total of $\mathscr{L}$ iterations, the final estimate for a code bit x_n at a VN of degree i is computed based on the incoming CN messages and the channel LLR with a mapping Ψ_i : $\mathcal{L} \times \overline{\mathcal{M}}_{\mathscr{L}}^{i} \to \mathrm{GF}(2)$ defined by

$$\hat{x}_n = \Psi_i(L_n, \overline{\mu}_n). \tag{2.52}$$

Belief Propagation

The *belief propagation (BP) algorithm* is a special instance of a message passing decoding algorithm where the messages exchanged across the edges of the graph represent probabilities for the value of the involved code bit, conditioned on the channel output and information received from neighbouring nodes. Assuming the factor graph of the code does not contain any cycles, BP produces the maximum a posteriori probability (MAP) estimate,

$$\hat{x}_i = \underset{x_i \in \mathrm{GF}(2)}{\arg\max}\, p(x_i|y) = \underset{x_i \in \mathrm{GF}(2)}{\arg\max} \sum_{x \in \mathcal{C}:(x)_i = x_i} p(x|y), \tag{2.53}$$

where the a posteriori probabilitys (APPs) $p(x_i|y)$ in (2.53) are obtained iteratively. For binary codes, it is convenient to express probabilities as LLRs, cf. (2.30). This way, the MAP detector

(2.53) can be based on the sign of the APP LLR

$$\hat{x}_i = \operatorname{sign}\big(\mathrm{L}(\mathrm{x}_i|\,\boldsymbol{y})\big) = \operatorname{sign}\left(\log\frac{p(x_i=+1|\boldsymbol{y})}{p(x_i=-1|\boldsymbol{y})}\right), \tag{2.54}$$

and the messages exchanged between nodes are real numbers. The posterior probabilities can be recovered from the LLRs as

$$p(x|y) = \frac{e^{x\,\mathrm{L}(\mathrm{x}|y)/2}}{e^{\mathrm{L}(\mathrm{x}|y)/2} + e^{-\mathrm{L}(\mathrm{x}|y)/2}}. \tag{2.55}$$

Assuming that the code bits x_i are a priori equally likely, $\mathrm{L}(\mathrm{x}_i|\,\boldsymbol{y})$ can be expressed as

$$\mathrm{L}(\mathrm{x}_i|\,\boldsymbol{y}) = \mathrm{L}(y_i|\mathrm{x}_i) + \mathrm{L}(\boldsymbol{y}_{\sim i}|\mathrm{x}_i), \tag{2.56}$$

where $\mathrm{L}(y_i|\mathrm{x}_i)$ is the channel LLR, cf. (2.30), and $\mathrm{L}(\boldsymbol{y}_{\sim i}|\mathrm{x}_i)$ is referred to as the *extrinsic* LLR. Thus, for BP, the channel LLRs $L_n \in \mathcal{L}$ are obtained from the channel output y_n via

$$L_n = \log\frac{p(y_n|x_n=+1)}{p(y_n|x_n=-1)}. \tag{2.57}$$

The extrinsic LLR $\mathrm{L}(\boldsymbol{y}_{\sim i}|\mathrm{x}_i)$ on the other hand incorporates all the information on x_i that is obtained from other observations $\boldsymbol{y}_{\sim i}$ and code constraints on $\boldsymbol{x}$. BP message updates represent extrinsic LLRs, meaning that they do not include the incoming message of the recipient node for the calculation of the update. For a length-i repetition code, we have

$$\mathrm{L}(\boldsymbol{y}_{\sim m}|\mathrm{x}_m) = \sum_{\substack{m'=1\\ m'\neq m}}^{i} \mathrm{L}(y_{m'}|\mathrm{x}_{m'}), \tag{2.58}$$

and thus for the VN updates,

$$\Phi_i^{\mathrm{BP}}(L,\overline{\mu}) = L + \sum_{m=1}^{i-1}\overline{\mu}_m, \tag{2.59}$$

For a length-j single parity check code on the other hand,

$$\mathrm{L}(\boldsymbol{y}_{\sim n}|\mathrm{x}_n) = 2\,\mathrm{atanh}\left(\prod_{\substack{n'=1\\ n'\neq n}}^{j}\tanh\left(\tfrac{\mathrm{L}(y_{n'}|\mathrm{x}_{n'})}{2}\right)\right), \tag{2.60}$$

hence,

$$\overline{\Phi}_j^{\mathrm{BP}}(\mu) = 2\,\mathrm{atanh}\left(\prod_{n=1}^{j-1}\tanh\left(\tfrac{\mu_n}{2}\right)\right). \tag{2.61}$$

Consequently, for the BP algorithm $\mathcal{M}_\ell = \overline{\mathcal{M}_\ell} = \mathbb{R}$ and the update rules are identical for all ℓ, i.e., $\Phi_i^{(\ell)} = \Phi_i^{\mathrm{BP}}$ and $\overline{\Phi}_j^{(\ell)} = \overline{\Phi}_j^{\mathrm{BP}}$.

The bit decision Ψ_i is based on the sign of the APP LLRs, cf. (2.54),

$$\Psi_i^{\mathrm{MS/BP}}(L,\overline{\mu}) = \mathrm{sign}\Big(L + \sum_{m=1}^{i} \overline{\mu}_m\Big). \tag{2.62}$$

Since (2.61) is costly to evaluate, it is sometimes replaced by

$$\overline{\Phi}_j^{\mathrm{MS}}(\mu) = \prod_{n=1}^{j-1} \mathrm{sign}(\mu_n) \cdot \min_n |\mu_n|, \tag{2.63}$$

yielding the min-sum (MS) algorithm.

2.2.2 Asymptotic Behaviour

One of the major theoretical results on LDPC codes is that *almost all* long codes from the ensemble $\mathcal{C}^N$ behave alike and their performance can be analyzed by considering the cycle-free case.

Theorem 3 (Concentration around cycle-free case, [62, 78]). *Over the probability space of all graphs $\mathcal{C}^N(d_v,d_c)$ and channel realizations let z be the number of incorrect messages among all Nd_v VN-to-CN messages passed at BP iteration ℓ. Let $P_e^{(\ell)}$[5] be the expected fraction of incorrect messages passed along an edge with a tree-like directed neighborhood of depth at least 2ℓ at the ℓth iteration. Then, there exist positive constants $\beta = \beta(d_v,d_c,\ell)$ and $\gamma = \gamma(d_v,d_c,\ell)$ such that the following holds:*

Concentration around expected value: *For any $\epsilon > 0$ we have*

$$\mathrm{P}\big\{|\mathsf{z} - \mathbb{E}[\mathsf{z}]| > Nd_v\epsilon/2\big\} \leq 2e^{-\beta\epsilon^2 N}. \tag{2.64}$$

Convergence to cycle-free case: *For any $\epsilon > 0$ and $N > \frac{2\gamma}{\epsilon}$ we have*

$$|\mathbb{E}[\mathsf{z}] - Nd_v P_e^{(\ell)}| < Nd_v\epsilon/2. \tag{2.65}$$

Concentration around cycle-free case: *For any $\epsilon > 0$ we have*

$$\mathrm{P}\Big\{|\mathsf{z} - Nd_v P_e^{(\ell)}| > Nd_v\epsilon/2\Big\} \leq 2e^{-\beta\epsilon^2 N}. \tag{2.66}$$

Theorem 3 implies that for increasingly larger codeword length N, the number of message errors of *any codeword from any code in the ensemble $\mathcal{C}^N(d_v,d_c)$ transmitted over any particular channel realization* is close to $Nd_v P_e^{(\ell)}$ with increasingly higher probability. Consequently, by calculating $P_e^{(\ell)}$ for various channel families, we can characterize the performance of almost all codes from $\mathcal{C}^N(d_v,d_c)$ for those channels.

[5] We will see shortly that $P_e^{(\ell)}$ is independent of the transmit codeword $\boldsymbol{x}$.

Moreover, it is shown in [78, Theorem 1], that for BP decoding, we have $P_e^{(\ell)'} > P_e^{(\ell)}$ for a channel $p(y'|x) = p(y'|y)p(y|x)$ that is physically degraded with respect to $p(y|x)$. For channels that can be characterized by a single parameter, this concept implies a monotonicity between the channel parameter α and the fraction of incorrect messages $P_e^{(\ell)}$. E.g., for the binary input additive white Gaussian noise (BI-AWGN) channel

$$p(y|x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(y-x)^2}{2\sigma^2}}, \tag{2.67}$$

we identify $\alpha = \sigma$, where as for the binary symmetric channel,

$$p(y|x) = (1-\epsilon)\Delta_x + \epsilon\Delta_{-x}, \tag{2.68}$$

or the erasure channel,

$$p(y|x) = \epsilon\Delta_0 + (1-\epsilon)\Delta_\infty, \tag{2.69}$$

$\alpha = \epsilon$. One can show that any channel with parameter $\alpha' > \alpha$ is physically degraded with respect to the original channel with parameter α. Consequently, for all those channels, if $\alpha' > \alpha$ it follows that $P_e^{(\ell)'} > P_e^{(\ell)}$ for all ℓ. This monotonicity gives rise to a thresholding phenomenon: For a given family of channels, we set out to find the highest parameter α such that for $\ell \to \infty$, $P_e^{(\ell)} \to 0$, i.e.,

$$\alpha^* = \sup_\alpha \{\alpha : \lim_{\ell\to\infty} P_e^{(\ell)}(\alpha) = 0\}, \tag{2.70}$$

is the *threshold* of the respective channel. Conversely, for $\alpha > \alpha^*$, there is a positive constant $\gamma(\alpha)$ such that $P_e^{(\ell)} > \gamma(\alpha)$ for all ℓ. Hence, for long codes, we can clearly separate the region where reliable transmission is possible from where it is not. Note that all of the above also holds for irregular ensembles $\mathcal{C}^N(\lambda, \rho)$, with appropriately adjusted constants in Theorem 3.

2.2.3 Density Evolution and Symmetry Conditions

In this section, we discuss how to obtain $P_e^{(\ell)}$, the error probability of the cycle-free case introduced in Theorem 3. The key observation here is that at iteration ℓ, if the code has no cycles of length shorter than 2ℓ, then under the assumption of iid channels, all messages incident to VNs and CNs are independent and thus, can be characterized by a single message density. Furthermore, under the following, fairly general symmetry constraints, the message densities are symmetric and the probability that a message is in error is independent of the transmitted codeword $\boldsymbol{x}$, i.e., $P_e^{(\ell)}(\boldsymbol{x}) = P_e^{(\ell)}$.

Assumption 1 (Symmetric message passing, [78]).

1. ***Channel symmetry:*** *Let $p_{\mathsf{L}|\mathsf{x}}(L|x)$ denote the conditional distribution of the iid channel, mapping input bits $x \in \mathrm{GF}(2)$ to channel output LLRs $L \in \mathcal{L}$. We require channel output symmetry*

$$p_{\mathsf{L}|\mathsf{x}}(L|x) = p_{\mathsf{L}|\mathsf{x}}(-L|-x). \tag{2.71}$$

2. ***Check node symmetry:*** *Signs factor out of CN maps, i.e., for each iteration ℓ, CN degree j and any ± 1 sequence $(b_1,\dots,b_{j-1})$,*

$$\overline{\Phi}_j^{(\ell)}(b_1\mu_1,\dots,b_{j-1}\mu_{j-1}) = \left(\prod_{n=1}^{j-1} b_n\right)\overline{\Phi}_j^{(\ell)}(\mu_1,\dots,\mu_{j-1}). \tag{2.72}$$

3. ***Variable node symmetry:*** *The VN updates must respect the sign inversion property*

$$\Phi_i^{(\ell)}(-L,-\overline{\mu}_1,\dots,-\overline{\mu}_{i-1}) = -\Phi_i^{(\ell)}(L,\overline{\mu}_1,\dots,\overline{\mu}_{i-1}) \tag{2.73}$$

for all $\ell > 0$ and degrees i. For the initial VN update, we assume

$$\Phi_i^{(0)}(-L) = -\Phi_i^{(0)}(L). \tag{2.74}$$

Lemma 1 (Independent, symmetric messages [78])**.** *Let G be a cycle-free factor graph of a parity check code and assume the codeword $\boldsymbol{x}$ has been transmitted. If the symmetry conditions of Assumption 1 are fulfilled, then for any decoding iteration ℓ, both VN-to-CN and CN-to-VN messages are conditionally iid and their densities factor according to*

$$p_{\mu|x}^{(\ell)}(\boldsymbol{\mu}|\boldsymbol{x}) = \prod_{\mathrm{e}\in\mathcal{E}} p_{\mu|x}^{(\ell)}(\mu_{\mathrm{e}}|x_{n(\mathrm{e})}), \tag{2.75}$$

where $\mathcal{E}$ is the set of edges of G and $n(\mathrm{e})$ denotes the index of the VN attached to edge e. Furthermore, both conditional VN-to-CN and CN-to-VN message distributions are symmetric according to

$$p_{\mu|x}^{(\ell)}(\mu_{\mathrm{e}}|-x_{n(\mathrm{e})}) = p_{\mu|x}^{(\ell)}(-\mu_{\mathrm{e}}|x_{n(\mathrm{e})}) \tag{2.76}$$

for all $\mathrm{e}\in\mathcal{E}$ and iterations ℓ.

Since by Lemma 1, the probability of a message error is independent of the transmitted codeword, it is commonly assumed that the all-one codeword $\boldsymbol{x} = +\boldsymbol{1}$ is transmitted. Henceforth, we write $p_{\mu|x}^{(\ell)}(\mu) = p_{\mu|x}^{(\ell)}(\mu|+1) =$ for VN-to-CN message densities and $p_{\overline{\mu}|x}^{(\ell)}(\overline{\mu}) = p_{\overline{\mu}|x}^{(\ell)}(\overline{\mu}|+1)$ for CN-to-VN message densities at iteration ℓ, implicitly assuming $\boldsymbol{x} = +\boldsymbol{1}$. The error probability can then be expressed as

$$P_e^{(\ell)} = \int_{-\infty}^{0^-} p_{\mu|x}^{(\ell)}(\mu)\mathrm{d}\mu + \frac{1}{2}\int_{0^-}^{0^+} p_{\mu|x}^{(\ell)}(\mu)\mathrm{d}\mu. \tag{2.77}$$

For most decoding algorithms, the densities $p_{\mu|x}^{(\ell)}(\mu)$ and $p_{\overline{\mu}|x}^{(\ell)}(\overline{\mu})$ can be computed recursively. In what follows, we derive the recursion for the BP decoding algorithm. Since the output of a VN update (2.59) is the sum of independent random variables, we have

$$p_{\mu|x}^{(\ell)} = p_{\mathsf{L}|x} \otimes p_{\overline{\mu}|x}^{(\ell)^{\otimes i-1}}, \tag{2.78}$$

where for a density p,

$$p^{\otimes i} = \overbrace{p \otimes p \otimes \cdots \otimes p}^{i-1 \text{ convolutions}} \tag{2.79}$$

denotes the $i$fold self convolution. For the CN update, note that (2.61) can be rewritten as a sum over $\mathrm{GF}(2) \times [0, \infty)$,

$$\gamma^{-1}\left(\sum_{n=1}^{j-1} \gamma(\mu_n)\right), \tag{2.80}$$

with transformation $\gamma : \mathbb{R} \to \mathrm{GF}(2) \times [0, \infty)$,

$$\gamma(x) \triangleq \left(\operatorname{sign} x, -\log\tanh\frac{|x|}{2}\right), \tag{2.81}$$

and corresponding inverse $\gamma^{-1} : \mathrm{GF}(2) \times [0, \infty) \to \mathbb{R}$. Thus, due to the independence of incoming messages,

$$p^{(\ell+1)}_{\overline{\mathsf{\mu}}|\mathsf{x}} = \Gamma^{-1}\left(\Gamma\left(p^{(\ell)}_{\mathsf{\mu}|\mathsf{x}}\right)^{\otimes j-1}\right), \tag{2.82}$$

where Γ denotes the transformation of densities corresponding to (2.81), cf. [77].

Taking into account irregular codes, we can combine (2.78) and (2.82) and obtain a recursion on VN-to-CN densities,

$$p^{(\ell)}_{\mathsf{\mu}|\mathsf{x}} = p_{\mathsf{L}|\mathsf{x}} \otimes \lambda\left(\Gamma^{-1}\left(\rho\left(\Gamma(p^{(\ell-1)}_{\mathsf{\mu}|\mathsf{x}})\right)\right)\right), \tag{2.83}$$

with degree polynomials λ and ρ according to (2.44) and[6]

$$a(p) = \sum_{i=0}^{n} a_i p^{\otimes i} \tag{2.84}$$

for a polynomial $a(x) = \sum_{i=0}^{n} a_i x^i$ applied to a density p. The recursion (2.83) can be shown to always converge to a fixed point and allows for an efficient evaluation based on Fourier transforms of densities. Suitable discretization of densities and various techniques to implement (2.83) efficiently have been proposed in [18] and [79, Appendix B]. Using (2.70), (2.77) and (2.83), we can determine the thresholds for various channels. Besides exact density evolution, approximative methods such as Gaussian message passing [19] and EXIT charts (cf. Section 2.3) can be used to determine thresholds with low computational complexity.

Finally, another symmetry condition is defined in [77, Definition 1] regarding the message densities. A density $p(\mu)$ is *LLR-consistent* iff

$$p(\mu) = e^{\mu} p(-\mu), \qquad \forall \mu \in \mathbb{R}. \tag{2.85}$$

When applied to symmetric conditional densities, i.e., $p_{\mathsf{\mu}|\mathsf{x}}(\mu) = p_{\mathsf{\mu}|\mathsf{x}}(\mu|+1) = p_{\mathsf{\mu}|\mathsf{x}}(-\mu|-1)$

[6] We define $p^{\otimes 0} = \Delta_0$.

with equal a priori probability, this property links the argument of the density to the corresponding LLR,

$$\mu = \mathrm{L}(\mathsf{x}|\mu) = \log \frac{p_{\mu|\mathsf{x}}(\mu|+1)}{p_{\mu|\mathsf{x}}(\mu|-1)}. \tag{2.86}$$

It can be shown that the recursion (2.83) preserves LLR-consistency. Hence, for BP decoding, the message densities are consistent according to (2.86) for all iterations.

2.3 EXIT Analysis

Extrinsic information transfer (EXIT) charts have originally been introduced in [97] as a tool to analyze the convergence behaviour of iteratively decoded BICM systems. Consider a component of an iterative decoder receiving iid input messages $\overline{\mu}$ with conditional density $p_{\overline{\mu}|\mathsf{x}}$ and producing iid output messages μ with conditional density $p_{\mu|\mathsf{x}}$. We then refer to $I_A = I(\overline{\mu}, \mathsf{x})$ as the a priori information of the component and to $I_E = I(\mu, \mathsf{x})$ as the extrinsic information of the component. The functional relationship $I_E(I_A)$ is referred to as the EXIT function of the component.

2.3.1 Gaussian a priori Information

A common assumption is to model the a priori messages as Gaussian random variables. It can be shown that for Gaussian LLRs, i.e., $\overline{\mu}|\mathsf{x} \sim \mathcal{N}(\mu_A, \sigma_A^2)$, it follows from (2.86) that $\mu_A = \sigma_A^2/2$ and thus,

$$I_A(\sigma_A) = I(\overline{\mu}, \mathsf{x}) = 1 - \frac{1}{\sqrt{2\pi\sigma_A^2}} \int_{-\infty}^{\infty} \log_2(1+e^{-\zeta}) \exp\left(-\frac{(\zeta - \sigma_A^2/2)^2}{2\sigma_A^2} \right) \mathrm{d}\zeta \triangleq J(\sigma_A). \tag{2.87}$$

For efficient implementations, the function $J(\sigma_A)$ as well as its inverse $J^{-1}(I_A)$ can be approximated as described in the appendix of [98] or can be based on a table of precomputed values and suitable interpolation. The corresponding extrinsic information $I_E(I_A)$ for a component under Gaussian input messages can be determined by Monte Carlo simulations, i.e., we generate Gaussian samples $\overline{\mu}_i|\mathsf{x} \sim N(\sigma_A^2/2, \sigma_A^2)$ where $\sigma_A = J^{-1}(I_A)$, feed them to the component and obtain iid output samples $\mu_i|\mathsf{x}$ from which we estimate $p_{\mu|\mathsf{x}}$ and compute I_E. This technique can be applied to obtain EXIT functions even for systems where the output density can not be determined analytically from the input density. E.g., in Figure 2.4, we plot the EXIT functions of an 8-PAM soft demapper and a recursive systematic convolutional code of rate 1/2 [97]. As illustrated in Figure 2.5, the demapper produces extrinsic soft information μ_i based on the channel LLRs L_i and the a priori information $\overline{\mu}_i$. The messages μ_i are in turn incident to the decoder, i.e.,

$$I_{E1} = I_{A2}. \tag{2.88}$$

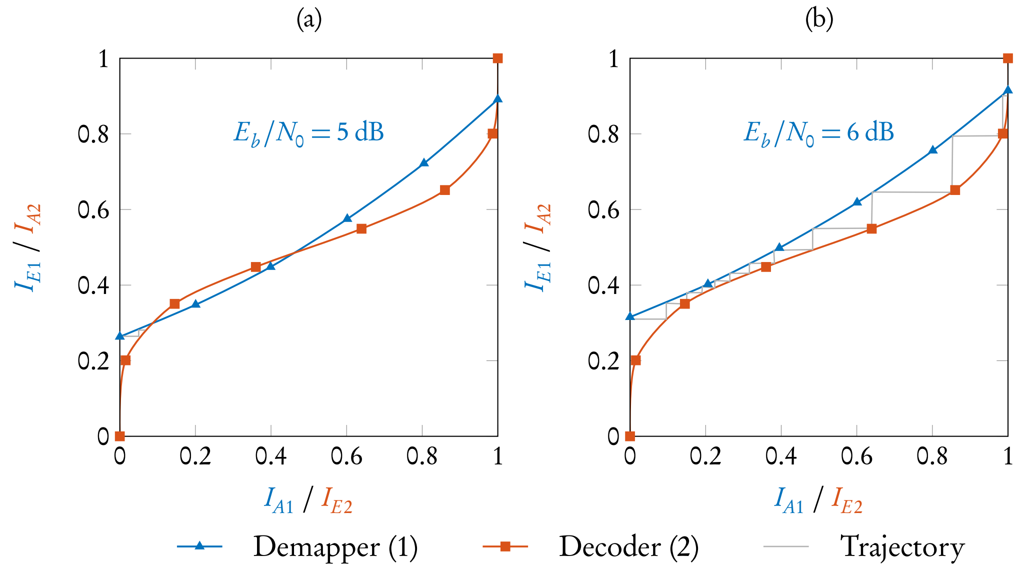


Figure 2.4: Example of an EXIT chart with corresponding decoding trajectory [97]

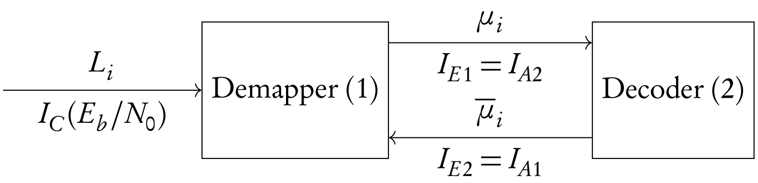


Figure 2.5: Iterative decoding and demodulation block diagram

The decoder then produces the extrinsic messages $\overline{\mu}_i$ which are fed back to the demapper, thus

$$I_{E2} = I_{A1}. \tag{2.89}$$

The relations (2.88) and (2.89) are represented in Figure 2.4 by switching the axis when plotting one of the two component EXIT functions. Starting from $I_{A1} = 0$, the demapper produces output messages with $I_{E1} = I_C(N_0/E_b) = I(\mathsf{L};\mathsf{x})$ which is then fed to the decoder, following the staircase trajectory between the two EXIT functions. Note that the EXIT function of the demapper is parametrized by the channel quality as it depends on the channel output LLRs L_i. In Figure 2.4a, we can see that the decoding procedure gets stuck after several iterations due to the intersection of the two EXIT curves for $E_b/N_0 = 5$ dB. For higher signal to noise ratio (SNR), $E_b/N_0 = 6$ dB, a gap opens between the two EXIT functions and the iterative decoding procedure can converge towards $I_{E2} \to 1$, cf. Figure 2.4b.

2.3.2 EXIT Charts of LDPC Node Updates

For simple systems such as the repetition and single parity check component codes of an LDPC code, the EXIT functions can be determined analytically if we resort to approximations. Specifically, simulation results show that the messages emitted from VNs follow a Gaussian distribu-

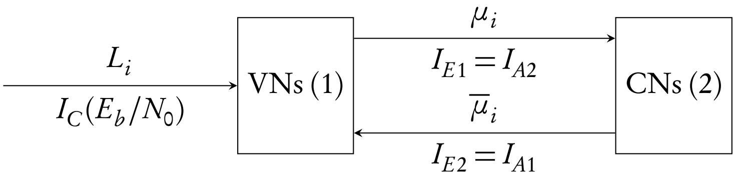


Figure 2.6: LDPC decoding block diagram

tion very closely [19]. Therefore, we can employ (2.87) to determine $\sigma^2_{\overline{\mathsf{\mu}}|\mathsf{x}} = J^{-1}(I_{A1})$ and due to (2.59), the output variance is given by $\sigma^2_{\mathsf{\mu}|\mathsf{x}} = \sigma^2_{\mathsf{L}|\mathsf{x}} + (i-1)\sigma^2_{\overline{\mathsf{\mu}}|\mathsf{x}}$, where $\sigma^2_{\mathsf{L}|\mathsf{x}} = J^{-1}\big(I_C\big)$ with channel capacity $I_C = I(\mathsf{x};\mathsf{L})$. Thus, for the EXIT function of a degree i VN

$$I_{E1}(I_{A1}, i, I_C) = J\Big(\sqrt{(i-1)\big(J^{-1}(I_{A1})\big)^2 + \big(J^{-1}(I_C)\big)^2}\Big). \tag{2.90}$$

For CNs, it is shown in [3] that for the erasure channel, the EXIT function $I_{E,\mathrm{SPC}}(I_A, d)$ of a single parity check code of degree d and the EXIT function $I_{E,\mathrm{REP}}(I_A, d)$ of a degree d repetition code are related as

$$I_{E,\mathrm{SPC}}(I_A, d) = 1 - I_{E,\mathrm{REP}}(1 - I_A, d). \tag{2.91}$$

While (2.91) is not exact for other channels such as the BI-AWGN channel and the binary symmetric channel, (2.91) is still very accurate for those cases [88, 98]. Hence, we can obtain the EXIT function of a degree j CN from (2.90) and (2.91),

$$I_{E2}(I_{A2}, j) = 1 - J\big(\sqrt{j-1}\, J^{-1}(1 - I_{A2})\big). \tag{2.92}$$

For irregular codes, the EXIT function of code mixtures can be obtained by averaging the EXIT functions for a fixed degree using edge perspective degree distributions [102],

$$I_{E1}(I_{A1}, \lambda, I_C) = \sum_{i \in \mathcal{D}_\mathrm{v}} \lambda_i I_{E1}(I_{A1}, i, I_C), \tag{2.93}$$

$$I_{E2}(I_{A2}, \rho) = \sum_{j \in \mathcal{D}_\mathrm{c}} \rho_j I_{E2}(I_{A2}, j). \tag{2.94}$$

In Figure 2.7, we show an EXIT chart of an LDPC Ensemble with $\lambda(z) = .251z + .309z^2 + .001z^3 + .439z^9$ and $\rho(z) = .636z^6 + .364z^7$ over an BI-AWGN channel with $\sigma = .93$. Conducting a bisection threshold search (cf. (2.70)) using (2.77) and (2.83) yields $\sigma^\star = .9558$ [77]. A similar threshold can be obtained using EXIT charts. First, note that using (2.88) to (2.90)

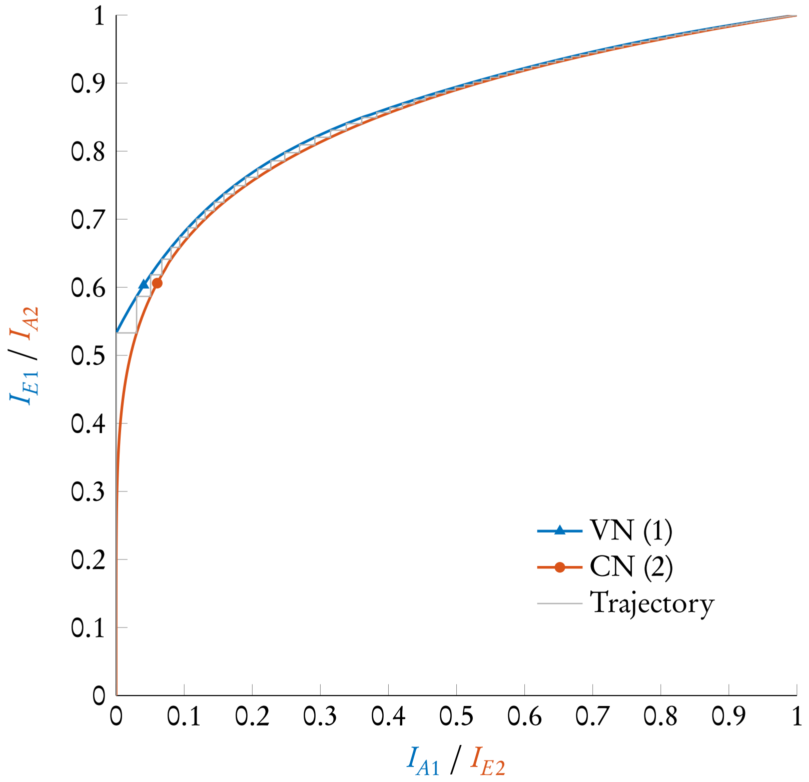


Figure 2.7: EXIT chart of an LDPC ensemble

and (2.92) to (2.94), we can derive a recursion on the extrinsic VN information

$$I_{E1}^{(\ell)}(I_C) = \sum_{i\in\mathcal{D}_\mathrm{v}} \lambda_i J\left(\sqrt{(i-1)\left(J^{-1}\left(1-\sum_{j\in\mathcal{D}_\mathrm{c}} \rho_j J\left(\sqrt{j-1}\,J^{-1}(1-I_{E1}^{(\ell-1)})\right)\right)\right)^2 + \left(J^{-1}(I_C)\right)^2}\right). \tag{2.95}$$

From (2.57) and (2.67), it follows that for a BI-AWGN channel, $L_i = \frac{2y_i}{\sigma^2}$ and thus $\left(J^{-1}(I_C)\right)^2 = \sigma^2_{\mathsf{L}|\mathsf{x}} = \frac{4}{\sigma^2}$, which allows us to express (2.95) as function of the channel noise standard deviation $I_{E1}^{(\ell)}(\sigma)$. Similar to (2.70), we can then exploit (2.95) to define

$$\sigma^*_{\mathrm{EXIT}} = \sup\{\,\sigma : \lim_{\ell\to\infty} I_{E1}^{(\ell)}(\sigma) = 1\} \tag{2.96}$$

as the threshold obtained using the EXIT method. Indeed, we obtain $\sigma^*_{\mathrm{EXIT}} = .9560$, which is fairly close to the true threshold.

Chapter 3

Rate Information Coding for Gaussian Signals

In Section 2.1.5, the Gaussian information bottleneck (GIB) was introduced as the analytic solution to the IB rate information tradeoff problems (2.22) and (2.23) for jointly Gaussian relevance and observation. The major result there was that the optimal mapping for this setup is given by the linear projection (2.38), where the dimensionality of the projection increases continuously for increasing Lagrange parameter β. From a source coding perspective, the optimal projection characterizes the ultimate tradeoff between compression and accuracy, however, the projection itself can not be used as a quantizer. Unlike for the discrete case (cf. Section 2.1.4), this even holds true as we let $\beta \to \infty$.[1]

Thus, the question remains how information-optimal quantizers can be designed for continuous observations and relevance. In this chapter, we provide a possible answer to this question. For that, we establish a relationship between the GIB and rate distortion theory that gives rise to an information-optimal quantizer design based on linear filtering and MSE-optimal quantization, i.e., we show that existing MSE-optimal quantizer designs can be used for good information rate performance given that the input to the MSE-optimal quantizer undergoes suitable linear preprocessing. The remainder of this chapter is structured as follows.

In Section 3.1, we analyze the GIB in a communications setting, i.e., we frame relevance and observation into a setting similar to a linear, additive noise communications channel. This reformulation is possible without loss of generality and helps to establish a link between the GIB and Wiener filtering [108].

In Section 3.2, we proceed by showing that the optimal information rate tradeoff can be achieved via the rate distortion optimal tradeoff of a linearly filtered version of the observation. We identify the square root of the Wiener filter as the information-optimal filter and investigate the effects on information rate as well rate distortion performance of different filtering approaches. Furthermore, we provide an extension of our results to stationary random

[1] To see this, divide (2.40) by $\sqrt{\beta}$ and let $\beta \to \infty$. The transformed projection has the same mutual information and becomes a noiseless bijection, so that $I(\mathbf{y};\mathbf{z}) = \infty$ and $I(\mathbf{x};\mathbf{z}) = I(\mathbf{x};\mathbf{y})$.

processes.

Section 3.3 then focuses on the operational relevance of the tradeoff problems we discussed in previous sections. We present a source coding theorem for jointly Gaussian relevance and discuss a potential extension to stationary ergodic processes. Furthermore, we investigate the applicability of the information rate tradeoff for finding the capacity of a channel with a quantizer at its output. Lastly, we discuss consequences of the relations we derived in Section 3.2 for the design of information-optimal source codes. We show that the results of Section 3.2 imply the existence information-optimal codes derived from MSE-optimal codes. Simulation results illustrate that codes with good information rate performance can indeed be constructed by designing MSE-optimal codes for linearly filtered observations.

3.1 On the Relation between the GIB and Wiener Filtering

3.1.1 Linear Relation between Relevance and Observation

In this section, we provide a novel reformulation and interpretation of the GIB in terms of Wiener filtering [108]. For that, we assume that observation x and relevance y are related via the linear model

$$\mathsf{y} = H\mathsf{x} + \mathsf{n}. \tag{3.1}$$

Here, $H \in \mathbb{R}^{m_y \times m_x}$ is a deterministic matrix and n is a Gaussian random vector distributed according to $\mathcal{N}(0, C_{\mathsf{n}})$ and independent of both x and y. Our interest in (3.1) is rooted in communications, where H and n represent channel and additive Gaussian noise. In this context, the relevance vector x represents the transmitted data and the observation y is the channel output that is subject to compression. Due to [7, Theorem 4.5.5], any two zero-mean, jointly Gaussian random vectors x and y can be represented by the model (3.1) and hence, all results that are derived under the assumption of (3.1) also hold for the general case.

Recall that the GIB solution (2.40) is based on the eigendecomposition (2.41) of the matrix

$$\overline{W} \triangleq C_{\mathsf{y}|\mathsf{x}} C_{\mathsf{y}}^{-1} = I - C_{\mathsf{x},\mathsf{y}}^{\mathrm{T}} C_{\mathsf{x}}^{-1} C_{\mathsf{x},\mathsf{y}} C_{\mathsf{y}}^{-1}. \tag{3.2}$$

Under the model (3.1), the involved covariance matrices are given by

$$C_{\mathsf{y}} = H C_{\mathsf{x}} H^{\mathrm{T}} + C_{\mathsf{n}}, \tag{3.3}$$

$$C_{\mathsf{y}|\mathsf{x}} = C_{\mathsf{n}}, \tag{3.4}$$

and thus,

$$\overline{W} = C_{\mathsf{n}} \big(H C_{\mathsf{x}} H^{\mathrm{T}} + C_{\mathsf{n}} \big)^{-1}. \tag{3.5}$$

Interestingly, $\overline{W}$ is seen to be the Wiener filter for estimating n from y. Furthermore, $\overline{W} = I - W$, where

$$W = H C_{\mathsf{x}} H^{\mathrm{T}} \big(H C_{\mathsf{x}} H^{\mathrm{T}} + C_{\mathsf{n}} \big)^{-1} \tag{3.6}$$

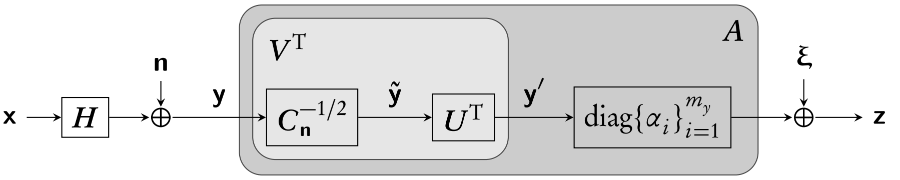


Figure 3.1: Linear model (3.1) and decomposition of the optimal projection $A = \mathrm{diag}\{\alpha_i\}_{i=1}^{m_y} U^{\mathrm{T}} C_{\mathsf{n}}^{-1/2}$

is the Wiener filter for estimating $H\mathsf{x}$ from y, i.e., it minimizes the MSE $\mathbb{E}\left[||W\mathsf{y} - H\mathsf{x}||^2\right]$. Note that both $\overline{W}$ and W share the same eigenbasis, i.e., they have the same left eigenvectors v_i^{T} and their eigenvalues are related by

$$\mu_i = 1 - \lambda_i, \qquad 0 \le \mu_i \le 1, \qquad i = 1, \dots, m_y. \tag{3.7}$$

Thus, the GIB solution (2.40) can be equivalently formulated in terms of the eigendecomposition of W. This formulation appears to be more natural given the Wiener filtering interpretation as well as the fact that the denominator for the expression of α_i in (2.40) can be reformulated according to

$$\lambda_i v_i^{\mathrm{T}} C_{\mathsf{y}} v_i = v_i^{\mathrm{T}} C_{\mathsf{y}|\mathsf{x}} C_{\mathsf{y}}^{-1} C_{\mathsf{y}} v_i = v_i^{\mathrm{T}} C_{\mathsf{y}|\mathsf{x}} v_i, \tag{3.8}$$

i.e., λ_i can be eliminated from the denominator and only enters complementary via $\mu_i = 1 - \lambda_i$ in the numerator.

3.1.2 GIB Solution and Rate Information Tradeoff under the Linear Model

In what follows, we derive the optimal GIB projection for the Markov chain $\mathsf{x} - \mathsf{y} - \mathsf{z}$ assuming that x and y are related via (3.1). For that, we decompose the optimal projection $A = V^{\mathrm{T}} \mathrm{diag}\{\alpha_i\}_{i=1}^{m_y}$ into three parts, as illustrated in Figure 3.1.

First, we will apply a noise whitening filter to the observation y. This transformation allows parametrization of the mutual information in terms of the SNR and later gives rise to expressions that are structurally similar to the capacity of a Gaussian vector channel. Moreover, the transformation allows for joint decorrelation of both noise and relevance terms. The whitened vector $\tilde{\mathsf{y}} = C_{\mathsf{n}}^{-1/2} \mathsf{y}$ has covariance $C_{\tilde{\mathsf{y}}} = S + I$ with the SNR matrix

$$S \triangleq C_{\mathsf{n}}^{-1/2} H C_{\mathsf{x}} H^{\mathrm{T}} C_{\mathsf{n}}^{-1/2}. \tag{3.9}$$

Since S is positive semi-definite and symmetric, it can be decomposed according to

$$S = U \Gamma U^{\mathrm{T}}, \tag{3.10}$$

where U consists of orthonormal eigenvectors and

$$\Gamma = \mathrm{diag}\{\gamma_i\}_{i=1}^{m_y}, \tag{3.11}$$

with non-negative *mode SNRs* $\gamma_i \geq 0$. The Wiener filters (3.5) and (3.6) transform to the whitened domain according to

$$\widetilde{\overline{W}} = C_{\mathsf{n}}^{-1/2} \overline{W} C_{\mathsf{n}}^{1/2} = (S+I)^{-1} = U\big(\Gamma+I\big)^{-1} U^{\mathrm{T}} = U \mathrm{diag}\{\lambda_i\}_{i=1}^{m_y} U^{\mathrm{T}}, \tag{3.12}$$

$$\widetilde{W} = C_{\mathsf{n}}^{-1/2} W C_{\mathsf{n}}^{1/2} = S(S+I)^{-1} = U\Gamma(\Gamma+I)^{-1} U^{\mathrm{T}} = U \mathrm{diag}\{\mu_i\}_{i=1}^{m_y} U^{\mathrm{T}}, \tag{3.13}$$

where

$$\lambda_i = \frac{1}{1+\gamma_i}, \qquad \mu_i = \frac{\gamma_i}{1+\gamma_i}. \tag{3.14}$$

Next, we apply a decorrelation transformation to the whitened observation. Using the eigendecomposition of the SNR matrix (3.10), the elements of

$$\mathsf{y}' = U^{\mathrm{T}} \tilde{\mathsf{y}} = U^{\mathrm{T}} C_{\mathsf{n}}^{-1/2} \mathsf{y} \tag{3.15}$$

can be seen to be uncorrelated with covariance matrices

$$C_{\mathsf{y}'} = \Gamma + I, \tag{3.16}$$

$$C_{\mathsf{y}'|\mathsf{x}} = I. \tag{3.17}$$

Lastly, we compute the optimal GIB solution for the information-optimal linear projection $A' = V'^{\mathrm{T}} \mathrm{diag}\{\alpha_i'\}_{i=1}^{m_y}$ from y' to z, i.e., we solve (2.39) for the Markov chain $\mathsf{x} - \mathsf{y}' - \mathsf{z}$. It follows from (2.41) and the covariance matrices (3.16) and (3.17) that $V' = I$. Hence, for the denominator in (2.40),

$$\mathrm{diag}\{\lambda_i\}_{i=1}^{m_y} V'^{\mathrm{T}} C_{\mathsf{y}'} V' = V'^{\mathrm{T}} C_{\mathsf{y}|\mathsf{x}'} V' = I \tag{3.18}$$

and thus,

$$A' = \mathrm{diag}\{\alpha_i'\}_{i=1}^{m_y}, \quad \alpha_i' = \sqrt{[\beta\mu_i - 1]^+}. \tag{3.19}$$

Note that since both whitening and decorrelation are invertible, they do not affect the mutual information. That is, we have $I(\mathsf{y};\mathsf{z}) = I(\mathsf{y}';\mathsf{z})$, which means that $I(\mathsf{y}';\mathsf{z})$ can be used to calculate the optimal rate information tradeoff for the original problem. Furthermore, it can be seen that the whitening and decorrelation operations constitute the matrix $V^{\mathrm{T}} = U^{\mathrm{T}} C_{\mathsf{n}}^{-1/2}$ for the information-optimal transformation of the original observation y to z. Similar to (3.18) for the whitened and decorrelated case, the denominator in the original domain reads

$$\mathrm{diag}\{\lambda_i\}_{i=1}^{m_y} V^{\mathrm{T}} C_{\mathsf{y}} V = V^{\mathrm{T}} C_{\mathsf{y}|\mathsf{x}} V = U^{\mathrm{T}} C_{\mathsf{n}}^{-1/2} C_{\mathsf{n}} C_{\mathsf{n}}^{-1/2} U = I. \tag{3.20}$$

Thus, the information-optimal projection of y is given by

$$A = \mathrm{diag}\{\alpha_i\}_{i=1}^{m_y} U^{\mathrm{T}} C_{\mathsf{n}}^{-1/2}, \quad \alpha_i = \alpha_i' = \sqrt{[\beta\mu_i - 1]^+}, \tag{3.21}$$

which proves the correctness of the decomposition introduced in Figure 3.1. We are now ready to formulate the following result.

Theorem 4. *Consider the Markov chain* x—y—z *under the model* (3.1)*. For this case, the optimum rate information tradeoff as introduced in Definition 2 is characterized by the parametric equations*

$$I(\beta) = \frac{1}{2}\sum_{i=1}^{m_y} \log_2^+\left(\frac{\beta-1}{\beta}(1+\gamma_i)\right), \tag{3.22}$$

$$R(\beta) = \frac{1}{2}\sum_{i=1}^{m_y} \log_2^+((\beta-1)\gamma_i), \tag{3.23}$$

where each choice of the parameter $\beta \in (1,\infty)$ *corresponds to a point on the rate information and information rate function.*

Proof: Due to the joint Gaussianity of all vectors and the invertibility of $\boldsymbol{U}^{\mathrm{T}}\boldsymbol{C}_{\mathsf{n}}^{-1/2}$ we have

$$I(\mathsf{y};\mathsf{z}) = I(\mathsf{y}';\mathsf{z}) = h(\mathsf{z}) - h(\mathsf{z}|\mathsf{y}') = \frac{1}{2}\log_2 \det \boldsymbol{C}_{\mathsf{z}}\boldsymbol{C}_{\mathsf{z}|\mathsf{y}'}^{-1}, \tag{3.24}$$

$$I(\mathsf{x};\mathsf{z}) = h(\mathsf{z}) - h(\mathsf{z}|\mathsf{x}) = \frac{1}{2}\log_2 \det \boldsymbol{C}_{\mathsf{z}}\boldsymbol{C}_{\mathsf{z}|\mathsf{x}}^{-1}, \tag{3.25}$$

where we used $h(\mathsf{x}) = \frac{1}{2}\log_2 \det(2\pi e \boldsymbol{C}_{\mathsf{x}})$ to denote the differential entropy of a multivariate Gaussian with covariance matrix $\boldsymbol{C}_{\mathsf{x}}$ [21]. The result follows by inserting into these expressions the covariance matrices

$$\boldsymbol{C}_{\mathsf{z}} = \boldsymbol{A}\boldsymbol{C}_{\mathsf{y}}\boldsymbol{A}^{\mathrm{T}} + \boldsymbol{C}_{\xi} = \mathrm{diag}\{\alpha_i\}_{i=1}^{m_y}\boldsymbol{V}^{\mathrm{T}}\boldsymbol{C}_{\mathsf{y}}\boldsymbol{V}\mathrm{diag}\{\alpha_i\}_{i=1}^{m_y} + \boldsymbol{I} \tag{3.26}$$

$$\overset{(3.20)}{=} \mathrm{diag}\{\alpha_i\}_{i=1}^{m_y}(\boldsymbol{\Gamma}+\boldsymbol{I})\mathrm{diag}\{\alpha_i\}_{i=1}^{m_y} + \boldsymbol{I} = \mathrm{diag}\left\{\alpha_i^2(\gamma_i+1)+1\right\}_{i=1}^{m_y}, \tag{3.27}$$

$$\boldsymbol{C}_{\mathsf{z}|\mathsf{y}'} = \boldsymbol{C}_{\xi} = \boldsymbol{I}, \tag{3.28}$$

$$\boldsymbol{C}_{\mathsf{z}|\mathsf{x}} = \boldsymbol{A}\boldsymbol{C}_{\mathsf{n}}\boldsymbol{A}^{\mathrm{T}} + \boldsymbol{C}_{\xi} = \mathrm{diag}\left\{\alpha_i^2+1\right\}_{i=1}^{m_y}. \tag{3.29}$$

Note that the requirement $\beta > 1$ eliminates the trivial solution (2.10). ■

3.1.3 Critical Rates and Optimal Rate Allocation

The parameter β originally introduced in the variational problem (2.37) restricts the active modes contributing to the information rate function $I(R)$. As it was the case for the original formulation of the GIB, β transitions trough a series of critical values

$$\beta_{c,k} = 1 + \frac{1}{\gamma_k}, \tag{3.30}$$

at each of which another mode with SNR γ_k becomes active. Here, (3.30) follows by setting the terms in (3.22) equal to zero, or equivalently, from (2.42) and (3.14).

We now set out to obtain an explicit relation $I(R)$ in the form of (2.43) in terms of mode SNRs and allocated rates. For that, we define the *critical rate* $R_{c,k}$ for mode k as the value R for which the mode becomes active, i.e., starts contributing to $I(R)$. Plugging (3.30) into the rate expression (3.23), we obtain the critical rates as

$$R_{c,k} \triangleq R(\beta_{c,k}) = \frac{1}{2}\sum_{i=1}^{m_y} \log_2^+ \frac{\gamma_k}{\gamma_i} = \frac{1}{2}\sum_{i=1}^{k} \log_2 \frac{\gamma_k}{\gamma_i}, \tag{3.31}$$

where we assume that the mode SNRs are in descending order. Furthermore, denoting the number of active modes as $j(R) = \max\{k : R_{c,k} \leq R\}$ we can use (3.23) to obtain

$$\beta = \frac{2^{2R/j}}{\bar{\gamma}_j} + 1, \tag{3.32}$$

with

$$\bar{\gamma}_j \triangleq \prod_{i=1}^{j} \gamma_i^{1/j} \tag{3.33}$$

denoting the geometric mean of the active modes. Using (3.32), we can express the rate allocated to mode i as

$$R_i(R) \triangleq \frac{1}{2}\log_2^+\big((\beta-1)\gamma_i\big) = \left[\frac{R}{j} + \frac{1}{2}\log_2 \frac{\gamma_i}{\bar{\gamma}_j}\right]^+. \tag{3.34}$$

Both (3.32) and (3.34) can then be used to eliminate the variational parameter from (3.22),

$$\begin{aligned} I(R) &= \frac{1}{2}\sum_{i=1}^{j} \log_2\left(\frac{2^{2R/j}}{2^{2R/j}+\bar{\gamma}_j}(1+\gamma_i)\right) \\ &= \frac{1}{2}\sum_{i=1}^{j} \log_2\left(\frac{1+\gamma_i}{1+2^{-2R_i(R)}\gamma_i}\right) = \sum_{i=1}^{j}\Big(C(\gamma_i) - C\big(2^{-2R_i(R)}\gamma_i\big)\Big). \end{aligned} \tag{3.35}$$

Here,

$$C(\gamma) \triangleq \frac{1}{2}\log_2(1+\gamma) \quad \text{and} \quad C(\boldsymbol{\gamma}) \triangleq \sum_{i=1}^{m_y} C(\gamma_i) \tag{3.36}$$

denote the capacity for a scalar and vector AWGN channel with SNR(s) γ and $\boldsymbol{\gamma}$, respectively. As β increases, more modes begin to contribute; the rate R increases, allowing for a higher accuracy of the compression and in turn higher relevant information I. From (3.34), we see that the rate is allocated linearly to the modes depending on their SNR. Modes with SNR higher than the geometric mean $\bar{\gamma}_j$ receive additional rate on top of the uniform default allocation $\frac{R}{j}$, whereas the rate allocated to modes with SNR smaller that $\bar{\gamma}_j$ is reduced below the uniform allocation.

Note that due to due to the Markov chain $\mathbf{x} - \mathbf{y} - \mathbf{z}$, we have $I(\mathbf{x};\mathbf{z}) \leq I(\mathbf{y};\mathbf{z})$ and $I(\mathbf{x};\mathbf{z}) \leq$

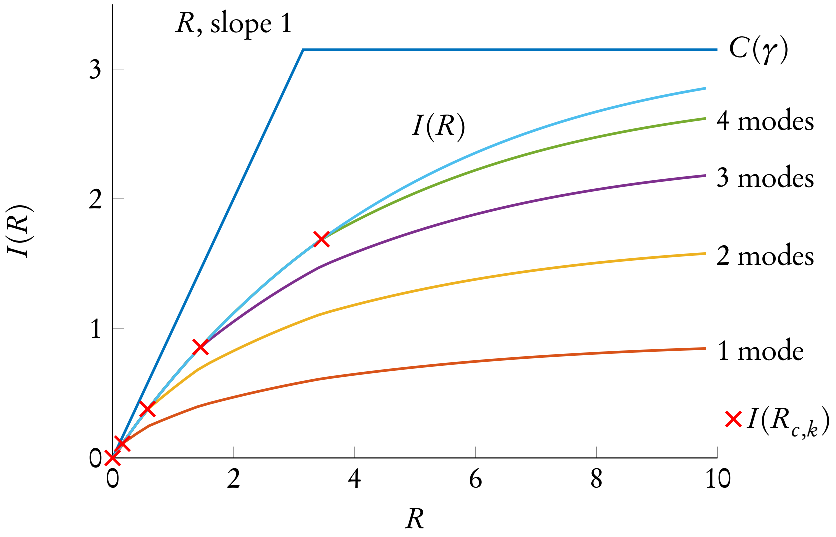


Figure 3.2: Information rate function for $\gamma = (2.5, 2, 1.5, 1, .5)$. We also added the critical points $I(R_{c,k})$ at which an additional mode becomes active and illustrate the decomposition of $I(R)$ into individual modes. The upper bound (3.37) is plotted as well.

$I(\mathbf{x}, \mathbf{y})$ and therefore,

$$I(R) \leq \min\{R, C(\gamma)\}, \tag{3.37}$$

which is also evident from (2.43) and (3.35). For $R, \beta \to \infty$, I is bounded from above by the capacity of the vector channel (3.1) for fixed input distribution $p_{\mathbf{x}} \sim \mathcal{N}(\mathbf{0}, \mathbf{C}_{\mathbf{x}})$,

$$\lim_{R\to\infty} I(R) = C(\gamma). \tag{3.38}$$

3.2 Relation between the GIB and MSE-Optimal Quantization

3.2.1 Linear Filtering and MSE-Optimal Quantization

In this section, we are going to explore the relation between the information rate tradeoff as characterized by the GIB and the rate distortion tradeoff for Gaussian random vectors and MSE distortion measure. Specifically, we will show that the optimization problem (2.7) that defines the rate distortion function also yields the optimal rate information tradeoff given suitable linear preprocessing. To this end, we investigate MSE-optimal rate distortion quantization preceded by a filter in the whitened and decorrelated domain, i.e., we consider (cf. (3.15))

$$\mathbf{w} \triangleq \boldsymbol{F}\mathbf{y}' \sim \mathcal{N}\Big(\mathbf{0}, \operatorname{diag}\big\{f_i^2(1+\gamma_i)\big\}_{i=1}^{m_y}\Big), \tag{3.39}$$

where $\boldsymbol{F} = \operatorname{diag}\{f_i\}_{i=1}^{m_y}$, and we set out to obtain the rate distortion function

$$R(D, \boldsymbol{F}) \triangleq \min_{p(\mathbf{z}|\mathbf{w})} I(\mathbf{w}; \mathbf{z}) \quad \text{subject to} \quad \mathbb{E}\Big[\|\mathbf{z} - \mathbf{w}\|^2\Big] \leq D. \tag{3.40}$$

Lemma 2. *The solution to* (3.40) *is characterized by the parametric equations*

$$R(\vartheta,\boldsymbol{F}) = \frac{1}{2}\sum_{i=1}^{m_y} \log_2^+ \frac{f_i^2(1+\gamma_i)}{\vartheta}, \tag{3.41}$$

$$D(\vartheta,\boldsymbol{F}) = \sum_{i=1}^{m_y} \min\big(\vartheta, f_i^2(1+\gamma_i)\big), \tag{3.42}$$

with waterlevel parameter $\vartheta \in [0,\infty)$.

Proof: Noting that $\mathbf{w}$ is Gaussian with covariance matrix $\boldsymbol{C}_{\mathbf{w}} = \operatorname{diag}\{\omega_i\}_{i=1}^{m_y}$ and eigenvalues $\omega_i = f_i^2(1+\gamma_i)$, the claim follows from [21, Theorem 10.3.3]. ■

While the rate distortion tradeoff in Lemma 2 is well understood, we next assess the associated rate information tradeoff (recall that the relevant information equals $I(\mathbf{x};\mathbf{z})$).

Theorem 5. *The rate information tradeoff for MSE-optimal quantization of the filtered vector* $\mathbf{w}$ *is characterized by the parametric equations*

$$I(\vartheta,\boldsymbol{F}) = \frac{1}{2}\sum_{i=1}^{m_y} \log_2^+ \left(\frac{1+\gamma_i}{1+\vartheta\frac{\gamma_i}{f_i^2(1+\gamma_i)}} \right), \tag{3.43}$$

$$R(\vartheta,\boldsymbol{F}) = \frac{1}{2}\sum_{i=1}^{m_y} \log_2^+ \left(\frac{f_i^2(1+\gamma_i)}{\vartheta} \right). \tag{3.44}$$

Here, the waterlevel parameter $\vartheta \in [0,\infty)$ *is determined by the distortion* D.

Proof: The expression (3.44) for the rate $R(\vartheta,\boldsymbol{F}) = I(\mathbf{w};\mathbf{z})$ directly follows from Lemma 2. The relevant information $I(\vartheta,\boldsymbol{F}) = I(\mathbf{x};\mathbf{z})$ in (3.43) is calculated similarly as in the proof of Theorem 4, except that the mapping (2.38), which is required to compute the covariance matrices is replaced by the *forward quantization channel* [7, p.101]. That is,

$$\mathbf{z} = \boldsymbol{B}\mathbf{w} + \mathbf{u} \tag{3.45}$$

where

$$\boldsymbol{B} = \operatorname{diag}\{b_i\}_{i=1}^{m_y} \quad \text{with} \quad b_i = 1 - \frac{D_i}{\omega_i}, \quad D_i = \min(\vartheta, \omega_i), \tag{3.46}$$

and

$$\mathbf{u} \sim \mathcal{N}(0, \boldsymbol{C}_{\mathbf{u}}), \quad \text{with} \quad \boldsymbol{C}_{\mathbf{u}} = \operatorname{diag}\{b_i D_i\}_{i=1}^{m_y}, \tag{3.47}$$

is the mapping $p(\boldsymbol{z}|\boldsymbol{w})$ solving (3.40), cf. Figure 3.3. Based on (3.45) to (3.47), the covariances

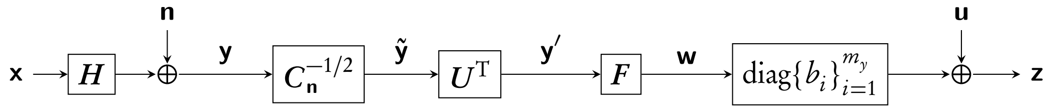


Figure 3.3: Optimal forward channel (3.45) to (3.47), linear preprocessing (3.15) and (3.39) and source model (3.1)

can be computed similarly to (3.26) and (3.29),

$$C_{\mathsf{z}} = BC_{\mathsf{w}}B^{\mathrm{T}} + C_{\mathsf{u}} = \mathrm{diag}\{b_i^2\omega_i + b_iD_i\}_{i=1}^{m_y}, \tag{3.48}$$

$$C_{\mathsf{z}|\mathsf{x}} = BC_{\mathsf{w}|\mathsf{x}}B^{\mathrm{T}} + C_{\mathsf{u}} = \mathrm{diag}\{b_i^2f_i^2 + b_iD_i\}_{i=1}^{m_y}. \tag{3.49}$$

Inserting (3.48) and (3.49) into (3.25) yields

$$I(\mathsf{x};\mathsf{z}) = \frac{1}{2}\sum_{i=1}^{m_y}\log_2\frac{b_i\omega_i + D_i}{b_if_i^2 + D_i} = \frac{1}{2}\sum_{i=1}^{m_y}\log_2\frac{\omega_i(1+\gamma_i)}{\omega_i + \gamma_iD_i} = \frac{1}{2}\sum_{i=1}^{m_y}\log_2\frac{1+\gamma_i}{1+\gamma_i\frac{D_i}{\omega_i}}, \tag{3.50}$$

which coincides with (3.43). ■

Eliminating the waterlevel ϑ from (3.43) and (3.44) yields an explicit relation between relevant information $I(\vartheta, F)$ and compression rate $R(\vartheta, F)$ similar to (3.35). Assuming that the variances ω_i are sorted in descending order, we obtain

$$I(F,R) = \frac{1}{2}\sum_{i=1}^{m_y}\log_2\frac{1+\gamma_i}{1+2^{-2R_i(R,F)}\gamma_i}, \tag{3.51}$$

where the rate allocated to mode i is given by

$$R_i(F,R) = \left[\frac{R}{j(R,F)} + \frac{1}{2}\log_2\frac{\omega_i}{\overline{\omega}_j}\right]^+. \tag{3.52}$$

Here, $j(R,F) = \max\{i : R_{\mathrm{c},i}(F) \le R\}$ denotes the number of active modes, which increases at the critical rates

$$R_{\mathrm{c},i}(F) = \frac{1}{2}\sum_{k=1}^{i}\log_2\frac{\omega_k}{\omega_i}, \tag{3.53}$$

cf. (3.31), (3.33) and (3.34).

Direct MSE-optimal quantization of y' corresponds to $F = I$ (i.e., no filtering) and noisy source coding [82] corresponds to $F = F_{\mathrm{W}} = \Gamma(I+\Gamma)^{-1}$ (i.e., Wiener filtering). Surprisingly, these two approaches are in general suboptimal in terms of the rate information tradeoff. We next identify the uniformly rate information optimum filter $F_\star$ that satisfies $I(F_\star, R) \ge I(F,R)$ for all F and any R.

Theorem 6. *The optimum filter $\boldsymbol{F}_\star$ is given by the square root of the Wiener filter (cf.* (3.13)*),*

$$\boldsymbol{F}_\star = \boldsymbol{F}_{\mathrm{W}}^{1/2} = \boldsymbol{\Gamma}^{1/2}(\boldsymbol{I}+\boldsymbol{\Gamma})^{-1/2} = \mathrm{diag}\big\{\sqrt{\mu_i}\big\}_{i=1}^{m_y} \tag{3.54}$$

and achieves the same rate information tradeoff as the GIB information-optimal projection.

Proof: The claim follows from observing that with $\boldsymbol{F} = \boldsymbol{F}_\star$ and $\vartheta = 1/(\beta - 1)$, (3.43) and (3.44) coincide with the optimal GIB tradeoff (3.22) and (3.23), respectively. (Recall the relations of eigenvalues (3.14).) ■

Lemma 3. *The number of active modes satisfies*

$$j(R,\boldsymbol{I}) \geq j(R,\boldsymbol{F}_\star) \geq j(R,\boldsymbol{F}_{\mathrm{W}}), \tag{3.55}$$

which in turn is equivalent to

$$R_{\mathrm{c},i}(\boldsymbol{I}) \leq R_{\mathrm{c},i}(\boldsymbol{F}_\star) \leq R_{\mathrm{c},i}(\boldsymbol{F}_{\mathrm{W}}). \tag{3.56}$$

The critical rates are furthermore related as

$$R_{\mathrm{c},i}(\boldsymbol{F}_\star) = \frac{R_{\mathrm{c},i}(\boldsymbol{I}) + R_{\mathrm{c},i}(\boldsymbol{F}_{\mathrm{W}})}{2}. \tag{3.57}$$

Proof: The expression (3.56) and (3.57) can be verified directly from (3.53). The double inequality (3.55) follows from the definition of $j(R,\boldsymbol{F})$ in terms of the critical rates. ■

3.2.2 Information Rate Tradeoff for Different Filtering Approaches

We note that any scaled version of $\boldsymbol{F}_\star$ is also rate information optimal. If the nonzero mode SNRs are identical, i.e., if $\gamma_i \in \{\gamma, 0\}$, then we have $\boldsymbol{F}_{\mathrm{W}} = \sqrt{\gamma/(\gamma+1)}\boldsymbol{F}_\star$ and hence in this case MSE-optimal noisy source coding is rate information optimal. However, for widely different mode SNRs γ_i, $\boldsymbol{F}_{\mathrm{W}}$ and other suboptimal filters perform substantially worse.

We next consider the filters $\boldsymbol{F}(n) \triangleq \boldsymbol{F}_{\mathrm{W}}^{n} = \mathrm{diag}\big\{\mu_i^n\big\}_{i=1}^{m_y}$ to illustrate the transition from the unfiltered case ($n = 0$) to rate information optimal filtering ($n = 1/2$) and Wiener filtering ($n = 1$). We assume $m_y = 10$ and mode SNRs $\gamma_i = 2^{-ci}$, $i = 1, \ldots, m_y$, with c chosen such that $C(\gamma) = 1$. Figure 3.4a shows the information rate curve $I\big(\boldsymbol{F}(n), R\big)$ for various n. Direct quantization without filtering ($n = 0$) is seen to perform worst among the curves shown because it uses too many modes and allocates too little rate to the strongest modes (cf. Lemma 3). As n increases, the information rate tradeoff improves and is identical to the GIB optimum for $n = 1/2$. Increasing n beyond $1/2$ deteriorates the rate information performance. Noisy source coding with Wiener filtering ($n = 1$) performs slightly poorer than the optimal solution since according to Lemma 3 too few modes are used, i.e., too much rate is allocated to the strongest modes. Interestingly, the information rate curve is no longer concave for $n > 1/2$.

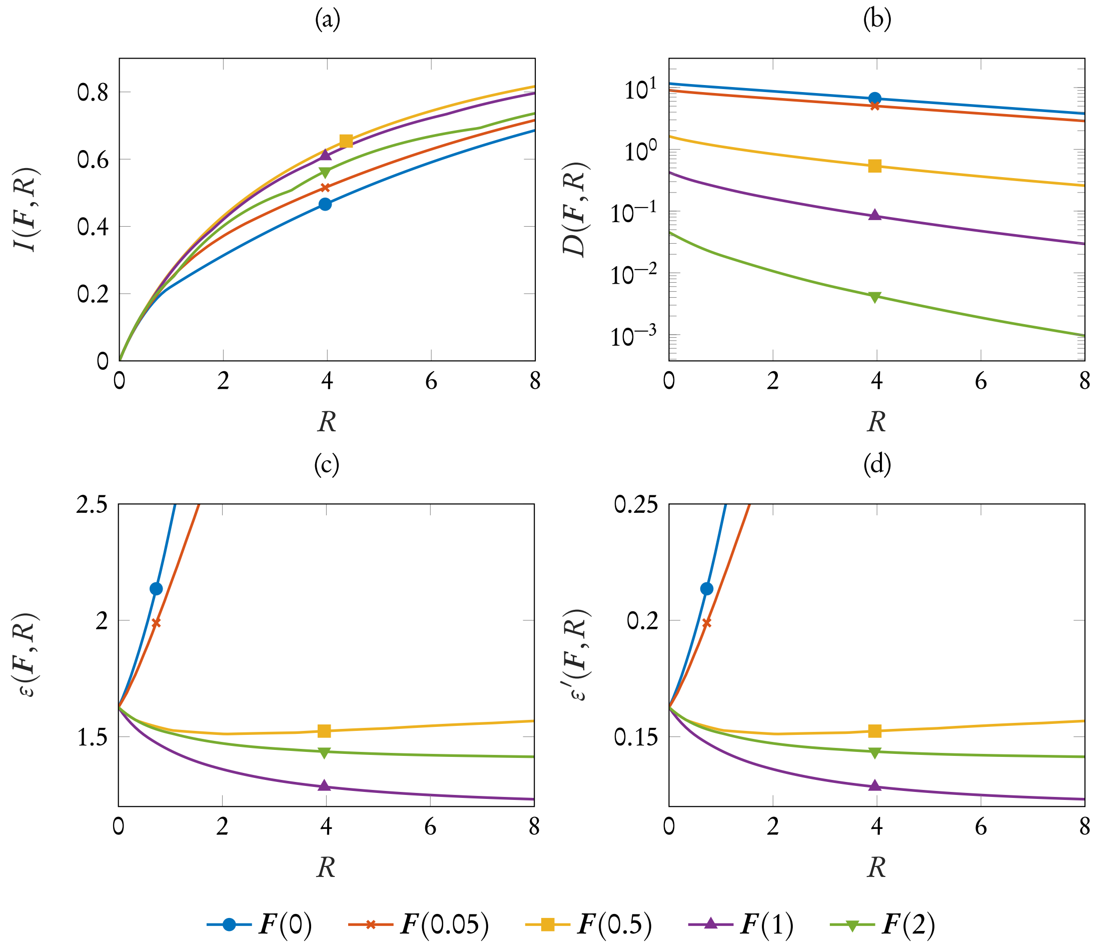


Figure 3.4: (a) Information rate function $I(F(n),R)$, (b) distortion rate function, (c) MSE (3.60) when using a noise whitening filter, (d) MSE (3.62) for white noise without whitening filter, for various filter exponents n

Finally, we note that in general, the relative order in terms of information rate performance for various n depends on the distribution of the mode SNRs γ_i.

Next, we analyze the impact of $F(n)$ on the rate distortion tradeoff (3.41) and (3.42) as well as on the MSE between quantizer output $\mathbf{z}$ and channel input $\mathbf{x}$. Recall that W is the Wiener filter for estimating $H\mathbf{x}$ from $\mathbf{y}$. Thus, in the whitened and decorrelated domain, $F = F_{\mathrm{W}}$ minimizes $\mathbb{E}\left[||F\mathbf{y}' - \mathbf{x}'||^2\right]$ where we define $\mathbf{x}' \triangleq U^{\mathrm{T}} C_{\mathbf{n}}^{-1/2} H\mathbf{x}$. According to [82], the Wiener filter is also optimal for noisy source coding, i.e., using a source code designed for $\mathbf{x}$ on the Wiener filtered observation $\mathbf{y}$ is optimal in terms of minimizing the MSE distortion between the source and the decoder output. In the whitened and decorrelated domain, this MSE is given by

$$\varepsilon(F) = \mathbb{E}\left[||\mathbf{z} - \mathbf{x}'||^2\right]. \tag{3.58}$$

According to (3.15), (3.39) and (3.45),

$$\mathbf{z} = \boldsymbol{B}\boldsymbol{F}\boldsymbol{U}^{\mathrm{T}}\boldsymbol{C}_{\mathbf{n}}^{-1/2}\boldsymbol{H}\mathbf{x} + \boldsymbol{B}\boldsymbol{F}\boldsymbol{U}^{\mathrm{T}}\boldsymbol{C}_{\mathbf{n}}^{-1/2}\mathbf{n} + \mathbf{u} \tag{3.59}$$

and $\mathbf{x}$ are jointly Gaussian, so we can express (3.58) in terms of the eigenvalues ε_i of the covariance matrix $\boldsymbol{C}_{\mathbf{z}-\mathbf{x}'}$,

$$\varepsilon(\boldsymbol{F}) = \sum_{i=1}^{m_y} \varepsilon_i(f_i) = \sum_{i=1}^{m_y} (b_i f_i - 1)^2 \gamma_i + b_i^2 f_i^2 + b_i D_i. \tag{3.60}$$

Note that due to the effect of the noise whitening filter,

$$\lim_{\gamma_i \to \infty} \lim_{D_i \to 0} \varepsilon_i(f_{\mathrm{W},i}) = \lim_{\gamma_i \to \infty} \frac{\gamma_i}{1+\gamma_i} = 1, \tag{3.61}$$

i.e., the error does not vanish for infinite SNR and resolution. For white noise, $\boldsymbol{C}_{\mathbf{n}} = \sigma^2 \boldsymbol{I}$, we can omit the noise whitening filter in our derivations and obtain

$$\varepsilon'(\boldsymbol{F}) = \sum_{i=1}^{m_y} \varepsilon_i'(f_i) = \sum_{i=1}^{m_y} (b_i' f_i - 1)^2 \rho_i + b_i'^2 f_i^2 \sigma^2 + b_i' D_i', \tag{3.62}$$

with mode powers ρ_i given by the eigendecomposition $\boldsymbol{H}\boldsymbol{C}_{\mathbf{x}}\boldsymbol{H}^{\mathrm{T}} = \boldsymbol{U}\mathrm{diag}\{\rho_i\}_{i=1}^{m_y}\boldsymbol{U}^{\mathrm{T}}$. Hence, for the noiseless case and perfect quantization,

$$\lim_{\sigma^2 \to 0} \lim_{D_i' \to 0} \varepsilon_i'(f_{\mathrm{W},i}') = \lim_{\sigma^2 \to 0} \sigma^2 \frac{\gamma_i'}{1+\gamma_i'} = 0, \tag{3.63}$$

where the components of the Wiener filter are again given by $f_{\mathrm{W},i}' = \frac{\gamma_i'}{1+\gamma_i'}$ with $\gamma_i' = \frac{\rho_i}{\sigma^2}$. Note that both (3.60) and (3.62) depend on the rate R via the waterlevel parameter, cf. (3.46). In Figure 3.4b, we plot $D(\boldsymbol{F}(n), R)$ as a function of the rate for various filters $\boldsymbol{F}(n)$. As we can see, D becomes increasingly smaller for larger n. This is because the optimal forward channel is the solution minimizing $D = \mathbb{E}\left[||\mathbf{w}-\mathbf{z}||^2\right]$, i.e., it takes into account $\boldsymbol{F}$ and minimized D for any $\boldsymbol{F}$. Since increasing n reduces the overall variance of $\mathbf{w}$ due to $\mu_i < 1$, increasing n also reduces the absolute value of $\mathbb{E}\left[||\mathbf{w}-\mathbf{z}||^2\right]$.

This is different for the MSEs (3.60) and (3.62). Here, we face a mismatch between the objective of the forward channel and minimizing (3.60) and (3.62), where the mismatch can be corrected by using the appropriate filter. In fact, Figure 3.4c and Figure 3.4d confirm the result of [82] that the Wiener filter is optimal for noisy source coding. Interestingly, the MSEs are increasing in R for some choice of coefficients n, reflecting that the mismatch is not adequately corrected by those filters. At this point, we want to emphasize that except for $\boldsymbol{F}_{\mathrm{W}}$, neither (3.60) nor (3.62) are distortion rate functions, i.e., they are not necessarily non-increasing or convex. Finally, note that Figure 3.4c and Figure 3.4d appear to be very similar up to a difference in scaling which is due to the effect of the noise whitening. For this particular example, we chose a noise variance $\sigma^2 = 0.1$ for the non-whitened error (3.62). We can confirm that both $\varepsilon(\boldsymbol{F},R)$ and $\varepsilon'(\boldsymbol{F},R)$ converge to their respective limits $\lim_{R\to\infty}\varepsilon(\boldsymbol{F}_{\mathrm{W}},R) \approx 1.202$ and $\lim_{R\to\infty}\varepsilon'(\boldsymbol{F}'_{\mathrm{W}},R) \approx 0.1202$ as given by (3.61) and (3.63).

3.2.3 Extension to Stationary Random Processes

We next derive an extension of our results to the case where $\mathsf{x}[k]$ and $\mathsf{n}[k]$ are independent stationary Gaussian processes with power spectral densities $S_{\mathsf{x}}(\theta)$ and $S_{\mathsf{n}}(\theta)$ and

$$\mathsf{y}[k] = \sum_{k'=-\infty}^{\infty} h[k']\mathsf{x}[k-k'] + \mathsf{n}[k], \tag{3.64}$$

with $h[k]$ the impulse response of a linear time-invariant filter. For a finite time interval of duration n, this model reduces to (3.1) with $\boldsymbol{H}$ a Toeplitz matrix induced by $h[k]$ and all covariance matrices being Toeplitz as well.

We can then obtain asymptotic frequency-domain versions of all results derived above by defining $R(I)$ and $I(R)$ in terms of the *mutual information rate*

$$\mathscr{I}(\mathsf{x};\mathsf{y}) \triangleq \lim_{n\to\infty} \frac{1}{n} I(\mathsf{x}^n;\mathsf{y}^n) \tag{3.65}$$

and by invoking the fundamental eigenvalue theorem of Grenander and Szegö (cf. [41, Chapter 7] and [39, Theorem 4.2]) for the continuous positive function $\log_2^+$. In particular, MSE-optimal source coding of the filtered observation $\mathsf{w}[k] = \sum_{k'=-\infty}^{\infty} f[k']\mathsf{y}[k-k']$ with power spectral density

$$S_{\mathsf{w}}(\theta) = |F(\theta)|^2 \left(|H(\theta)|^2 S_{\mathsf{x}}(\theta) + S_{\mathsf{n}}(\theta)\right) \tag{3.66}$$

leads to the rate information tradeoff (cf. (3.43), (3.44))

$$\mathscr{I}(\vartheta,F) = \frac{1}{4\pi}\int_{-\pi}^{\pi} \log_2^+\left(\frac{1+\Gamma(\theta)}{1+\vartheta\frac{\Gamma(\theta)}{|F(\theta)|^2(1+\Gamma(\theta))}}\right)\mathrm{d}\theta, \tag{3.67}$$

$$\mathscr{R}(\vartheta,F) = \frac{1}{4\pi}\int_{-\pi}^{\pi} \log_2^+\left(\frac{|F(\theta)|^2\left(1+\Gamma(\theta)\right)}{\vartheta}\right)\mathrm{d}\theta. \tag{3.68}$$

Here, $F(\theta)$ and $H(\theta)$ denote the frequency responses of the filter $f[k]$ and the channel $h[k]$ and we used the SNR spectrum

$$\Gamma(\theta) = |H(\theta)|^2 S_{\mathsf{x}}(\theta)/S_{\mathsf{n}}(\theta). \tag{3.69}$$

The optimal filter is given by

$$F_\star(\theta) = \sqrt{\frac{\Gamma(\theta)}{1+\Gamma(\theta)}}, \exp\big(\mathrm{j}\phi(\theta)\big) \tag{3.70}$$

with arbitrary real valued phase function $\phi(\theta)$. Plugging (3.70) into (3.67) and (3.68) and substituting $\vartheta = 1/(\beta-1)$, the optimal rate information tradeoff for stationary sources is characterized by the parametric equations

$$\mathscr{I}(\beta) = \frac{1}{4\pi}\int_{-\pi}^{\pi} \log_2^+\left(\frac{\beta-1}{\beta}\big(1+\Gamma(\theta)\big)\right)\mathrm{d}\theta, \tag{3.71}$$

$$\mathscr{R}(\beta) = \frac{1}{4\pi}\int_{-\pi}^{\pi} \log_2^+\big((\beta-1)\Gamma(\theta)\big)\mathrm{d}\theta. \tag{3.72}$$

Hence, Theorems 4 to 6 remain valid and the parametric expressions (3.22), (3.23), (3.43) and (3.44) are replaced by (3.67), (3.68), (3.71) and (3.72).

3.3 Operational Setting and Coding Theorems

In the preceding sections, we mainly focused on analyzing information theoretic tradeoff problems. In this section, we will give operational meaning to some of those tradeoffs. Furthermore, based on the results from Section 3.2, we argue that for Gaussian sources, information-optimal source codes can be designed based on codes that are optimized towards a small MSE.

3.3.1 A Coding Theorem for Jointly Gaussian Relevance and Observation

In Section 2.1.3, rate information coding was introduced for sequences of discrete, iid random variables. In this section, we show that similar results can be obtained for Gaussian vectors, i.e., the achievable region of rate information codes is again determined by the information rate or rate information function, cf. Definition 2. Recall that we define the achievable region $\mathcal{R}$ as the set of pairs (R,I) for which a $(2^{nR},n)$ rate information code exists such that for some (large) n, $I^{\mathrm{RI}} \geq I$, where I^{RI} can be defined as either (2.18) or (2.19). We can then formulate a coding theorem for a sequence of Gaussian vectors with jointly Gaussian relevance similar to Theorem 2.

Theorem 7. *Consider a sequence of vectors $(\mathsf{x}^n,\mathsf{y}^n)$ drawn iid from a Gaussian joint distribution, i.e. $(\mathsf{x}_i,\mathsf{y}_i)\sim\mathcal{N}(\mathsf{0},\mathbf{C}_{\mathsf{x},\mathsf{y}})$ for $i=1,\dots,n$, with full rank $\mathbf{C}_{\mathsf{x},\mathsf{y}}\in\mathbb{R}^{m_x\times m_y}$. We then have $R(I)=\inf_{(R,I)\in\mathcal{R}} R$ and $I(R)=\sup_{(R,I)\in\mathcal{R}} I$ with $R(I)$ and $I(R)$ as in Definition 2 for the above setting of relevance and information.*

Proof sketch: The proof is rather technical, so we will only outline it and refer to the literature whenever possible. First, note that it suffices to show the proposition for $R(I)$, since $I(R)$ is the inverse to $R(I)$, cf. [34, 111].

To show the converse, we need to show that for any code satisfying $I^{\mathrm{RI}} \geq I$, the code rate R code is lower-bounded by $R(I)$. This can be done analogously to the discrete case (cf. [34] or [21, Section 10.4]) by replacing entropies with differential entropies and noting that for Gaussians, $R(I)$ is convex and non increasing [112].

In order to prove achievability, we need to show that for any I and $\epsilon > 0$, a code of rate $R = R(I)$ can be constructed with $I^{\mathrm{RI}} \geq I - \epsilon$. For rate distortion coding and discrete sources, achievability is usually shown using random codebook generation and typicality de- and encoding, cf. [21]. The extension to continuous sources is then reduced to the achievability for the discrete case via quantization. Here, one exploits the fact that the mutual information of two continuous variables can be approximated arbitrarily well by the mutual information of increasingly finer quantizations of those variables cf. [28, Section 2.3 and Section 3.8] and [38, Chapter 5]. Specifically, we have

$$I(\mathbf{x},\mathbf{y}) = \sup_{a,b} I\big(a(\mathbf{x}); b(\mathbf{y})\big), \tag{3.73}$$

where the supremum is taken over all finite-valued functions a and b.

For rate information coding the argument is similar, however, the extension is less straightforward since in addition to the mutual information between encoder input and decoder output, we also have to consider the mutual information between source and relevance. Let $[x] = a(x)$ and $[y] = b(y)$ denote the quantization with finite valued functions a and b. Based on (3.73), we argue that the achievability for any sequence of discrete pairs $\big([\mathbf{x}_i],[\mathbf{y}_i]\big)$ implies the achievability for the continuous case. For the details on this transition cf. [72]. ■

Next, we discuss a potential extension of Theorem 7 to the case of stationary ergodic Gaussian processes. Note that for rate distortion codes, one can define

$$R_n(D) = \inf_{p(\mathrm{z}^n|\mathrm{y}^n)} \frac{1}{n} I(\mathrm{y}^n,\mathrm{z}^n) \quad \text{subject to} \quad \mathbb{E}[\rho_n(\mathrm{y}^n,\mathrm{z}^n)] \leq D, \tag{3.74}$$

with ρ_n as in (2.3) and (2.4). It is then shown in [7, Section 7] that

$$\lim_{n\to\infty} R_n(D) = \mathscr{R}(D) \tag{3.75}$$

exists and that $\mathscr{R}(D)$ is in fact the rate distortion function of a stationary ergodic source $\mathrm{y}[k]$. Similarly, we define

$$R_n(I) = \inf_{p(\mathrm{z}^n|\mathrm{y}^n)} \frac{1}{n} I(\mathrm{y}^n,\mathrm{z}^n) \quad \text{subject to} \quad \frac{1}{n} I(\mathrm{x}^n,\mathrm{z}^n) \geq I, \tag{3.76}$$

and note that for $n \to \infty$, the tradeoff (3.76) is given by (3.71) and (3.72). We thus conjecture that (3.71) and (3.72) characterize the rate information function for jointly stationary- and Gaussian, ergodic relevance and observation (note that the joint stationarity of $\mathrm{x}[k]$ and $\mathrm{y}[k]$ is implied by (3.64)). We do however not have a formal proof for this, as the measure-theoretic

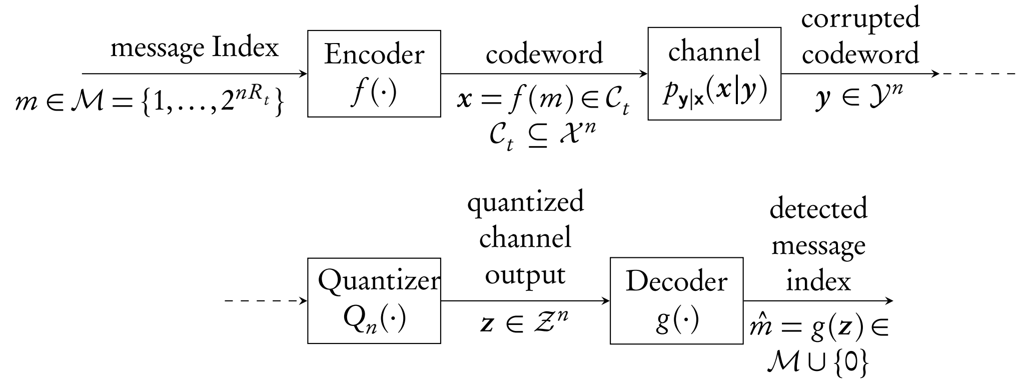


Figure 3.5: Operational model for channel output quantization

concepts that are required to conduct a proof along the lines of [7, Theorems 7.2.4 and 7.2.5] are beyond the scope of this work.

3.3.2 Capacity under Channel Output Quantization

Due to (3.35) and (3.38), one might be tempted to think of $I(R)$ as the capacity of the channel (3.1) under output compression, i.e., the maximum attainable rate of communication I over the channel under some compression rate constraint R on the output. In Figure 3.5, we illustrate such a setting for a general iid channel. Here, a discrete message m is encoded by $f:\{1,\dots,2^{nR_t}\}\to\mathcal{C}_t\subset\mathcal{X}^n$ and transmitted over an iid channel

$$p(x|y) = \prod_{i=1}^{n} p(x_i|y_i). \tag{3.77}$$

The channel output y is then subject to quantization

$$Q_n : \mathcal{Y}^n \to \mathcal{C}_q \subset \mathcal{Z}^n \quad \text{with rate} \quad R_q = \frac{\log_2|\mathcal{C}_q|}{n} \tag{3.78}$$

followed by a decoder $g:\mathcal{Z}^n\to\{1,\dots,2^{nR_t}\}$. Define the achievable region $\mathcal{R}$ for this setting as the pairs (R_r,R_q) for which a code and quantizer exist such that for any $\epsilon>0$, $\mathrm{P}\{\mathsf{m}\neq\hat{\mathsf{m}}\}<\epsilon$ for some n. Unfortunately, we could not find a simple single letter characterization for the general problem of a vector quantizer (3.78). However, once we restrict ourselves to a single, scalar (potentially probabilistic) quantizer, we can obtain the following result.

Theorem 8. *Consider the system depicted in Figure 3.5 with discrete memoryless channel (DMC) $p(y|x)$ where a single scalar quantizer is used on all channel outputs, i.e. $z_i = \left(Q_n(z^n)\right)_i = Q(y_i)$ with $Q:\mathcal{Y}\to\mathcal{Z}$ for all $i=1,\dots,n$. Then*

$$\sup_{(R_t,R_q)\in\mathcal{R}} R_t = \max_{p(x),Q} I(\mathsf{x},\mathsf{z}) \quad \textit{subject to} \quad \log_2|\mathcal{Z}|\le R_q, \tag{3.79}$$

with deterministic quantizer Q.

Proof: The statement follows straightforwardly from the channel coding theorem [87] for DMCs by noting that any quantizer Q can be represented by a conditional pmf $p(z|y)$, essentially transforming the DMC $p(y|x)$ into a DMC

$$p(z|x) = \sum_{y \in \mathcal{Y}} p(z|y)p(y|x). \tag{3.80}$$

The quantizer introduces an additional degree of freedom into the maximization of the mutual information between channel input and transformed channel output, giving rise to the optimization over both $p(x)$ and Q in (3.79). The constraint $\log_2|\mathcal{Z}| \leq R_q$ is due to $\mathcal{C}_q = \mathcal{Z}^n$ for scalar quantizers, cf. (3.78). To show that Q is deterministic, note that we argued in Section 2.1.4 that (2.24) is maximized by a deterministic mapping $p(z|y)$ for any $p(x,y)$. ■

An extension of Theorem 8 to continuous output channels under an input power constraint is possible by first adding a cost constraint to the discrete case followed by a transition to the continuous domain via the limit of increasingly finer quantization, cf. [28, Section 3.3 and 3.4]. For an average power constraint $\frac{1}{n}\sum_{i=1}^{n} x_i \leq P$ for all $x^n \in \mathcal{C}_t$, the capacity is then given by

$$\max_{p(x),Q} I(\mathrm{x},\mathrm{z}) \quad \text{subject to} \quad \log_2|\mathcal{Z}| \leq R_q, \quad \mathbb{E}\left[\mathrm{x}^2\right] \leq P. \tag{3.81}$$

Unfortunately, unlike the unquantized case, (3.81) is notoriously difficult to solve due to the double optimization. The authors of [90] analyze (3.81) for AWGN channels and symmetric quantizers with convex assignment regions, i.e., $p(y|x) \sim \mathcal{N}(x,\sigma^2)$, $Q(y) = z \Rightarrow Q(-y) = -z$ and $Q(y_1) = Q(y_2) = z \Rightarrow Q\big(\theta y_1 + (1-\theta)y_2\big) = z$, $\forall \theta \in [0,1]$. Under symmetry and binary quantization ($|\mathcal{Z}| = 2$), there is no need for an optimization over Q and (3.81) is maximized by $p(x = \pm\sqrt{P}) = 1/2$, yielding

$$I(\mathrm{x},\mathrm{z}) = 1 - h\left(\mathcal{Q}\left(\sqrt{\frac{P}{\sigma^2}}\right)\right), \tag{3.82}$$

where $h(p) \triangleq -p\log_2 p - (1-p)\log_2(1-p)$, $0 \leq p \leq 1$ is the binary entropy function and $\mathcal{Q}(x) \triangleq \frac{1}{\sqrt{2\pi}}\int_x^\infty \exp\big(-\zeta^2/2\big)\mathrm{d}\zeta$. For $|\mathcal{Z}| > 2$, (3.81) can only be optimized numerically. In Figure 3.6, we compare (3.82) to the Gaussian tradeoff (3.35) for $m_x = m_y = 1$. While a Gaussian input distribution $p(x)$ maximizes $I(\mathrm{x};\mathrm{z})$ for a Gaussian $p(z|x)$ and a Gaussian $p(z|y)$ maximizes $I(\mathrm{x};\mathrm{z})$ for jointly Gaussian $p(x,y)$, Figure 3.6 shows that Gaussian $p(x)$ and $p(z|y)$ are generally not jointly optimal. Lastly, note that —similar to the design of deterministic quantizers in Section 2.1.4— the rate constraint $I(\mathrm{y};\mathrm{z}) \leq R$ is dropped in favour of a limitation on the size of the output alphabet $|\mathcal{Z}|$.

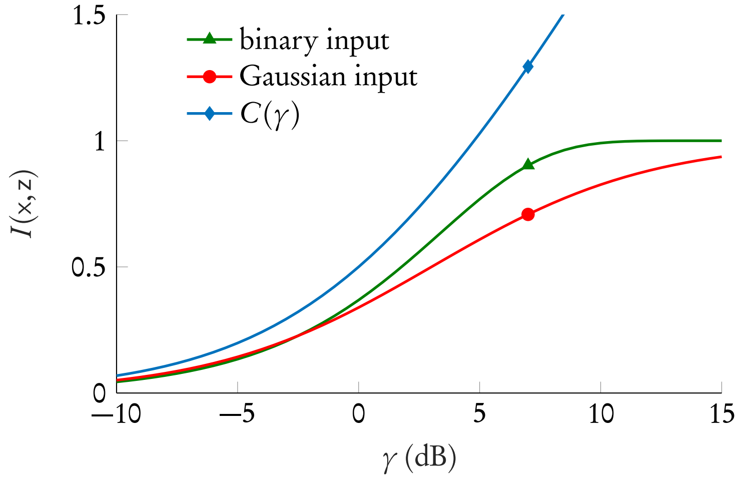


Figure 3.6: $I(\mathsf{x},\mathsf{z})$ as a function of SNR γ for a fixed quantization rate of $R = I(\mathsf{y},\mathsf{z}) = 1$ bit.

3.3.3 Deriving Rate Information Codes from Rate Distortion Codes

In this section, we discuss how the relationships we derived in Section 3.2 can be used to obtain codes with high average relevant information from codes that are optimized towards low MSE. Throughout this subsection, we denote length-n operational sequences of elements as vectors, i.e., $x^n = \boldsymbol{x}$, where x could be a vector itself. Assuming $x_i = (x_{i,1}, \ldots, x_{i,m})$ is a length-m vector, then $\boldsymbol{x} = (x_1, \ldots, x_n) = (x_{1,1}, \ldots, x_{1,m}, \ldots, x_{n,m})$.

We begin by briefly revisiting the case of MSE-optimal source coding of a disturbed source. Sakrison showed in [82] that

$$\begin{aligned} \mathcal{Y}_k^n &\triangleq \left\{ y \mid \mathbb{E}\left[||\,\mathbf{x} - \boldsymbol{x}_k||^2 \mid \mathbf{y} = y \right] \leq \mathbb{E}\left[||\,\mathbf{x} - \boldsymbol{x}_j||^2 \mid \mathbf{y} = y \right], \forall k \neq j \right\} \\ &= \left\{ y \mid ||\hat{x}(y) - \boldsymbol{x}_k||^2 \leq ||\hat{x}(y) - \boldsymbol{x}_j||^2, \forall k \neq j \right\}, \end{aligned} \tag{3.83}$$

where $\mathbf{y}$ is a randomly perturbed version of $\mathbf{x}$ and $\hat{x}(y)$ is the (generally non-linear) MMSE estimate of x from some particular y. For jointly Gaussian $\mathbf{x}$ and $\mathbf{y}$ the MMSE estimation is equivalent to Wiener filtering. This means that *any* MSE-optimal code designed for a source $\mathbf{x}$ can by employed on $\hat{\mathbf{x}}(\mathbf{y})$ and minimizes the MSE between decoder output and source given the observation $\mathbf{y}$.

In Section 3.2, we derived the equivalence between the GIB and the linearly prefiltered rate distortion forward channel, i.e., we showed that through linear preprocessing, $p(z|y)$ can be made identical for both cases. Unfortunately, this does not imply a direct operational relation such as (3.83), i.e., in general, irrespective of the filter $\boldsymbol{F}$,

$$\left\{ y \mid d(y, z_k) \leq d(y, z_j) \,\forall k \neq j \right\} \neq \left\{ y \mid ||\boldsymbol{F} y - z_j||^2 \leq ||\boldsymbol{F} y - z_k||^2 \,\forall k \neq j \right\}, \tag{3.84}$$

for $\boldsymbol{z}_k$ taken from an MSE optimized codebook $\mathcal{Z}^n = \{\boldsymbol{z} \mid \boldsymbol{z} = f_n(g_n(\boldsymbol{y})), \boldsymbol{y} \in \mathcal{Y}^n\}$ and[2]

$$d(\boldsymbol{y},\boldsymbol{z}) = D\big(p(x|\boldsymbol{y})\|p(x|\boldsymbol{z})\big) \tag{3.85}$$

with

$$p(\boldsymbol{z}|\boldsymbol{y}) = \begin{cases} 1, & f_n(g_n(\boldsymbol{y})) = \boldsymbol{z}, \\ 0, & \text{otherwise}, \end{cases} \tag{3.86}$$

implied by the code.

Nonetheless, the equivalence in $p(z|y)$ for appropriately chosen $\boldsymbol{F}$ implies the *existence of codes* that minimize $\mathbb{E}\Big[\|\boldsymbol{F}\mathbf{y} - g_n\big(f_n(\boldsymbol{F}\mathbf{y})\big)\|^2\Big]$ and maximize the average relevant information (2.19) under a given code rate constraint, i.e., there exist codes that reach both the rate distortion tradeoff (Lemma 2) as well as the rate information tradeoff (Theorem 4). To see this, we turn to the achievability proof of the rate distortion theorem using typicality coding. For typicality coding, the codebook $\mathcal{Z}^n$ is randomly generated from $p(z|y)$, i.e., $\boldsymbol{z}_k$ are generated by drawing iid samples from $p(z) = \sum_{y \in \mathcal{Y}} p(z|y)p(y)$ and $\boldsymbol{y}$ is mapped to $\boldsymbol{z}_k$ if $(\boldsymbol{y}, \boldsymbol{z}_k)$ are jointly typical. Since $p(z|y)$ is identical when using the square root Wiener filter, this implies the existence of said code.

Furthermore, even though the theory does not allow for an exact equivalence between information- and MSE-optimal codes, numeric results suggest that there is practical merit to the equivalence of $p(z|y)$. For $m_x = m_y = m = 5$ we generated 10^5 length-m vectors $x = (x_1, \ldots, x_m)$ and applied (3.1), (3.15) and different filters $\boldsymbol{F}(n) = \operatorname{diag}\{\mu_i^n\}_{i=1}^5$, cf. Section 3.2. We then designed MSE optimized codebooks of different rates using the LLoyd-Max algorithm on the filtered samples and estimated the average mutual information[3]

$$\frac{1}{m}\sum_{k=1}^{m} \hat{I}(\mathsf{x}_k, \mathsf{z}). \tag{3.87}$$

Here, x_k, $k = 1, \ldots, m$ is the kth input component, $\mathsf{z}(\mathsf{x}) \in \{1, \ldots, 2^R\}$ is the index of the codeword corresponding to x and $\hat{I}(\mathsf{x}_k, \mathsf{z})$ has been estimated independently for each dimension based on [55]. In Figure 3.7, we plot the estimated average relevant information as a function of the code rate. As we can see, the best information rate performance is achieved by the square root Wiener filter $\boldsymbol{F}(0.5)$, followed by the Wiener Filter $\boldsymbol{F}(1)$.

[2] Assuming we employ the definition (2.19). When using (2.18), then $d(\boldsymbol{y},\boldsymbol{z}) = \sum_{i=1}^{n} d(y_i, z_i)$ with d as in (2.12) and marginals of (3.86) .

[3] We chose (3.87) as a figure of merit because computing the mutual information analytically is not possible and the mutual information of random vectors (e.g. $I(\mathsf{x},\mathsf{z})$) can not be estimated reliably.

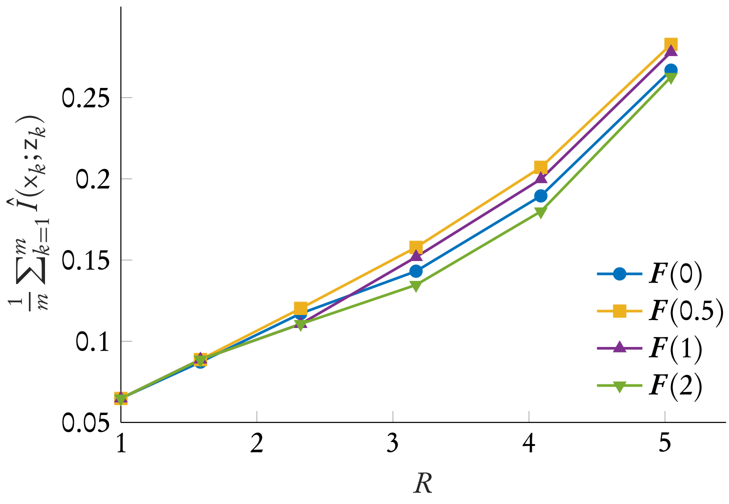


Figure 3.7: Estimated average relevant information (2.18)

Chapter 4

Quantized Message Passing LDPC Decoding

In this chapter, we derive quantized message passing decoders where the LDPC VN and CN node updates are realized as generic look-up tables (LUTs). cf. Section 2.2. Furthermore, we provide a general analysis of LDPC decoding with LUTs that is independent of the specific LUT design and we identify LUT decoding as a form of quantized BP.

Specifically, in Section 4.1, we derive fundamental symmetry constraints for LUT decoding and obtain the general result that LUT decoding can be viewed as a form of BP, where after every iteration, the outputs of the BP node updates are projected onto a finite message set by the use of LUTs. Section 4.2 then discusses the design of information-optimal LUTs that respect the required symmetries, as well as hybrid min-LUT decoding and hierarchical LUT structures for complexity reduction. Next, Section 4.3 introduces LUT decoders for irregular LDPC codes. Eventually, Section 4.4 discusses performance and design aspects of LUT decoders and presents numerical results that illustrate how design parameters affect decoding performance.

4.1 Symmetric Discrete Message Passing and Density Evolution

Recall that for BP-based decoding, the update rules (2.59) and (2.61) combine real-valued LLR messages by algebraic operations and well-known real functions. By design, the BP decoding algorithm seeks to find the probability that a certain code bit was either zero or one, conditioned on the received channel output, while taking into account the parity check constraints of the code. For practical implementations of BP, messages are quantized and approximated by fixed point numbers but the LLR interpretation of messages is still upheld.

There is also a large class of heuristic discrete message passing schemes that are designed to mimic BP, (e.g., [78, Example 7]). These algorithms operate on finite message sets that are subsets of the reals $\mathbb{R}$ or integers $\mathbb{Z}$ and therefore also use operations such as multiplication and addition to combine messages.

In the case of LUT decoding, we deal with discrete message sets and update rules (2.50) and (2.51) that are a priori devoid of any algebraic structure or probabilistic interpretation.

Previous work mainly focused on design methodologies for the LUT updates, e.g., maximizing the local information flow through the code's factor graph [59, 60, 80, 81], or reducing error floors that are a common problem for quantized decoding [74]. To the best of our knowledge, no general, systematic analysis of LUT decoding exists in the literature.

In what follows, we derive a generalized framework based on discrete density evolution that links LUT decoding to BP — irrespective of the particular LUT design methodology. For that, we first extend the sign-based symmetry concepts from Section 2.2.3 to discrete messages using involutions, cf. Section 4.1.1. Furthermore, we introduce a duality between discrete messages and their LLR values that we define using the message pmfs of density evolution. This duality will turn out to be useful in several aspects: First of all, we use it to establish algebraic structure on the discrete message sets. We later exploit this structure in Section 4.2 for reduced complexity LUT designs as well as for the derivation of the hybrid min-LUT algorithm. Secondly, we use the duality to show that discrete symmetric message passing can be interpreted as quantized BP, where after every message update, LLRs are projected back onto a finite set. This is a fundamental result, as it implies that many concepts originally developed for BP decoding carry over to LUT decoding. In fact, we use this result to analyze the stability of LUT decoding in Section 4.3.

4.1.1 Message Labels, Values and Symmetry

Let $x \in \mathrm{GF}(2)$ and k denote a *label* from a finite set $\mathcal{K} = \{0, 1, \ldots, K-1\}$ with conditional label pmf $p_{\mathsf{k}|\mathsf{x}}(k|x)$. At this point of the discussion, we use the generic label set $\mathcal{K}$ as a placeholder; after establishing some basic concepts, we will substitute the LLR and message sets $\mathcal{L}$, $\mathcal{M}_\ell$, $\overline{\mathcal{M}}_\ell$ and product sets thereof for $\mathcal{K}$, cf. Section 2.2.1. We define the *(LLR) value* corresponding to label k in terms of the conditional label pmf,

$$k' \triangleq \mathrm{L}(k) = \log \frac{p_{\mathsf{k}|\mathsf{x}}(k|+1)}{p_{\mathsf{k}|\mathsf{x}}(k|-1)} \in \mathcal{K}', \tag{4.1}$$

cf. (2.30). Without loss of generality, we only consider the case where (4.1) is bijective, i.e., there are no labels k_1, k_2 that share the same LLR, $k_1' = k_2'$. The more general case can always be reduced to a bijective relation by means of label merging. That is, if $k_1' = k_2'$, we merge k_1 and k_2 into a new label k_3 with conditional probability $p_{\mathsf{k}|\mathsf{x}}(k_3|x) = p_{\mathsf{k}|\mathsf{x}}(k_1|x) + p_{\mathsf{k}|\mathsf{x}}(k_2|x)$, corresponding to $k_3' = k_2' = k_1'$. As pointed out in Section 2.1.4 , merging labels with the same LLR leaves the mutual information $I(\mathsf{k}, \mathsf{x})$ unchanged and hence, both the merged and unmerged case are identical from an information-theoretic point of view.

Due to the bijection between $\mathcal{K}$ and $\mathcal{K}'$,

$$p_{\mathsf{k}|\mathsf{x}}(k|x) = p_{\mathsf{k}'|\mathsf{x}}(k'|x). \tag{4.2}$$

Combining (4.1) and (4.2), we see that LLR-consistency (2.86) holds for all LLRs $k' \in \mathcal{K}'$.

Instead of using the sign-based symmetry concepts from Section 2.2.3, we define symmetry for discrete conditional pmfs as follows:

Definition 3 (Discrete, symmetric, conditional pmf). *A pmf $p_{\mathsf{k}|\mathsf{x}}(k|x)$, $k \in \mathcal{K}$, $x \in \mathrm{GF}(2)$ is symmetric if there exists an involution $\pi : \mathcal{K} \to \mathcal{K}$ such that for any $k \in \mathcal{K}$ and $\neg k \triangleq \pi(k)$*

$$p_{\mathsf{k}|\mathsf{x}}(k|+1) = p_{\mathsf{k}|\mathsf{x}}(\neg k|-1). \tag{4.3}$$

In this context an involution π on $\mathcal{K}$ is a permutation on $\mathcal{K}$ that is self inverse:

$$\pi(k_1) = k_2 \Longrightarrow \pi(k_2) = k_1 \qquad \forall k_1. \tag{4.4}$$

We use the unary $\neg$ operator on $\mathcal{K}$ to denote the inverse to an element according to the involution π. By (4.4), we have $\neg(\neg k) = \pi\big(\pi(k)\big) = k$. In terms of LLR values, we can see from (4.1) and (4.3) that label inversion corresponds to a sign inversion of the corresponding LLR value:

$$\mathrm{L}(\neg k) = -\mathrm{L}(k) = -k'. \tag{4.5}$$

Without loss of generality, we order labels according to their LLRs[1], i.e.,

$$k'_0 < k'_1 < \cdots < k'_{K-1}. \tag{4.6}$$

Using this ordering, we have

$$\pi : k \mapsto \mathrm{mod}_{K-1}(K-1-k). \tag{4.7}$$

All the sign-based definitions and results of Section 2.2.3 carry over to the labels in a one-to-one fashion, except that the message inversions based on the $-$ sign are replaced by the involution inversion $\neg$. Let $\neg_{\boldsymbol{b}}\boldsymbol{\mu}$ denote the involution inversion according to the signs of the GF(2) vector $\boldsymbol{b}$ of a second vector of the same length, i.e. $(\neg_{\boldsymbol{b}}\boldsymbol{\mu})_i = \neg\mu_i$ if $b_i = -1$ and $(\neg_{\boldsymbol{b}}\boldsymbol{\mu})_i = \mu_i$ if $b_i = +1$. Now, assuming symmetry as per Assumption 1 holds for LUT-based message passing, i.e.,

$$p_{\mathsf{L}|\mathsf{x}}(L|x) = p_{\mathsf{L}|\mathsf{x}}(\neg L|-x), \tag{4.8}$$

$$\overline{\Phi}_j^{(\ell)}(\neg_{\boldsymbol{b}}\boldsymbol{\mu}) = \neg_{\mathrm{par}\,\boldsymbol{b}}\overline{\Phi}_j^{(\ell)}(\boldsymbol{\mu}), \tag{4.9}$$

$$\Phi_i^{(\ell)}(\neg L, \neg\overline{\boldsymbol{\mu}}) = \neg\Phi_i^{(\ell)}(L, \overline{\boldsymbol{\mu}}), \tag{4.10}$$

the well-known proofs apply in an analogous way and it follows that the messages are iid and

[1] We once again use the argument that changing the order of labels (which is a bijective permutation) does not affect the mutual information $I(\mathsf{x}, \mathsf{k})$.

symmetric, where (2.76) is replaced by

$$p^{(\ell)}_{\mu|x}(\mu_e|-x_{n(e)}) = p^{(\ell)}_{\mu|x}(\neg\mu_e|x_{n(e)}). \tag{4.11}$$

Consequently, the VN-to-CN and CN-to-VN message pmfs $p^{(\ell)}_{\mu|x}(\mu|x)$ and $p^{(\ell)}_{\overline{\mu}|x}(\overline{\mu}|x)$ are iid and symmetric throughout the decoding process and thus, the use of density evolution assuming $x = +1$ is justified. The elements of the value sets $\mathcal{M}'_\ell$, $\overline{\mathcal{M}}'_\ell$ and $\mathcal{L}'$ are defined in coherence to (4.1),

$$L' = \log\frac{p_{L|x}(L|+1)}{p_{L|x}(L|-1)}, \quad \mu' = \log\frac{p^{(\ell)}_{\mu|x}(\mu|+1)}{p^{(\ell)}_{\mu|x}(\mu|-1)}, \quad \overline{\mu}' = \log\frac{p^{(\ell)}_{\overline{\mu}|x}(\overline{\mu}|+1)}{p^{(\ell)}_{\overline{\mu}|x}(\overline{\mu}|-1)}, \tag{4.12}$$

for all iterations ℓ. Note that even if the label sets $\mathcal{M}_\ell$ and $\overline{\mathcal{M}}_\ell$ are constant over ℓ, the message value sets change over the course of iterations, reflecting the evolution of message densities. Implementations of LUT decoders only deal with labels, however, the fact that the value sets evolve during the decoding process is accounted for by using different LUTs for each iteration in general. Realizing that LUTs act as quantizers on LLR values (cf. Section 4.1.4), this also explains the good performance that LUT decoders can achieve with very low resolutions: Rather than covering the entire dynamic range of message values over all iterations, separate LUT updates can be designed for different iterations, only covering the dynamic range of values for their corresponding iteration.

4.1.2 Sign-Magnitude Interpretation of Labels

In Definition 3, we introduced inverse elements to all labels based on a self inverse permutation. As pointed out in (4.5), label inversion corresponds to a sign inversion of the respective LLR. Furthermore, we established an ordering on the set of labels in terms of their LLR values, cf. (4.6) and (4.7). Combining all of the above allows us to infer information on the LLR value from the message label:

- For K even and a given label $k \in \mathcal{K} = \{0,\dots,K-1\}$, we can obtain the sign of the associated LLR k' from the sign of the label, which we define as follows:

$$\operatorname{sign}(k') = \operatorname{sign}(k) \triangleq \begin{cases} -1, & k < K/2 \\ +1, & k \geq K/2 \end{cases} \tag{4.13}$$

- For $k_1, k_2 \in \mathcal{K}$, we can compare two messages in terms of their LLR

$$k_1 \lesseqgtr k_2 \quad \Longleftrightarrow \quad k'_1 \lesseqgtr k'_2 \tag{4.14}$$

- Finally, we can move back and forth between the conventional label representation and a sign-magnitude based representation with magnitude $|\cdot| : \{0,\ldots,K-1\} \to \{0,\ldots,K/2-1\}$

$$|k| \triangleq \begin{cases} k, & k < K/2, \\ K-1-k, & k \geq K/2, \end{cases} \tag{4.15}$$

and sign defined (4.13).

4.1.3 Product Distributions

At the VNs, tuples $(L,\overline{\mu})$ of incoming messages are combined into outgoing messages by the update rules (2.50); similarly, at the CNs a tuple μ is processed by the update rule (2.51). For discrete density evolution, the conditional pmf of the outgoing messages simply follows by collecting the probability mass of the preimage of the outgoing messages, i.e.,

$$p^{(\ell)}_{\mathsf{\mu}|\mathsf{x}}(\mu|x) = \sum_{(L,\overline{\mu}):\, \Phi^{(\ell)}(L,\overline{\mu})=\mu} p^{(\ell)}_{\mathsf{L},\overline{\mathsf{\mu}}|\mathsf{x}}(L,\overline{\mu}|x), \tag{4.16}$$

$$p^{(\ell+1)}_{\overline{\mathsf{\mu}}|\mathsf{x}}(\overline{\mu}|x) = \sum_{\mu:\, \overline{\Phi}^{(\ell)}(\mu)=\overline{\mu}} p^{(\ell)}_{\mathsf{\mu}|\mathsf{x}}(\mu|x). \tag{4.17}$$

Consequently, the product pmfs $p^{(\ell)}_{\mathsf{L},\overline{\mathsf{\mu}}|\mathsf{x}}(L,\overline{\mu}|\, x)$ and $p^{(\ell)}_{\mathsf{\mu}|\mathsf{x}}(\mu|x)$ of incoming messages are essential to discrete density evolution. For iid input messages distributed according to $p_{\overline{\mathsf{\mu}}|\mathsf{x}}(\overline{\mu}|x)$ and $p_{\mathsf{\mu}|\mathsf{x}}(\mu|x)$, the product distributions have first been derived in [59]:

$$p_{\mathsf{L},\overline{\mathsf{\mu}}|\mathsf{x}}(L,\overline{\mu}|x) = p_{\mathsf{L}|\mathsf{x}}(L|\, x)\prod_{m=1}^{i-1} p_{\overline{\mathsf{\mu}}|\mathsf{x}}(\overline{\mu}_m|x), \tag{4.18}$$

$$p_{\mathsf{\mu}|\mathsf{x}}(\mu|x) = 2^{2-j} \sum_{\boldsymbol{x}:\operatorname{par}\boldsymbol{x}=x} \prod_{n=1}^{j-1} p_{\mathsf{\mu}|\mathsf{x}}(\mu_n|x_n). \tag{4.19}$$

The product distributions can be shown to exhibit the following symmetry properties, which will turn out to be useful for LUT design in Section 4.2.

Lemma 4 (Symmetry of product pmfs). *For any $\boldsymbol{b} \in \mathrm{GF}(2)^{j-1}$ and iid input messages with symmetric pmfs* (4.11)*, the product distributions* (4.18) *and* (4.19) *are symmetric according to*

$$p_{\mathsf{L},\overline{\mathsf{\mu}}|\mathsf{x}}(\neg L, \neg\overline{\mu}|x) = p_{\mathsf{L},\overline{\mathsf{\mu}}|\mathsf{x}}(L,\overline{\mu}|-x), \tag{4.20}$$

$$p_{\mathsf{\mu}|\mathsf{x}}(\neg_{\boldsymbol{b}}\mu|x) = p_{\mathsf{\mu}|\mathsf{x}}(\mu|x \operatorname{par}\boldsymbol{b}). \tag{4.21}$$

Proof: (4.20) follows trivially from (4.18) using the symmetry (4.11) of the individual pmfs. To show (4.21), note that

$$\begin{aligned} p_{\mu|\mathsf{x}}(\neg_{\boldsymbol{b}}\boldsymbol{\mu}|x) &= 2^{2-j} \sum_{\boldsymbol{x}:\,\mathrm{par}\,\boldsymbol{x}=x} \prod_{n=1}^{j-1} p_{\mu|\mathsf{x}}(\mu_n|b_n\; x_n) = 2^{2-j} \sum_{\boldsymbol{x}':\,\mathrm{par}(\boldsymbol{x}'\odot\boldsymbol{b})=x} \prod_{n=1}^{j-1} p_{\mu|\mathsf{x}}(\mu_n|\; x'_n) \\ &= 2^{2-j} \sum_{\boldsymbol{x}':\,\mathrm{par}\,\boldsymbol{x}'\,\mathrm{par}\,\boldsymbol{b}=x} \prod_{n=1}^{j-1} p_{\mu|\mathsf{x}}(\mu_n|\; x'_n) = p_{\boldsymbol{\mu}|\;\mathsf{x}}(\boldsymbol{\mu}|x\,\mathrm{par}\,\boldsymbol{b}) \end{aligned}$$ ■

By combining (4.8) to (4.10), (4.16) and (4.17) and Lemma 4, an induction proof for Lemma 1 (with (2.76) replaced by (4.11)) can also be given in terms of the product distributions.

4.1.4 LLRs of Product pmfs - Relating LUT decoding and BP

A relationship between LUT decoding and BP can be established by calculating the LLR values of tuples that are incident to the node updates. By combining (4.12), (4.12), (4.18) and (4.19), the LLR values of the tuples can be expressed in terms of the individual message LLRs.

Theorem 9. *For a tuple of CN-to-VN message labels $\overline{\boldsymbol{\mu}} = (\overline{\mu}_1,\dots,\overline{\mu}_{i-1})$ and channel output L, the LLR value of the combination is given by*

$$\mathrm{L}(L,\overline{\boldsymbol{\mu}}) = \log\frac{p_{\mathsf{L},\overline{\boldsymbol{\mu}}|\mathsf{x}}(L,\overline{\boldsymbol{\mu}}|+1)}{p_{\mathsf{L},\overline{\boldsymbol{\mu}}|\mathsf{x}}(L,\overline{\boldsymbol{\mu}}|-1)} = L' + \sum_{m=1}^{i-1}\overline{\mu}'_m. \tag{4.22}$$

For a tuple of VN-to-CN message labels $\boldsymbol{\mu} = (\mu_1,\dots,\mu_{j-1})$, the LLR of the combination is given by

$$\mathrm{L}(\boldsymbol{\mu}) = \log\frac{p_{\boldsymbol{\mu}|\mathsf{x}}(\boldsymbol{\mu}|+1)}{p_{\boldsymbol{\mu}|\mathsf{x}}(\boldsymbol{\mu}|-1)} = 2\,\mathrm{atanh}\left(\prod_{n=1}^{j-1}\tanh\left(\frac{\mu'_n}{2}\right)\right). \tag{4.23}$$

Proof: Eqn. (4.22) follows trivially from (4.18) and (4.12).

To prove (4.23), let $\boldsymbol{x} = (x_1,\dots,x_{j-1}) \in \mathrm{GF}(2)^{j-1}$ and let the index set $\mathcal{J}(\boldsymbol{x},\boldsymbol{\mu}) = \{n_1,\dots,n_k\}$ be defined such that sign $\mu_n = x_n$ for $n \in \mathcal{J}(\boldsymbol{x},\boldsymbol{\mu})$ and sign $\mu_n \neq x_n$ for $j \notin \mathcal{J}$. Without loss of generality, assume $p_{\mu|\mathsf{x}}(\mu|\,\mathrm{sign}\,\mu) \geq p_{\mu|\mathsf{x}}(\mu|-\mathrm{sign}\,\mu)$, then by (4.12),

$$|\mu'| = \log\frac{p_{\mu|\mathsf{x}}(\mu|\,\mathrm{sign}\,\mu)}{p_{\mu|\mathsf{x}}(\mu|-\mathrm{sign}\,\mu)} \tag{4.24}$$

and hence

$$\prod_{n=1}^{j-1} p_{\mu'|\;\mathsf{x}}(\mu'_n|x_n) = \exp\left(\sum_{n\in\mathcal{J}(\boldsymbol{x},\boldsymbol{\mu})}|\mu'_n|\right)\prod_{n=1}^{j-1} p_{\mu'|\;\mathsf{x}}(\mu'_n|-\mathrm{sign}\,\mu'_n). \tag{4.25}$$

Consequently, due to (4.2), we can rewrite (4.19) as

$$p_{\boldsymbol{\mu}|\mathsf{x}}(\boldsymbol{\mu}|x) = C\sum_{\boldsymbol{x}\in\mathcal{P}_x}\exp\left(\sum_{n\in\mathcal{J}(\boldsymbol{x},\boldsymbol{\mu})}|\mu'_n|\right), \tag{4.26}$$

where the constant C depends on μ but not on x, and $\mathcal{P}_x \triangleq \{\boldsymbol{x} \mid \operatorname{par} \boldsymbol{x} = x\}$. Thus, the LLR of μ can be computed as

$$\mathrm{L}(\mu) = \log \frac{\sum_{\boldsymbol{x}\in\mathcal{P}_{+1}} \exp\left(\sum_{n\in\mathcal{J}(\boldsymbol{x},\mu)} |\mu'_n|\right)}{\sum_{\boldsymbol{x}\in\mathcal{P}_{-1}} \exp\left(\sum_{n\in\mathcal{J}(\boldsymbol{x},\mu)} |\mu'_n|\right)}. \tag{4.27}$$

From (4.27), we can already see that the magnitude of $\mathrm{L}(\mu)$ only depends on the magnitude of the elements of μ' and the sign is determined by the parity of μ'. Thus, it is sufficient to consider the case where μ' has positive parity. For $k \in \{1,\dots,j-1\}$, let us define

$$\Sigma_k = \sum_{C\in\mathcal{C}_k^J} \exp\left(\sum_{j\in C} |\mu'_j|\right), \tag{4.28}$$

where $\mathcal{C}_k^{j-1}$ denotes the set of all $\binom{j-1}{k}$ message index combinations of length k. Then (4.27) becomes

$$\log \frac{\Sigma_{j-1} + \Sigma_{j-3} + \dots + \Sigma_2 + 1}{\Sigma_{j-2} + \Sigma_{j-4} + \dots + \Sigma_1} \tag{4.29}$$

for j odd and

$$\log \frac{\Sigma_{j-1} + \Sigma_{j-3} + \dots + \Sigma_2}{\Sigma_{j-2} + \Sigma_{j-4} + \dots + \Sigma_1 + 1} \tag{4.30}$$

for j even, which we can combine to

$$\mathrm{L}(\mu) = \log \frac{\prod_{n=1}^{j-1}\left(e^{|\mu'_n|}+1\right) + \prod_{n=1}^{j-1}\left(e^{|\mu'_n|}-1\right)}{\prod_{n=1}^{j-1}\left(e^{|\mu'_n|}+1\right) - \prod_{n=1}^{j-1}\left(e^{|\mu'_n|}-1\right)} = 2\operatorname{atanh}\left(\prod_{n=1}^{j-1} \tanh\left(\frac{|\mu'_n|}{2}\right)\right). \tag{4.31}$$

Together with the aforementioned sign inversion property of (4.27), this proves the claim. ■

By comparing (2.59) and (4.22) as well as (2.61) and (4.23), we can see that Theorem 9 establishes a link between discrete message passing and BP decoding. Tuples of input labels are combined by LUT updates, corresponding to a quantization in the LLR value domain. E.g., let $\Phi^{-1}(\mu) \subset \mathcal{L} \times \overline{\mathcal{M}}^{i-1}$ be the preimage of some updated message μ under the VN update Φ and let $\Phi'^{-1}(\mu)$ be the set of LLRs corresponding to the preimage based on (4.22). Then applying the LUT Φ on the labels, all LLRs in $\Phi'^{-1}(\mu)$ are quantized to

$$\mathrm{L}(\mu) = \mathrm{L}\left(\Phi(L,\overline{\mu})\right) = \log \frac{\sum_{(L,\overline{\mu})\in\Phi^{-1}(\mu)} p_{\mathsf{L},\overline{\boldsymbol{\mu}}|\mathsf{x}}(L,\overline{\mu}|+1)}{\sum_{(L,\overline{\mu})\in\Phi^{-1}(\mu)} p_{\mathsf{L},\overline{\boldsymbol{\mu}}|\mathsf{x}}(L,\overline{\mu}|-1)}. \tag{4.32}$$

Thus, LUT decoding is essentially a form of quantized BP. The discrete structure of labels and LUTs allows for very simple, low resolution implementations, while essentially acting as iteration-dependent LLR quantizers, given that we use different update mappings for each

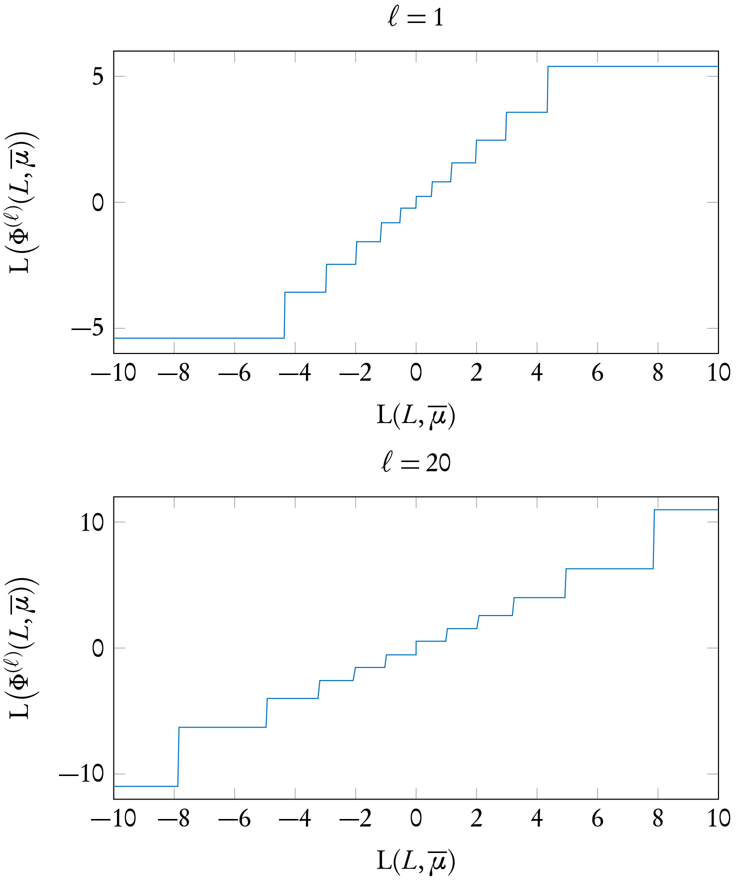


Figure 4.1: VN LUTs acting as LLR quantizers at the beginning ($\ell = 1$) and towards the end ($\ell = 20$) of the decoding of a $(3,6)$ LDPC code over a BI-AWGN channel with $\sigma = .84$ and information-optimal LUTs (cf. Section 4.2) with $|\mathcal{M}_\ell| = |\overline{\mathcal{M}_\ell}| = 12\ \forall \ell$. Notice that both quantizer input and output LLR values are larger towards the end of decoding.

iteration, cf. Figure 4.1.

Without using LUTs to constraint the size of message sets, discrete message passing is equivalent to BP decoding starting with a discrete set of channel outputs:

Example 1. *Consider a concatenation of a BI-AWGN channel with $\sigma = .84$ and a 2 bit quantizer, followed by a discrete decoder for a $(3,6)$ LDPC code. The channel output quantizer is designed to maximize the mutual information between channel input and quantizer output [109], producing the conditional pmf $p_{\mathsf{L}|\mathsf{x}}(L|+1) = \begin{pmatrix} 0.0216 & 0.0953 & 0.2429 & 0.6402 \end{pmatrix}$ for $L = \begin{pmatrix} 0 & 1 & 2 & 3 \end{pmatrix}$ and $p_{\mathsf{L}|\mathsf{x}}(L|-1) = p_{\mathsf{L}|\mathsf{x}}(3-L|+1)$, cf. (4.7). According to (4.12), $L' = \begin{pmatrix} -3.39 & -0.94 & 0.94 & 3.39 \end{pmatrix}$. Initially, there are no CN-to-VN messages and thus $\mathcal{M}_0 = \mathcal{L}$. For the first CN update, (4.23) yields $|\overline{\mathcal{M}}_1| = 12$ distinct output message values. Plugging combinations of those into (4.23), we obtain $|\mathcal{M}_1| = 292$ different VN-to-CN message values. Figure 4.2 illustrates the evolution of message pmfs and the rapid growth in the number of labels.*

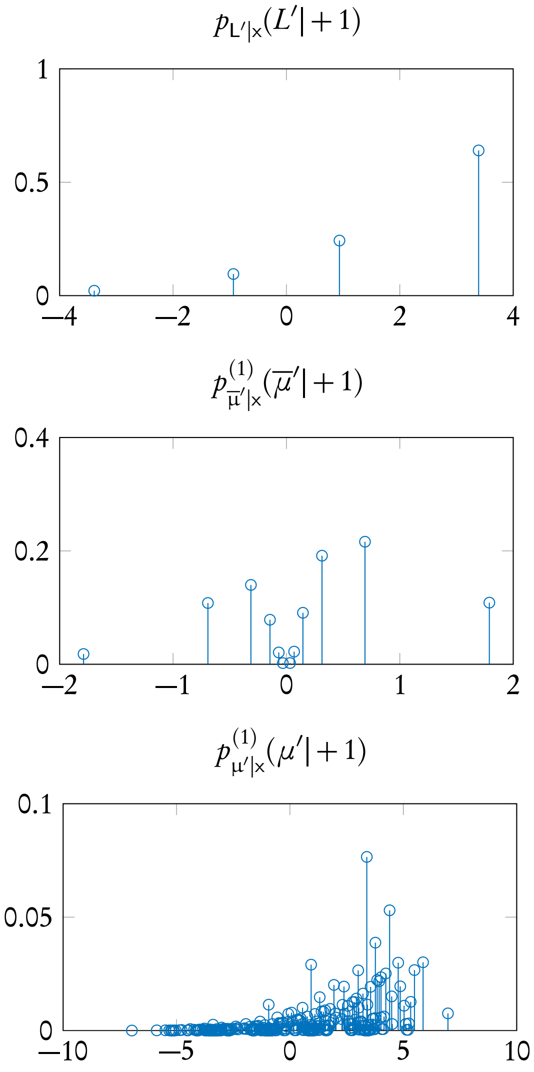


Figure 4.2: Evolution of message pmfs without using a LUT to constrain the size of the message sets. Starting from 4 messages, we already have 292 messages after one CN and one VN update.

4.2 LUT based Node Updates

In the previous section, we established some basic algebraic structure for the discrete messages of LUT decoding and derived a fundamental relationship between discrete message passing and BP. In this context, LUT decoding can be regarded as BP, where after each node update, the set of output LLRs is quantized to a smaller set by using a LUT.

In this section, we focus on the design of such LUTs, following the approach of [58, 59] (cf. Section 2.1.4 and Algorithm 3) to construct LUTs that maximize the local information flow through the code's factor graph. However, instead of just applying Algorithm 3, we simplify the LUT design by exploiting the algebraic relations we established in Section 4.1.2, while also ensuring that the symmetries of Assumption 1 are guaranteed by the LUTs. Consequently, we are able to reduce LUT complexity and show that it is even possible to replace CN LUTs by an algebraic update rule applied to the labels. We conclude this section by introducing the concept of LUT trees to further reduce LUT size and make LUTs applicable to high-degree nodes.

4.2.1 Mutual Information based LUT Design

We are interested in finding LUT node updates that maximize the flow of relevant information, i.e., the mutual information between messages and code bits. More precisely, an information-optimal LUT maximizes

$$\Phi^\star = \operatorname*{arg\,max}_{\Phi\in\mathcal{Q}} I\big(\Phi(\mathsf{y});\mathsf{x}\big), \tag{4.33}$$

where the LUT inputs $\mathsf{y}\in\mathcal{Y}$ depend on the type of the node update. E.g., when designing $\Phi_i^{(\ell)}$, i.e. the VN update of degree i for iteration ℓ, we associate $\mathcal{Y}=\mathcal{L}\times\overline{\mathcal{M}}_\ell^{i-1}$. We considered this problem in Section 2.1.4, where it was shown that for binary relevance, an optimum solution can be obtained by Algorithm 3 with complexity $O\big(|\mathcal{Y}|^3\big)$.

In what follows, we will further restrict the search space of mappings to symmetric quantizers,

$$\Phi(\neg y)=\neg\Phi(y), \qquad \forall y\in\mathcal{Y}, \tag{4.34}$$

and show that this is sufficient to arrive at a symmetric message passing scheme according to Assumption 1. Too see this, we need to associate $\mathcal{Y}$ with the set of message tuples and identify the inverse elements of tuples in terms of Definition 3 and the product pmfs (4.18) and (4.19). This can be done by utilizing the symmetries of Lemma 4. For the VN updates, $\mathcal{Y}=\mathcal{L}\times\overline{\mathcal{M}}^{i-1}$ and according to (4.20), the inverse element to $y=(L,\overline{\mu})$ is $\neg y=(\neg L,\neg\overline{\mu})$, and thus, (4.10) follows from (4.34). For the CN LUTs, we first partition

$$\mathcal{Y}=\mathcal{M}^{j-1}=\bigcup_{|\mu|}\big(\mathcal{M}_{|\mu|}\cup\neg\mathcal{M}_{|\mu|}\big), \tag{4.35}$$

where $\mathcal{M}_{|\mu|}\subset\mathcal{M}^{j-1}$ denotes the set of all tuples with magnitudes $|\mu|$ and even parity and

$\neg\mathcal{M}_{|\mu|}$ is the equivalent set of odd parity vectors. Now note that due to (4.23), all tuples within $\mathcal{M}_{|\mu|}$ share the same LLR and can thus be merged into a single LUT input y. Conversely, the elements of $\neg\mathcal{M}_{|\mu|}$ can be merged into the corresponding inverse LUT input $\neg y$. Indeed, it follows from (4.21) that $p(\mathcal{M}_{|\mu|}|+1) = p(\neg\mathcal{M}_{|\mu|}|-1)$ so that (4.9) is guaranteed by applying symmetric quantizers (4.34) to the merged events $\mathcal{M}_{|\mu|}$ and $\neg\mathcal{M}_{|\mu|}$. Implementation wise, this corresponds to a preprocessing of input messages according to (4.13) and (4.15), i.e., separating sign and magnitude of the tuple prior to the CN LUT updates. For both VN and CN product pmfs, label inversion corresponds to LLR value sign inversion, cf. (4.5), (4.22) and (4.23). Moreover, the (merged) product pmfs (4.18) and (4.23) are symmetric, and thus we can assume $p(\neg y|x) = p(y|-x)$ for designing a symmetric quantizer (4.34).

Considering the above and denoting the preimage of some $z \in \mathcal{Z}$ as $\mathcal{Y}_z \subset \mathcal{Y}$, we can develop the mutual information as follows:

$$I(\mathsf{x};\mathsf{z}) = \sum_x p_{\mathsf{x}}(x) \sum_z p_{\mathsf{z}|\mathsf{x}}(z|x) \log_2 \frac{p_{\mathsf{z}|\mathsf{x}}(z|x)}{p_{\mathsf{z}}(z)} = \sum_{\operatorname{sign} z>0} i_z, \tag{4.36}$$

where the partial mutual information (2.35) is given by

$$i_z \triangleq \sum_{\mathcal{Y}_z} p_{\mathsf{y}|\mathsf{x}}(y|1) \log_2 \frac{\sum_{y\in\mathcal{Y}_z} p_{\mathsf{y}|\mathsf{x}}(y|1)}{\sum_{y\in\mathcal{Y}_z} p_{\mathsf{y}}(y)} + \sum_{\mathcal{Y}_z} p_{\mathsf{y}|\mathsf{x}}(\neg y|1) \log_2 \frac{\sum_{y\in\mathcal{Y}_z} p_{\mathsf{y}|\mathsf{x}}(\neg y|1)}{\sum_{y\in\mathcal{Y}_z} p_{\mathsf{y}}(y)}. \tag{4.37}$$

Due to the symmetry constraints, the quantizer design is reduced to the positive domain $\operatorname{sign} z > 0$, effectively halving the number of quantizer input and output labels. Consequently, we can use a reduced complexity variant of Algorithm 3 with a complexity reduction from $O(|\mathcal{Y}|^3)$ to $O(|\mathcal{Y}|^3/8)$ due to $|\mathcal{Y}| \to |\mathcal{Y}|/2$ for the symmetric case. Furthermore, the implementation complexity of a symmetric LUT is only half of the unsymmetric case, because for any Φ and $y \in \mathcal{Y}$, we can use a mapping Φ^+ which is restricted to the positive domain of Φ to arrive at the corresponding output by separating the input into sign and magnitude,

$$z = \operatorname{sign} y\ \Phi^+(|y|). \tag{4.38}$$

Lastly, note that due to the merges of the VN-to-CN message tuples μ into events $\mathcal{M}_{|\mu|}$ and $\neg\mathcal{M}_{|\mu|}$, the size of the CN LUTs is reduced even further by preprocessing input tuples.

4.2.2 The min-LUT Algorithm

In the previous section, we discribed how the complexity of CN LUTs can be reduced by the algebraic structure and symmetry we introduced in Section 4.1. Based on this structure, it is even possible to completely eliminate CN LUTs. Specifically, (4.13) to (4.15) are sufficient to use the MS update rule (2.63) based on only the labels, completely eliminating the need for a CN LUT. This hybrid min-LUT decoding algorithm has first been introduced in [6, 67] for regular codes. Especially for high-rate regular codes, the CN degree d_c is substantially larger than the VN degree, resulting in higher implementation complexity; E.g., for the 10GBaseT LDPC code [53], $d_v = 6$ and $d_c = 32$. In Section 4.4, we present a comparison between purely LUT based decoders and min-LUT decoders, highlighting the excellent performance-complexity tradeoff of the min-LUT approach.

4.2.3 LUT Trees

So far, we discussed LUTs that combine all input messages at once, i.e., for VNs, we identified $\mathcal{Y} = \mathcal{L} \times \overline{\mathcal{M}}_\ell^{i-1}$ and for CNs $\mathcal{Y} = \mathcal{M}_\ell^{j-1}$ as the LUT input sets. However, with increasing node degrees, the size of the input sets grows exponentially. E.g., for $|\mathcal{M}| = 8$, i.e., a message resolution of 3 bit, and 8 incoming messages, there are already more than 16 million input message combinations, which is prohibitively complex for practical purposes.

To overcome this problem, we restrict ourselves to nested update rules, e.g., for a VN of degree $i = 6$, a possible nesting could take the form

$$\Phi_6(L, \overline{\mu}_1, \ldots, \overline{\mu}_5) = \Phi_{6,0}\big(\Phi_{6,1}(\overline{\mu}_1, \overline{\mu}_2, \overline{\mu}_3), \Phi_{6,2}(\overline{\mu}_4, \overline{\mu}_5), L\big). \tag{4.39}$$

Any such nesting can be represented graphically by a directed tree, where the leaf nodes correspond to the LUT inputs and the inner nodes represent intermediate LUTs. The particular nesting (4.39) corresponds to tree T_2 in Figure 4.3. Since we assume iid messages, the ordering of the arguments in the nesting is immaterial and we consider nestings that differ only in the ordering as equivalent. Recalling the LLR value quantization effect of LUTs, cf. (4.32) and Figure 4.1, intermediate VN LUT updates can be thought of as a sum-and-quantize operation. Likewise, intermediate CN LUT updates correspond to a boxplus-and-quantize operation, cf. [42]. Furthermore, note that the symmetry of intermediate mappings implies the symmetry of the overall mappings.

Lemma 5. *Let Φ be a nested mapping represented by a LUT tree T with J non-leaf nodes and let $\mathcal{Q} = \{\Phi_1, \ldots, \Phi_J\}$ denote the set of all constituent mappings. Then the following propositions are true:*

1. *If all $\Phi_j \in \mathcal{Q}$ fulfil the symmetry condition* (4.10), *then so does the overall mapping* Φ.
2. *If all $\overline{\Phi}_j \in \mathcal{Q}$ fulfil the symmetry condition* (4.9), *then so does the overall mapping* $\overline{\Phi}$.

Proof: We prove the proposition for the simple case of a binary nesting, i.e., $\Phi(\mu_1, \mu_2) = \Phi_0\big(\Phi_1(\mu_1), \Phi_2(\mu_2)\big)$, and note that the proof extends straightforwardly to any nesting. If Φ_0, Φ_1 and Φ_2 fulfill (4.10), then with $\mu^{\mathrm{T}} = \begin{pmatrix} \mu_1^{\mathrm{T}} & \mu_2^{\mathrm{T}} \end{pmatrix}$,

$$\begin{aligned} \Phi(\neg\mu) &= \Phi(\neg\mu_1, \neg\mu_2) = \Phi_0\big(\Phi_1(\neg\mu_1), \Phi_2(\neg\mu_2)\big) \\ &= \Phi_0\big(\neg\Phi_1(\mu_1), \neg\Phi_2(\mu_2)\big) = \neg\Phi_0\big(\Phi_1(\mu_1), \Phi_2(\mu_2)\big) \\ &= \neg\Phi(\mu). \end{aligned}$$

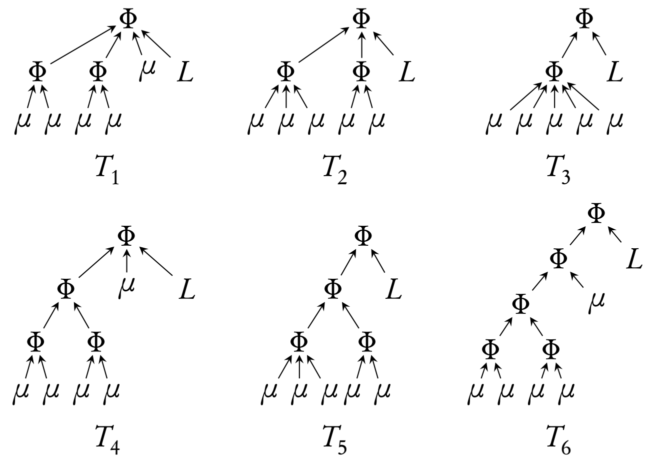


Figure 4.3: Six different LUT tree structures. Note that $T_1 \geq_{\mathcal{T}} T_4 \geq_{\mathcal{T}} T_6$, $T_2 \geq_{\mathcal{T}} T_5$, $T_3 \geq_{\mathcal{T}} T_5$, and $T_3 \geq_{\mathcal{T}} T_6$. However, we cannot compare T_2 with T_3 or T_5 with T_6 using the relation $\geq_{\mathcal{T}}$.

Similarly, if $\overline{\Phi}_0$, $\overline{\Phi}_1$ and $\overline{\Phi}_2$ fulfill (4.9), then with $\mu^{\mathrm{T}} = \big(\mu_1^{\mathrm{T}} \quad \mu_2^{\mathrm{T}})\big)$ and $\boldsymbol{b}^{\mathrm{T}} = \big(\boldsymbol{b}_1^{\mathrm{T}} \quad \boldsymbol{b}_2^{\mathrm{T}}\big)$,

$$\begin{aligned}
\overline{\Phi}(\neg_{\boldsymbol{b}}\mu) &= \overline{\Phi}(\neg_{\boldsymbol{b}_1}\mu_1, \neg_{\boldsymbol{b}_2}\mu_2) \\
&= \overline{\Phi}_0\big(\overline{\Phi}_1(\neg_{\boldsymbol{b}_1}\mu_1), \overline{\Phi}_2(\neg_{\boldsymbol{b}_2}\mu_2)\big) \\
&= \overline{\Phi}_0\big(\neg_{\mathrm{par}\,\boldsymbol{b}_1}\,\overline{\Phi}_1(\mu_1), \neg_{\mathrm{par}\,\boldsymbol{b}_2}\,\overline{\Phi}_2(\mu_2)\big) \\
&= \neg_{\mathrm{par}\,\boldsymbol{b}_1}\neg_{\mathrm{par}\,\boldsymbol{b}_2}\,\overline{\Phi}_0\big(\Phi_1(\mu_1), \overline{\Phi}_2(\mu_2)\big) \\
&= \neg_{\mathrm{par}\,\boldsymbol{b}}\,\overline{\Phi}(\mu) \qquad \blacksquare
\end{aligned}$$

While the nested structure clearly reduces complexity, it is not clear a priori which tree structures are preferable. In what follows, we provide guidelines on how to choose the tree structure based on information-theoretic arguments and a heuristic metric as well as practical considerations. For the moment, we do not distinguish between messages $\overline{\mu}$ and channel input L; the discussion of the location of L within the tree is deferred to the end of this subsection.

Partial ordering

Let $\mathcal{Q}_1$ and $\mathcal{Q}_2$ denote the set of all LUTs that respect the nesting induced by some trees T_1 and T_2. We call the tree T_2 a refinement of T_1 iff $\mathcal{Q}_1 \supseteq \mathcal{Q}_2$. Graphically, a refined tree is derived from the original by the placement of new inner nodes between parent and child nodes. Thus,

$$\max_{\Phi \in \mathcal{Q}_1} I\big(\Phi(\mathsf{L}, \overline{\boldsymbol{\mu}}); \mathsf{x}\big) \geq \max_{\Phi \in \mathcal{Q}_2} I\big(\Phi(\mathsf{L}, \overline{\boldsymbol{\mu}}); \mathsf{x}\big).$$

Consequently, tree refinement defines a partial ordering $\geq_{\mathcal{T}}$, effectively inducing a hierarchy in terms of maximum information flow. However, since the totality axiom is not fulfilled, not all tree structures can be compared in terms of the relation $\geq_{\mathcal{T}}$, cf. Figure 4.3.

T	T_1	T_2	T_3	T_4	T_5	T_6
δ	10	11	11	14	16	19
σ^*	0.5330	0.5328	0.5327	0.5313	0.5309	0.5305

Table 4.1: Comparison of cumulative depth and density evolution threshold for various tree structures (cf. Figure 4.3) for the regular (6, 32) ensemble. Here, all LUTs had a resolution of 3 bit.

A heuristic metric

The data processing inequality states that processing can only reduce mutual information. Therefore, for maximum information flow, the paths from the input leaves to the root output should be as short as possible. We thus define the cumulative depth $\delta(T)$ of a tree T as the sum of distances of all leaf nodes to the root node. Density evolution simulations confirm that cumulative depth is useful in ranking tree structures. Table 4.1 shows how a larger δ corresponds with a lower threshold. However, the threshold differences are small and our simulations have shown that all the trees presented here perform similar in terms of error rate.

Implementation Perspective

From an implementation point of view, binary trees are most attractive because they allow for the smallest LUTs. Previous work [59, 60, 80] considered only what we refer to as *maximum-height* binary trees, cf. Figure 4.4a. This type of trees are attractive for software implementations of LUT decoding, as the tree structure can be handled by a simple loop. However, in terms of the metric δ, *maximum-width* trees are to be preferred, cf. Figure 4.4b. Indeed, we found in density evolution threshold simulations as well as in BER simulations that maximum-width trees perform slightly better. Moreover, maximum-width trees are more parallelizable because of their smaller height, which is especially true for larger node degrees. The tree height scales logarithmic in the number of inputs, whereas the height of maximum-height trees grows linearly with the number of inputs. Consequently, these trees allow for shorter path delays and faster decoders [33] and thus are generally preferable to maximum-height trees. Due to these reasons, all density evolution and error rate simulation results presented in this chapter are carried out for binary, maximum-width LUT trees. Finally, note that no matter which binary tree is chosen, the number of LUTs is always equal to the number of tree inputs minus one. This is because any binary tree representing a nesting is a full binary tree.

Position of the Channel LLR

The mutual information between the CN-to-VN messages and the coded bits is initially zero and increases over the course of iterations until at some iteration $I(\overline{\mu}^{(\ell')};\mathsf{x}) \geq I(\mathsf{L};\mathsf{x})$. Using a similar argument as before, we can conclude that until iteration ℓ' the channel LLR should be placed close to the root node to ensure a large information flow. After iteration ℓ', the CN-to-VN messages tend to carry more information than the channel LLR and thus should

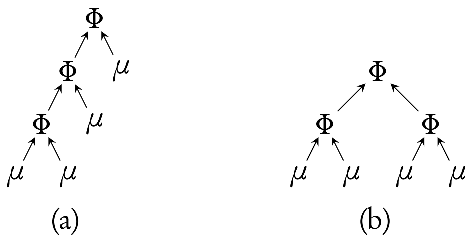


Figure 4.4: A maximum-height tree with $\mathscr{S} = 9$ (a) vs. a maximum width tree with $\mathscr{S} = 8$ (b) with 4 inputs

be placed closer to the root node. Our simulations show that this strategy indeed provides the best frame error rate (FER) performance; however, the loss as compared to the case where the channel LLR stays at the root node is only relevant for a large number of iterations ($\mathscr{L} > 20$).

4.3 LUTs for Irregular LDPC Codes

In this section, we focus on the analysis and design of information-optimal LUT decoders for irregular LDPC codes. For this scenario, multiple distinct LUTs $\Phi_i^{(\ell)} : \mathcal{L} \times \overline{\mathcal{M}}_\ell^{i-1} \to \mathcal{M}_\ell$ and $\overline{\Phi}_j^{(\ell)} : \mathcal{M}_\ell^{j-1} \to \overline{\mathcal{M}}_{\ell+1}$ must be designed for all degrees $i \in \mathcal{D}_\mathrm{v}$, $j \in \mathcal{D}_\mathrm{c}$ at any iteration ℓ. Due to the non trivial degree distributions (DDs) of irregular codes, an additional probabilistic dimension is introduced to the maximum mutual information LUT design. We take this into account by deriving a joint LUT design for irregular codes, which can be formulated as an instance of the symmetric quantizer optimization problem discussed in Section 4.2.1. In a second step, based on asymptotic stability analysis and numeric density evolution results, we argue that DDs optimized for conventional BP based algorithms [18, 77] are not well suited for LUT decoding and proceed to derive optimized DDs for LUT decoding.

4.3.1 Joint LUT Design

To shorten the discussion, we do not distinguish between VN and CN updates in what follows and use a generic notation with node updates Φ, degree indices $d \in \mathcal{D}$ and DDs $p_\mathrm{d}(d)$. A straightforward approach would be to optimize the mappings separately, i.e., we solve

$$\operatorname*{arg\,max}_{\Phi_d : \mathcal{Y}_d \to \mathcal{Z}} I\big(\mathsf{x}, \Phi_d(\mathsf{y}_d)\big) \tag{4.40}$$

over the probability space $\mathcal{X} \times \mathcal{Y}_d$ for any degree $d \in \mathcal{D}$ individually. However, this approach does not take into account the DD and hence, cannot be expected to maximize the quantity of interest,

$$\operatorname*{arg\,max}_{\Phi : \mathcal{Y} \to \mathcal{Z}} I\big(\mathsf{x}, \Phi(\mathsf{y})\big) \tag{4.41}$$

with joint distribution

$$p_{\mathsf{x},\mathsf{y}_d,\mathsf{d}}(x,y_d,d) = p_{\mathsf{x}}(x)p_{\mathsf{d}}(d)p_{\mathsf{y}_d|\,\mathsf{x},\mathsf{d}}(y_d|x,d). \tag{4.42}$$

In this setting, we identify pairs (y_d,d), i.e., the joint event that some degree d and a particular $y_d \in \mathcal{Y}_d$ occur as the quantizer inputs $\mathcal{Y}$. Thus, we can view $\Phi : \mathcal{Y} \to \mathcal{Z}$ as a $|\mathcal{D}|$-tuple of jointly optimized mappings $(\Phi_d)_{d\in\mathcal{D}}$, where each $\Phi_d : \mathcal{Y}_d \to \mathcal{Z}$ is itself an update rule for a certain node degree.

Due to the independence of the code degree and channel input (4.42), we have

$$\mathrm{L}(y_d,d) = \log\frac{p_{\mathsf{y}_d,\mathsf{d}|\mathsf{x}}(y_d,d|+1)}{p_{\mathsf{y}_d,\mathsf{d}|\mathsf{x}}(y_d,d|-1)} = \log\frac{p_{\mathsf{y}_d|\mathsf{x},\mathsf{d}}(y_d|+1,d)}{p_{\mathsf{y}_d|\mathsf{x},\mathsf{d}}(y_d|-1,d)} = \mathrm{L}(y_d). \tag{4.43}$$

Hence, any optimal symmetric quantizer $\Phi^\star$ for the optimization problem (4.41) with contiguous regions on LLRs of $(y_d,d) \in \mathcal{Y}$, can be broken down into a $|\mathcal{D}|$-tuple of quantizers $(\Phi^\star_d)_{d\in\mathcal{D}}$, each of which has contiguous quantization regions and is symmetric. Consequently, the symmetric quantizer design algorithm of Section 4.2 can be used to obtain a jointly optimal tuple of LUT updates. Note that the quantizer input LLRs (4.43) only depend on the conditional message distribution for a given degree $p_{\mathsf{y}_d|\mathsf{x},\mathsf{d}}(y_d|x,d)$ but are independent of the DD $p_{\mathsf{d}}(d)$. However, according to (4.42), the probability of the joint event (y_d,d) depends on the DD and thus, for different DDs, different LUTs are optimal in general. Care must be taken in the case of LUT trees, where the joint design procedure applies only to the design of the root node LUTs, as illustrated in the following example.

Example 2 (Joint LUT design according to (4.41))**.** *Consider a code with VN DD λ_d, $d \in \{2,3,4\}$ and tree structures as depicted in Figure 4.5. At iteration ℓ, the CN-to-VN message pmf $p^{(\ell)}_{\overline{\mu}|\mathsf{x}}$ is incident to the VNs, i.e., incident to the leafs of the trees. Now starting at the leaf nodes, for every VN degree $d \in \mathcal{D}$, the intermediate LUTs $\Phi_{d,k}$ are designed recursively, until arriving at the intermediate conditional pmfs $p_{\overline{\mu}|\mathsf{d},\mathsf{x}}$ that are incident to the root nodes $\Phi_{d,0}$.*

At this point the DD comes into play. Rather than solving (4.40) *and designing Φ_d separately, we are going to take into account the DD and perform a joint design according to* (4.41)*. To this end, we identify the inputs to the joint quantizers as a triple consisting of channel input L, intermediate message $\overline{\mu}$ and degree d with conditional pmf*

$$p^{(\ell)}_{\mathsf{L},\overline{\mu},\mathsf{d}|\mathsf{x}}(L,\overline{\mu},d|x) = p^{(\ell)}_{\overline{\mu}|\mathsf{d},\mathsf{x}}(\overline{\mu}|x)p_{\mathsf{L}|\mathsf{x}}(L|x)p_{\mathsf{d}}(d). \tag{4.44}$$

The performance difference between the individual design (4.40) and the joint design (4.41) can be significant. Figure 4.6 shows the density evolution thresholds for irregular LDPC codes with DDs from Table 4.2 obtained with the two LUT design strategies over a BI-AWGN channel. It is seen that the density evolution thresholds for the individual LUT design (4.40) are worse than those of the joint design (4.41) and even decrease for larger message alphabets (higher quantizer resolutions), thereby confirming the superiority of the joint LUT design.

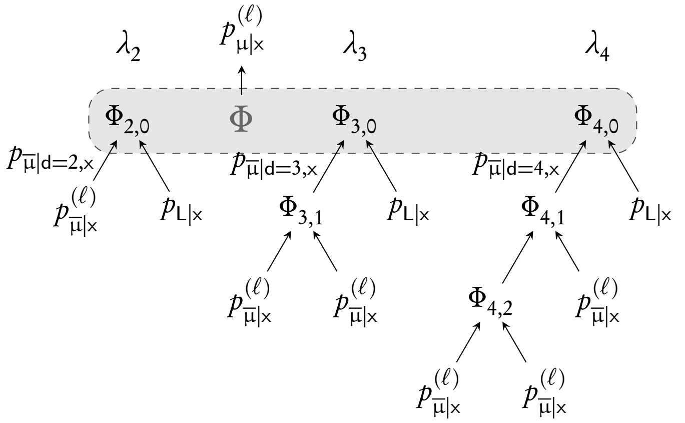


Figure 4.5: VN density evolution update for an irregular LDPC code using LUT trees

l	$\lambda_l(x)$	$\rho_l(x)$	$\sigma^\star_{\text{LUT}}$	$\sigma^\star_{\text{BP}}$
1	$0.16385x + 0.40637x^2 + 0.42978x^7$	$0.59105x^6 + 0.40876x^7 + 0.00019x^8$	0.89657	0.91775
2	$0.13805x + 0.40104x^2 + 0.02659x^8 + 0.43433x^{16}$	$0.32338x^7 + 0.67662x^8$	0.92919	0.95075
3	$0.30013x + 0.28395x^2 + 0.41592x^7$	$0.22919x^5 + 0.77081x^6$	0.583182	0.9497
4	$0.23802x + 0.20997x^2 + 0.03492x^3 + 0.12015x^4 +$ $0.01587x^6 + 0.00480x^{13} + 0.37627x^{14}$	$0.98013x^7 + 0.01987x^8$	0.603642	0.9622

Table 4.2: DD pairs (λ, ρ) for rate $\frac{1}{2}$ codes and density evolution thresholds for a BI-AWGN under LUT and BP decoding. The pairs 1 and 2 have been optimized for LUT decoding of quantized BI-AWGN channels using the method described in Section 4.3.3, pairs 3 and 4 are taken from [77] and have been optimized for BP decoding. For the LUT thresholds, a resolution of 4 bits and the min-LUT algorithm have been used.

Another take away point from Figure 4.6 is the performance of the BP-optimized ensemble (λ_4, ρ_4), cf. Table 4.2. Even with the joint design, the irregular ensemble performs worse than a regular $(3,6)$ code.

4.3.2 Degree Distributions for LUT Decoding

Another insight offered by Figure 4.6 is as follows. For the (λ_4, ρ_4) ensemble (taken from [77]), the threshold with jointly designed LUTs falls substantially short of the BP threshold even at high resolutions. This indicates that these ensembles are ill-suited for LUT decoding. Similar observations have been made in [93]. The authors of that paper also showed how irregular DDs can be designed to take into account and mitigate the effects of quantization in MS decoding.

In what follows, we set out to find DDs that are optimized for LUT decoding. To this end, special attention is paid to the degree 2 VNs. Due to the minimal extrinsic information they receive, they are the most difficult to decode, especially for quantized extrinsic information. For the case of BP decoding, the decisive role of degree 2 nodes has been quantified in [77] by means of an asymptotic stability analysis, providing upper bounds on the fraction of degree 2 edges (the probability of degree 2 edges in the (λ_4, ρ_4) ensemble from [77] is indeed fairly close to that bound). Since LUT decoding can be viewed as a degraded form of BP, it seems intuitive

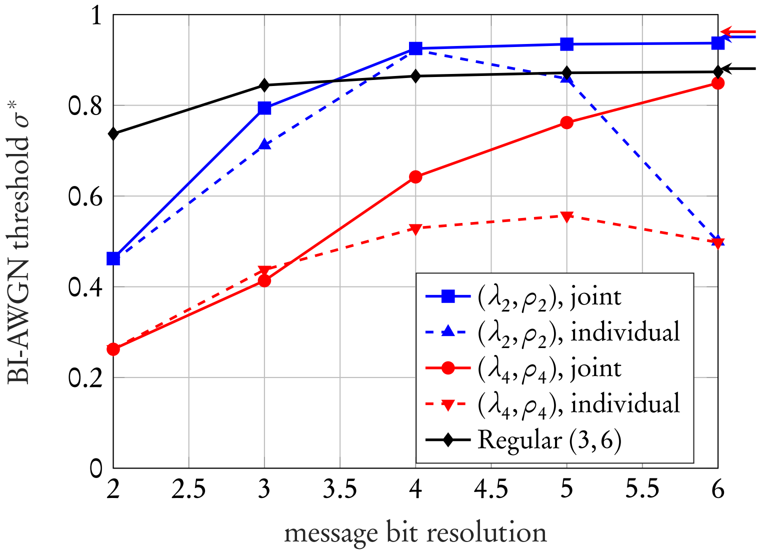


Figure 4.6: Density evolution thresholds versus message resolution for different LDPC ensembles (cf. Table 4.2) in a BI-AWGN channel with individual and joint LUT design. Arrows on the right margin indicate the corresponding BP thresholds.

that the fraction of degree 2 edges should be lower for LUT decoding. To quantify this line of reasoning, we consider a linearization of message pmfs around the fixed point Δ_∞ [77]. For erasure-type VN-to-CN messages with pmf $p_{\mu|x}^{(\ell)} = 2\epsilon\Delta_0 + (1-2\epsilon)\Delta_\infty$, we can then show that for too large of a fraction of degree 2 edges, the error probability is necessarily bounded away from 0, i.e., there exists $\xi > 0$ such that $\mathrm{P_e}(p_{\mu|x}^{(\ell)}) = \epsilon \geq \xi$.

Lemma 6. *Consider a DD (λ, ρ) and a symmetric channel pmf p_0. Furthermore, define*

$$r \triangleq -\lim_{n\to\infty} \frac{1}{n} \log \mathrm{P_e}(p_0^{(n)}), \tag{4.45}$$

where $p_0^{(n)} = \mathrm{Q}_\ell(p_0^{(n-1)} \otimes p_0)$ for some sequence of quantizers $\{\mathrm{Q}_\ell\}_{\ell=1}^n$. Then if $\lambda'(0)\rho'(1) > e^r$, there exists $\xi > 0$ such that $\mathrm{P_e}(p_{\mu|x}^{(\ell)}) \geq \xi$ for any erasure-type message pmf $p_{\mu|x}^{(\ell)}$.

Proof: We prove the proposition by contradiction, i.e., we show that if $p_{\mu|x}^{(\ell)}$ is a VN-to-CN erasure-type message pmf with arbitrarily small error probability ϵ, then under the conditions of the lemma, the error probability would increase over iterations, which is a contradiction. To see this, we assume that the mass at ∞ is not affected by quantization, and neglect the effects of CN quantization. Due to Theorem 9, without CN LUTs, one round of LUT decoding is identical to one round of BP decoding up to VN quantization. Since a symmetric VN quantizer (4.34) does not affect the error probability and for BP decoding, the error probability is non increasing and thus the claim follows. We first examine how $p_{\mu|x}^{(\ell)}$ evolves during a single iteration. Let $p_{\mu|x}^{(\ell)}$ be incident to a CN of degree j. We want to determine the output density and are only interested in terms that are at most linear in ϵ. Given the inputs 0 and ∞, the output of the BP CN update (2.61) is either ∞, if all $j-1$ inputs are ∞ or 0, if at least one or more inputs are 0. The event that more than one 0 occurs has probability proportional to $O(\epsilon^2)$ and can be neglected, so that for the pmf after the CN update

$$(1-2\epsilon)^{j-1}\Delta_\infty + \binom{j-1}{j-2} 2\epsilon\Delta_0 + O(\epsilon^2) = (j-1)2\epsilon\Delta_0 + \big(1-(j-1)2\epsilon\big)\Delta_\infty + O(\epsilon^2), \quad (4.46)$$

and averaged over CN degrees,

$$2\epsilon\rho'(1)\Delta_0 + \big(1-2\epsilon\rho'(1)\big)\Delta_\infty + O(\epsilon^2). \quad (4.47)$$

Now suppose this density is in turn incident to a VN of degree k and let p_0 denote the channel LLR density. At the output of the VN, we have

$$\begin{cases} 2\epsilon\rho'(1)p_0 + \big(1-\rho'(1)2\epsilon\big)\Delta_\infty + O(\epsilon^2), & k=2, \\ \Delta_\infty + O(\epsilon^2), & k>2, \end{cases}$$

and after averaging over all degrees, and quantizing we obtain (note that $\lambda'(0)=\lambda_2$)

$$2\epsilon\lambda'(0)\rho'(1)Q_1(p_0) + \big(1-2\epsilon\lambda'(0)\rho'(1)\big)\Delta_\infty + O(\epsilon^2). \quad (4.48)$$

Consequently, after n iterations, $p_{\mu|x}^{(\ell)}$ evolves to

$$p_{\mu|x}^{(\ell+n)} = 2\epsilon(\lambda'(0)\rho'(1))^n p_0^{(n)} + \big(1-2\epsilon\lambda'(0)\rho'(1)\big)^n\Delta_\infty + O(\epsilon^2).$$

Since, we assume that $\lambda'(0)\rho'(1) > e^r$ and the limit (4.45) exists, there exists ℓ such that

$$\begin{aligned} \mathrm{P_e}(p_{\mu|x}^{(\ell+n)}) &= 2(\lambda'(0)\rho'(1))^n\epsilon\mathrm{P_e}(p_0^{(n)}) \\ &> 2\epsilon + O(\epsilon^2) > \epsilon, \end{aligned}$$

for small enough ϵ. Thus $\mathrm{P_e}(p_{\mu|x}^{(\ell+n)}) > \mathrm{P_e}(p_{\mu|x}^{(\ell)})$, a contradiction, because the error probability is non increasing. ■

Note that for BP decoding, Lemma 6 can be extended from erasure-type message pmfs to arbitrary pmfs by using the optimality of the BP algorithm [77]. Unfortunately, the extension for the LUT decoding algorithm is not possible due to the lack of optimality. However, as we will see in what follows, the upper bound $\lambda'(0) = \lambda_2 \le \lambda_2^* \triangleq e^r/\rho'(1)$ is still a valuable criterion when searching for LUT-optimized DDs. For BP decoding (i.e., without quantizers), the decoding bound can be explicitly expressed as

$$r = -\lim_{n\to\infty}\frac{1}{n}\log\mathrm{P_e}(p_0^{\otimes n}) = \int_{\mathbb{R}} p_0(x)e^{-x/2}\,\mathrm{d}x$$

[77], whereas for LUT decoding, we have to resort to a numeric approximation. Figure 4.7

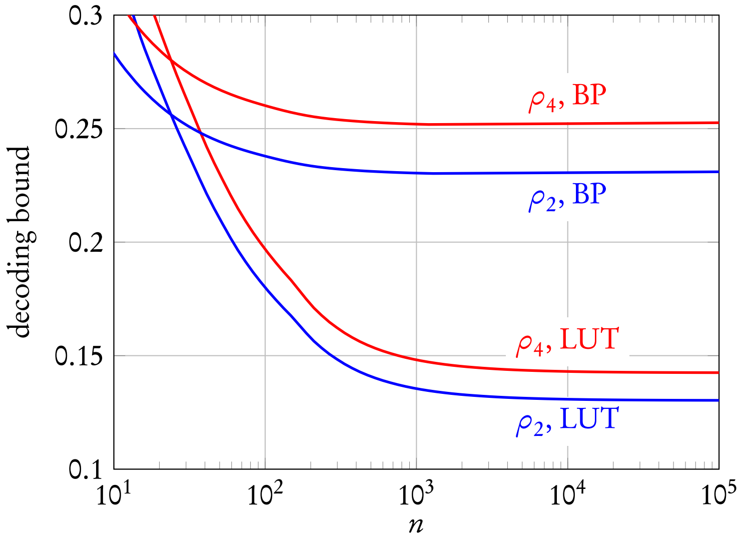


Figure 4.7: λ_2^* for different decoding algorithms and CN DDs (cf. Table 4.2) using resolution of 4 bits and a BI-AWGN channel with $\sigma = 0.929$.

shows the convergence behavior of the decoding bound for both decoder types. As expected, LUT decoding tolerates fewer degree 2 nodes. Furthermore, the limit can be determined numerically with about 10^4 iterations. In accordance to our initial reasoning, the bounds for LUT decoding turn out to be significantly lower than for BP decoding.

4.3.3 Numerical Optimization of Degree Distributions

For a target rate R, we are interested in a DD pair (λ, ρ), of maximum density evolution threshold under LUT decoding. We follow the hill climbing approach of [18], taking into account the bound $\lambda_2^\star$ derived in the previous section.

More specifically, we start with an initial DD with rate $R < R_t$ and iteratively solve linear programs maximizing the code rate (2.49) by alternating minimization of $\sum_j \frac{\rho_j}{j}$ and maximization if $\sum_i \frac{\lambda_i}{i}$. Additional linear constraints are necessary to limit the search space to the space of pmfs, to limit the step size, to ensure linearity and to guarantee that updated DDs improve in terms of error probability [18]. More over, for the VN DD we also impose $\lambda_2 \leq \lambda_2^*$. Without that constraint, the optimization procedure converges to low threshold DDs.

Summing up, the linear programs are given by

$$\begin{aligned}
&\max_{\lambda} \sum_{i\in\mathcal{D}_\mathrm{v}} \frac{\lambda_i}{i} \\
&\text{subject to} \\
&\quad \sum_{i\in\mathcal{D}_\mathrm{v}} \lambda_i = 1, \\
&\quad \lambda_i \geq 0, \quad i \in \mathcal{D}_\mathrm{v}, \\
&\quad \lambda_2 \leq \lambda_2^*, \\
&\quad \left| P_\ell - \sum_{i\in\mathcal{D}_\mathrm{v}} P_{\ell|i}\lambda_i \right| \leq \delta\left[P_{\ell-1} - P_\ell\right]^+ \quad \forall \ell, \\
&\quad \sum_{i\in\mathcal{D}_\mathrm{v}} P_{\ell|i}\lambda_i \leq P_{\ell-1} \quad \forall \ell,
\end{aligned} \tag{4.49}$$

and

$$\begin{aligned}
&\min_{\rho} \sum_{j\in\mathcal{D}_\mathrm{c}} \frac{\rho_j}{j} \\
&\text{subject to} \\
&\quad \sum_{j\in\mathcal{D}_\mathrm{c}} \rho_j = 1, \\
&\quad \rho_j \geq 0, \quad j \in \mathcal{D}_\mathrm{c}, \\
&\quad \left| Q_\ell - \sum_{j\in\mathcal{D}_\mathrm{c}} Q_{\ell|j}\rho_j \right| \leq \delta\left[Q_{\ell-1} - Q_\ell\right]^+ \quad \forall \ell, \\
&\quad \sum_{j\in\mathcal{D}_\mathrm{c}} Q_{\ell|j}\rho_j \leq Q_{\ell-1} \quad \forall \ell.
\end{aligned} \tag{4.50}$$

Here, the error probability traces $P_{\ell|i}$, $Q_{\ell|j}$ correspond to the VN-to-CN and CN-to-VN message error probabilities for iteration ℓ and node degrees i and j, respectively (assuming density evolution with the current DD to be improved by solving the linear program).

Algorithm 4 summarizes the procedure for DD optimization. Note that for the density evolution runs at lines 4 and 6, there is the possibility that the evolution does not converge withing sufficient precision for the updated DDs. If this is the case, we scale down the channel noise and rerun density evolution until convergence. Alternative update schedules are possible as well.

4.4 Performance and Design Aspects of LUT Decoders

In this section, we focus on the practical aspects of LUT decoder design. This includes choosing decoder parameters based on system requirements as well as examining decoder simplifications such as LUT reuse and alphabet downsizing and their impact on performance. Furthermore,

Algorithm 4 DD optimization for LUT decoding

```
Input: Initial DD $(\lambda,\rho)$ with rate $R$, target rate $R_t > R$, step size $\delta$, LUT resolution $N_q$
Calculate threshold $\sigma(\lambda,\rho,N_q)$
while $R < R_t$ do
    compute $\lambda_2^*(\sigma,\rho,N_q)$
    run density evolution and save probabilities $P_\ell$, $P_{\ell|i}$
    update $\lambda$ by solving the VN linear program (4.49)
    run density evolution and save probabilities $Q_\ell$, $Q_{\ell|j}$
    update $\rho$ by solving the CN linear program (4.50)
    compute updated rate $R(\lambda,\rho)$
end while
Output: Optimized DD pair $(\lambda,\rho)$ for rate $R_t$
```

we compare different LUT decoder variants with conventional BP and MS decoders.

4.4.1 Choosing an Operating Point

Due to the iterative, density evolution based design, any information-optimal LUT sequence depends on the initial density of the channel $p_{\mathsf{L}|\mathsf{X}}$. We observed in our simulations that for the case of quantized BI-AWGN channels, if a decoder was designed for some noise level σ, it also works well with lower levels $\sigma' < \sigma$. This behavior is consistent with density evolution results and can be understood in the context of LUTs acting as quantizers subsequent to BP updates, cf. (4.32) and Figure 4.1. Specifically, when decoding an output from a channel with standard deviation σ' with a decoder designed for $\sigma > \sigma'$, the quantizer sequence $\Phi^{(\ell)}(\sigma)$ acts more conservatively than $\Phi^{(\ell)}(\sigma')$ would, however, convergence is still achieved but takes longer due to the pessimistic quantizers.

We define the BI-AWGN *threshold* in accordance to (2.70),

$$\sigma^*(\mathscr{L},P_e,\lambda,\rho) = \sup\{\sigma : P_e^{(\mathscr{L})} \leq P_e\}, \tag{4.51}$$

i.e., the highest noise level at which a message error probability P_e can be reached within $\mathscr{L}$ iterations. For calculating limiting thresholds such as those listed in Table 4.2, we let $\mathscr{L} \to \infty$ and $P_e \to 0$ within machine precision and computational complexity constraints. For a practical LUT decoder design flow, we specify P_e and $\mathscr{L}$ according to the system requirements and run a density evolution bisection search to find the corresponding threshold which is chosen as the operating point for the decoder. In this context, we also refer to the operating point as *design SNR*, defined as the E_b/N_0 value of the threshold noise level, $\gamma \triangleq -10\log_{10}(2R\sigma^{*2})$. Algorithm 5 summarizes the iterative LUT decoder design process.

Figure 4.8 illustrates the tradeoff between error floor and waterfall performance corresponding to a tradeoff between error probability requirement and resulting threshold in (4.51): If we increase the tolerated message error probability P_e, higher thresholds are obtained with a corresponding LUT sequence. Consequently, those LUTs perform better for lower SNRs,

Algorithm 5 Density Evolution based LUT design

```
Input: DD $(\lambda, \rho)$, precision $\Delta\sigma > 0$, maximum error probability $P_e$, number of iterations $\mathscr{L}$,
    min-LUT flag ML
Calculate Rate $R(\lambda, \rho)$
$\sigma_{\text{min}} \leftarrow 0$
$\sigma_{\text{max}} \leftarrow \frac{1}{2^{2R}-1}$
while $\sigma_{\text{max}} - \sigma_{\text{min}} > \Delta\sigma$ do
    $\sigma \leftarrow (\sigma_{\text{max}} - \sigma_{\text{min}})/2$
    Get $p_{\mathsf{L}|\mathsf{x}}(L|x)$ corresponding to BI-AWGN($\sigma^2$)
    achievable ← false
    for $\ell = 0, \ldots, \mathscr{L}$ do
        Build the product distribution (4.18)
        Design VN LUT update(s) $\Phi^{(\ell)}$
        $p^{(\ell)}_{\mu|\mathsf{x}}(\mu|x) = \sum_{\Phi^{(\ell)}(L,\overline{\mu})=\mu} p^{(\ell)}_{\overline{\mu}|\mathsf{x}}(\overline{\mu}|\, x)$
        Build the product distribution (4.19)
        if ML then
            $p^{(\ell+1)}_{\overline{\mu}|\mathsf{x}}(\overline{\mu}|x) = \sum_{\mu:\,\text{sign}(\mu)\min|\mu|=\overline{\mu}} p^{(\ell)}_{\mu|\mathsf{x}}(\mu|\, x)$
        else
            Design CN LUT update(s) $\overline{\Phi}^{(\ell)}$
            $p^{(\ell+1)}_{\overline{\mu}|\mathsf{x}}(\overline{\mu}|x) = \sum_{\overline{\Phi}^{(\ell)}(\mu)=\overline{\mu}} p^{(\ell)}_{\mu|\mathsf{x}}(\mu|\, x)$
        end if
        if $\text{P}_\text{e}(p^{(\ell)}_{\mu|\mathsf{x}}) < P_e$ then
            achievable ← true
            break
        end if
    end for
    if achievable then
        $\sigma_{\text{min}} \leftarrow \sigma$
    else
        $\sigma_{\text{max}} \leftarrow \sigma$
    end if
end while
$\sigma^* \leftarrow \sigma$
Output: Threshold $\sigma^*$, VN LUT sequence $\Phi^{(\ell)}$ (and $\overline{\Phi}^{(\ell)}$ for ML = false)
```

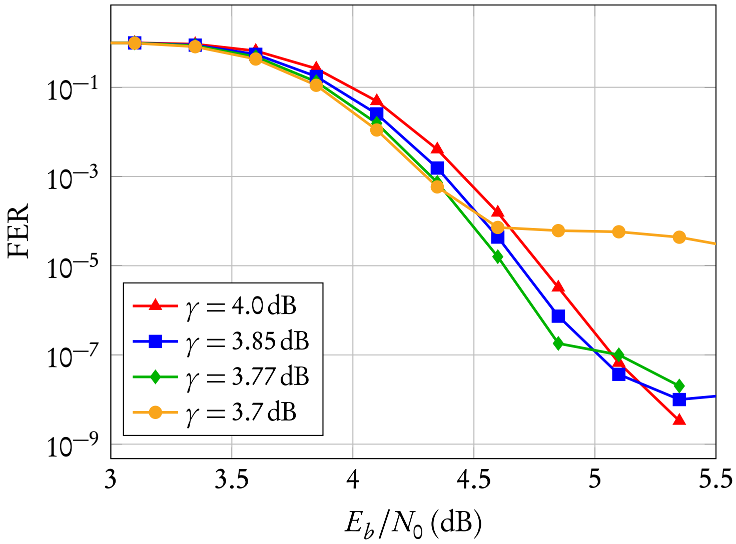


Figure 4.8: FER simulations revealing the relation between design SNR γ and error floor. For all simulations, min-LUT decoders for the length $N = 2048$, rate $R = 0.84$ 10G Ethernet LDPC code [53] have been used. All decoders feature 3 bit channel LLRs and messages downsized from 3 to 1 bit over the course of $\mathscr{L} = 8$ decoding iterations. Out of 8 iterations, only 4 have their own update rules and the rest is based on LUT reuse.

(i.e., in the waterfall region) at the cost of increased error floors (higher residual P_e).

4.4.2 LUT Reuse and Alphabet Downsizing

One drawback of LUT decoding as compared to conventional BP and MS decoding is the fact that LUTs may be distinct for different iterations, i.e., $\Phi^{(\ell)} \neq \Phi^{(\ell')}$ for $\ell \neq \ell'$ in general. This increases complexity when implementing certain decoder architectures, whereas other architectures such as unrolled decoders [6, 33] or serial decoders are not or only marginally affected.

In Figure 4.9 we show the dependency of the BI-AWGN threshold (4.51) on the maximum number of decoding iterations for different DDs for both BP and LUT decoding. As we can see, irrespective of the decoder, regular codes are reaching their threshold faster than irregular codes but also saturate at lower threshold values. Once again, we can see from the large gap between BP and LUT threshold performance that the BP optimized DD (λ_4, ρ_4) is ill suited for LUT decoding. Furthermore, the BP optimized DD only surpasses the threshold of the LUT optimized DD for 1000 or more iterations.

Interestingly, the LUT optimized DD (λ_2, ρ_2) converges towards the limit threshold substantially faster than the BP optimized ensemble (λ_4, ρ_4), reaching the limiting value already for roughly 100 iterations, whereas for (λ_4, ρ_4), it takes 10^4 iterations to converge. This indicates that the properties of LUT optimized DDs are not only suited to increase the limit thresholds but also tend to increase the speed of convergence towards the thresholds. Consequently, the complexity of implementing different LUTs for all iterations is mitigated by the LUT code design. To even further reduce the number of distinct LUT stages, we propose to reuse LUTs designed for some iteration ℓ for some number of subsequent iterations $\ell' > \ell$.

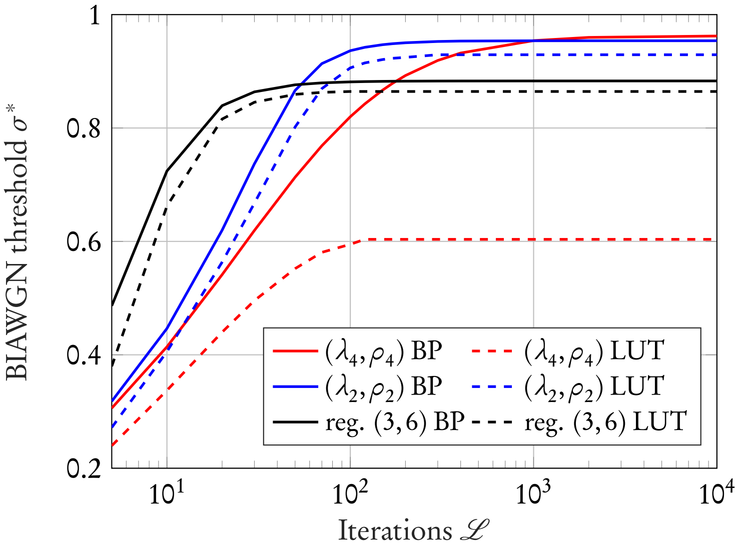


Figure 4.9: Density evolution threshold over number of iterations for different DDs (cf. Table 4.2)

While this is not optimal, if designed properly, it serves to reduce decoder complexity at the cost of a relatively small performance penalty. Let $\boldsymbol{r} \in \{0,1\}^{\mathscr{L}}$ denote the *reuse pattern*, i.e., r_ℓ is 1 if $\Phi^{(\ell)}$ is reused for iteration $\ell+1$ and 0 otherwise. In order to keep the impact on decoder performance as low as possible, we found that the reuse pattern must be carefully designed for each decoder. Simple, heuristic approaches such as reusing LUTs for a constant number of iterations throughout the decoding process do not yield satisfactory results. We can once again explain this by the fact that LUTs act as quantizers on BP message updates: It has been observed in [77] that the convergence behaviour of density evolution tends to be very inhomogeneous, i.e., during some iterations densities evolve fast and P_e decreases a lot, while for other iterations, decoding almost gets stuck at a fixed point and densities hardly change over many iterations. Thus, it makes sense to reuse LUTs for the latter case, since due to similar densities, the optimal LUTs for those iterations should be similar as well. To this end, we suggest a greedy reuse pattern optimization, cf. Algorithm 6. Starting with no reuse at all, i.e., $\boldsymbol{r} = 0$ the greedy algorithm iteratively reuses LUTs that lead to the least worsening in decoder convergence. That is, each iteration of the algorithm (i.e., lines 4 to 15) probes for the iteration number at which reusing a LUT leads to the fastest convergence of density evolution and adds a one to the reuse pattern at this iteration. The complexity of this algorithm is $O(\mathscr{L}^3)$, assuming that the number of distinct stages $\mathscr{L}_u$ is a fraction of the total number of iterations, $\mathscr{L}_u = \mathscr{L}(1-\alpha)$, where $0 < \alpha < 1$ is the *reuse factor*.

Yet another way to reduce implementation complexity is message resolution downsizing, i.e., reducing the size of the message set

$$|\mathcal{M}_{\ell'}| \leq |\mathcal{M}_\ell| \quad \text{for } \ell' > \ell. \tag{4.52}$$

The idea here is that during the decoding process, the distribution of labels becomes increas-

Algorithm 6 Greedy reuse pattern optimization

```
Input: Number of distinct LUT stages $\mathscr{L}_u$, total number of iterations $\mathscr{L}$, degree distribution
    $(\lambda, \rho)$, channel parameter $\sigma$, maximum message error probability $P_e$
i ← 0
r ← $0_{\mathscr{L}}$
while $i < \mathscr{L} - \mathscr{L}_u$ do
    $j \leftarrow \mathscr{L} 1_{\mathscr{L}}$
    for $\ell = 1 \dots \mathscr{L}$ do
        if $\neg r_\ell$ then
            $r' \leftarrow r$
            $r'_\ell \leftarrow 1$
            Run density evolution with $r'$ and save error trace $P_k$
            $j_\ell \leftarrow \min\{k : P_k \leq P_e\}$
        end if
    end for
    $\ell^* \leftarrow \arg\min j$
    $r_{\ell^*} \leftarrow 1$
    $i \leftarrow i + 1$
end while
Output: Optimized reuse pattern $r$
```

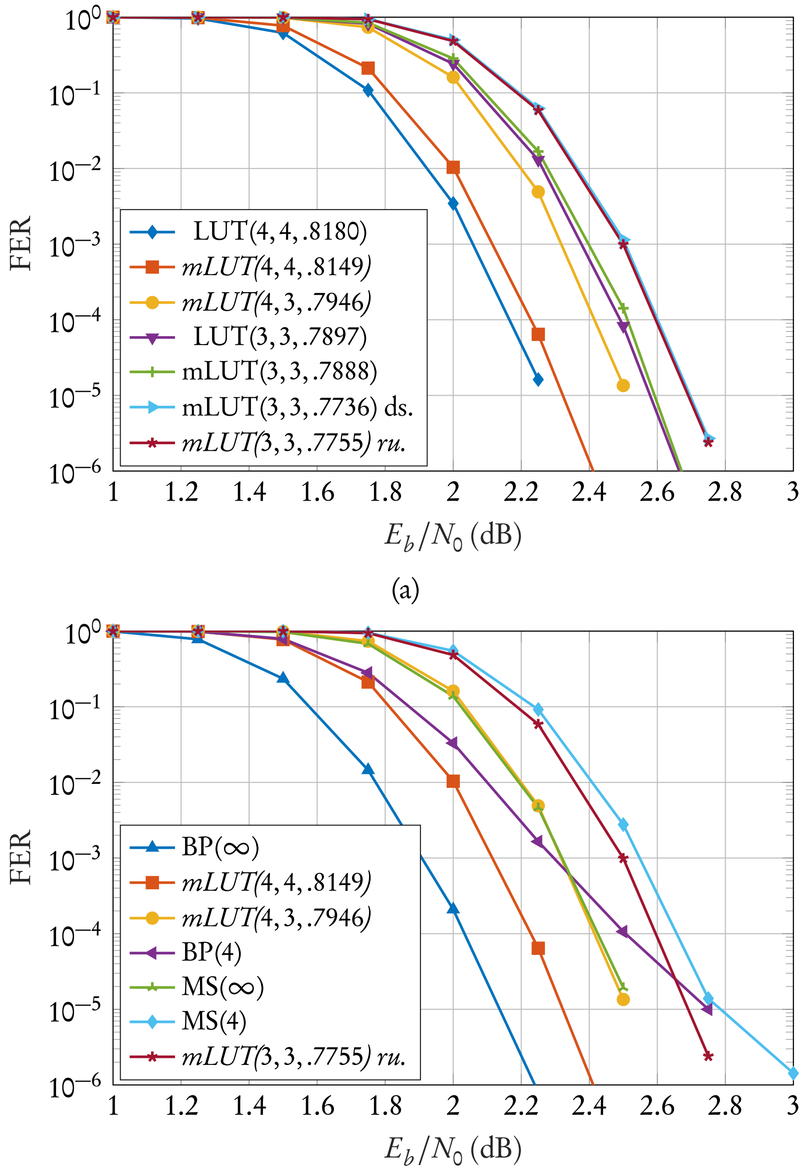
ingly deterministic over iterations, corresponding to the message value pmfs converging to Δ_∞, cf. (4.2) and (4.12). Consequently, message values (4.12) are highly concentrated for high ℓ and thus, a low resolution LUT quantizer can be used without sacrificing much of decoder performance, cf. Figure 4.1.

Figure 4.10a shows the effects of LUT reuse and alphabet downsizing for the case of a rate 1/2, regular (3, 6) LDPC code. The reuse-based decoder with reuse factor $\alpha = 0.8$ performs about the same as the decoder featuring a downsize from 3 to 2 bits message resolution for the last 10 decoding iterations. Both of these two reduced-complexity decoders lose only about 0.1 dB compared to the baseline decoder. Note that LUT reuse and downsizing can be combined but downsizing is only possible at iterations that are not being reused already. Figure 4.8 shows that even with both reuse and downsizing, error floors can be avoided even for short codes and 3 bits resolution.

4.4.3 Analyzing Decoder Performance

Relative performance of LUT decoder variants

Figure 4.10a shows the FER performance of various LUT decoders. In coherence to the density evolution threshold results, min-LUT decoders lose very little as compared to purely LUT based decoders with the same resolution, however, the gap between the two variants increases for increasing resolutions. Once again, this behaviour can be explained by the analogy between LUT and BP decoding, cf. Theorem 9: For decreasing message resolution, $|\mathcal{M}_\ell| \to 2$, the BP update rule (2.61) tends towards the MS update rule (2.63).



(a)

(b)

Figure 4.10: FER results for a length $N = 5000$ regular (3, 6) LDPC code decoded using $\mathscr{L} = 20$ iterations. (a) Relative performance of various LUT and min-LUT decoders. For the LUT decoders, the triple (Q_L, Q_μ, σ^*) denotes number of quantization bits for channel LLRs, messages and the design noise level. (b) Comparison of some of the LUT decoders from (a) with conventional BP and MS decoders. For fixed point BP and MS decoders, the number in paranthesis denotes the bit width, whereas ∞ indicates floating point precision.

Note that FER results are consistent with density evolution threshold results in the sense that higher thresholds correspond to lower FERs. Furthermore, the FER performance gap between two LUT decoders with thresholds σ_1^* and σ_2^* can be reasonably well approximated by their thresholds, $\Delta\gamma = |20\log_{10}(\sigma_1^*/\sigma_2^*)|\,\text{dB}$. E.g., for the case of the LUT $(4,4,0.8180)$ and the min-LUT $(3,3,0.7888)$ decoder, $\Delta\gamma \approx 0.32\,\text{dB}$, which matches the gap between the two respective FER curves fairly well.

Lastly, we also ran simulations using LUT tree structures other than binary maximum-breadth trees (cf. Section 4.2.3) and found that the FER results matched the density evolution predictions, thus confirming the superiority of maximum-breadth over maximum-height trees.

LUT decoders vs. conventional MS and BP decoders

Figure 4.10b compares LUT decoders to conventional BP and MS decoders. The min-LUT $(4,4,0.8149)$ decoder loses only $0.2\,\text{dB}$ as compared to the floating point BP decoder and it easily beats the 4 bit fixed point version of the BP decoder that exhibits error floors due to quantization effects. The min-LUT $(4,3,0.7946)$ decoder performs about the same as a conventional MS decoder at floating point precision using only 3 bits for message representation. After reducing the resolution of channel LLRs to 3 bits and reusing a fraction of $\alpha = 0.8$ of the decoder LUTs, the min-LUT $(3,3,.7755)$ performs still better than a fixed point MS decoder using 4 bits, especially in the error floor region.

We obtain similar results for irregular codes but we observed that higher LUT resolutions are necessary to seize their full potential. This also holds true for fixed point versions of MS decoders, as can be seen by comparing the relative performance of MS(4) decoders for regular codes (Figure 4.10b) to irregular codes (Figure 4.12). Similar observations have been made in [93]. Figure 4.11 compares different decoders for regular and irregular, LUT optimized codes in terms of FER. As expected, (unquantized) BP has the best performance, followed by the purely LUT based decoder with 4 bits resolution, losing less than 0.2 dB as compared to the optimal decoder. Consistent with threshold results, the min-LUT decoder performs slightly worse. With 4-bit messages, min-LUT decoding is within about 0.2 dB of BP decoding. The min-LUT decoder with reuse factor $\alpha = 0.8$ is only slightly worse than without reuse, while reducing the complexity by 80%. A substantial performance loss (0.55 dB from BP) occurs only after the LUT message resolution is reduced to 3 bit. We also included BP results for a regular $(3,6)$ code decoded at floating point precision for comparison. Even with 4-bit LUT decoding, the irregular code outperforms the BP-decoded regular $(3,6)$ code. This is in agreement with the density evolution results in Figure 4.6: the LUT threshold of (λ_2,ρ_2) is below the BP threshold of the $(3,6)$ code for 3 bits of resolution but outperforms the regular code for 4 or more bits.

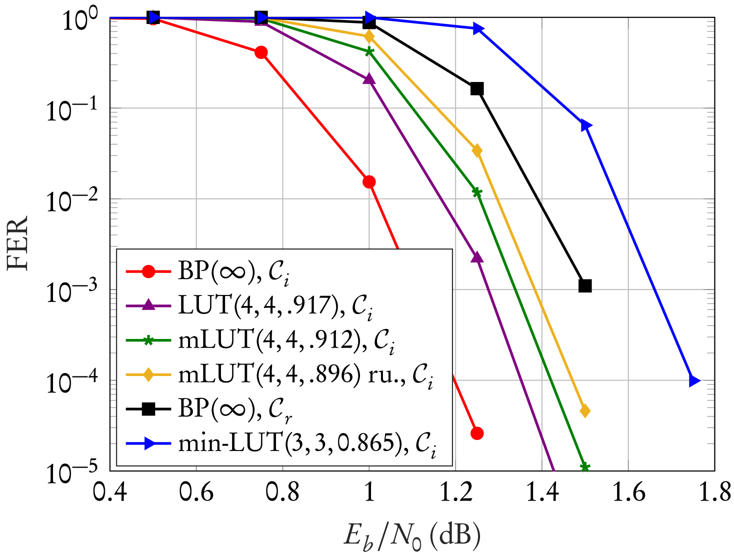


Figure 4.11: FER Simulations for different codes of rate 1/2, length $N = 10000$ and $\mathscr{L} = 100$ decoding iterations. Both the regular code $\mathcal{C}_r$ and the irregular code $\mathcal{C}_i$ have been created using PEG [49] from the regular (3,6) ensemble and the DD (λ_2, ρ_2) from Table 4.2, respectively.

LUT decoders for irregular codes

Next, we assess the performance of LUT decoders with LUT optimized codes in a practical scenario. By using the LUT optimized DD (λ_1, ρ_1) in (5.120), we created a code with similar degree structure as the rate 1/2 DVB-S2 LDPC code [29]. Note that the degree structure of the DVB-S2 code approximately matches the BP optimized DD (λ_3, ρ_3), and thus, due to the low threshold, cannot be expected to perform well with LUT decoding. We compare the two codes for different decoding algorithms in Figure 4.12. As we can see, the DVB-S2 code using BP performs best. However, quantized MS decoding with 4 and 5 bits entails a huge performance degradation and high error floors. For higher resolutions up to floating point precision (not shown), the water fall region remained unchanged (only the error floors vanished). In contrast, the LUT-optimized code with 4-bit min-LUT decoding performs much closer to the BP-decoded DVB-S2 code (BP decoding of the LUT-optimized code is only marginally better than LUT decoding). We conclude that a LUT-optimized code with 4-bit min-LUT decoding is capable of outperforming a MS decoder using a standard compliant code at floating point precision by roughly 0.5 dB.

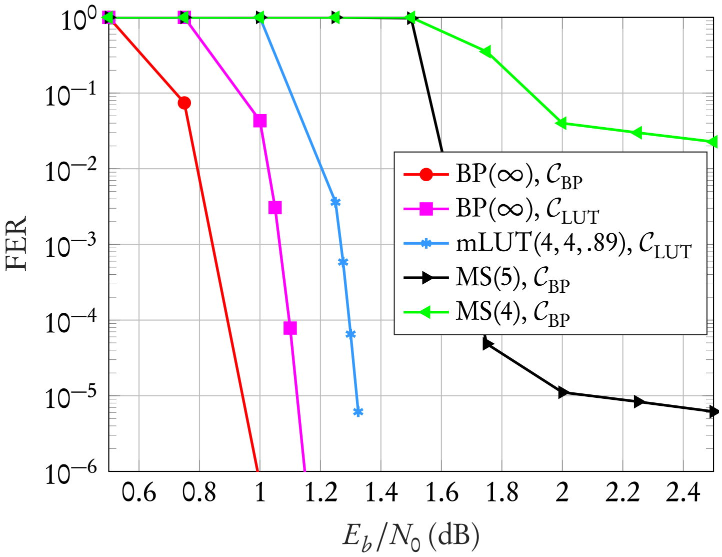


Figure 4.12: FER Simulations for the LUT optimized code $\mathcal{C}_{\mathrm{LUT}}$ and the BP optimized DVB-S2 [29] code $\mathcal{C}_{\mathrm{BP}}$. For for both codes $N = 64800$ and for all decoders $\mathscr{L} = 100$.

Chapter 5

SMLDPC Codes

In this chapter, we introduce superposition modulated low-density parity-check (SMLDPC) codes — a new class of codes for continuous input, additive white noise channels. SMLDPC codes are obtained by combining superposition modulation (SM) [46, 64] with LDPC coding (cf. Section 2.2). In that sense, they constitute a direct mapping from discrete messages to length-L codewords in $\mathbb{R}^L$ or $\mathbb{C}^L$ — much like in the setting of Shannon's famous channel coding theorem for the AWGN channel [87]. Although SMLDPC codes can be viewed as a concatenation of well known coding and modulation techniques, the resulting code can be decoded without explicit demodulation. Specifically, we can use message passing decoders with *linear complexity* that operate directly on the channel output and produce estimates of the transmitted data bits.

SMLDPC codes can be viewed as a generalization of conventional LDPC codes [32, 65, 77] and LDHC codes introduced in [114, 115]. Our analysis of SMLDPC codes differs from the analysis in [114, 115] in that we consider both components of the code —namely the modulation and the parity check constraints— in a joint and unified fashion rather than designing and optimizing them separately and matching them based on EXIT analysis (cf. Section 2.3). In terms of the factor graph of an SMLDPC code, this means that aside from the well-known check nodes (CNs), we consider variable nodes (VNs) with a 2 dimensional degree structure as well as *symbol nodes (SNs)* representing the constraints on the receive symbols. This is in contrast to the analysis of LDHC codes in [114, 115], which considers VN-to-SN and VN-to-CN degrees separately.

Due to our joint, comprehensive approach, and due to the introduction of random SM coefficients, we are able to extend the asymptotic theory of LDPC codes to SMLDPC codes, including the famous concentration result (Theorem 3) as well as monotonicity concepts that give rise to the very same thresholding phenomenon that is exhibited for long LDPC codes. Furthermore, we show that for BP decoding of SMLDPC codes, the message densities are symmetric and LLR-consistent, which tremendously simplifies EXIT analysis and guarantees that the threshold obtained from EXIT analysis is identical to the density evolution threshold under the assumption of exact EXIT functions. Furthermore, the unified approach allows for the consistent definition of irregular SMLDPC codes which gives rise to a joint optimization of degree distributions for both the modulation and parity check degree structure.

The remainder of this chapter is structured as follows. In Section 5.1, we introduce the system model of SMLDPC codes and the underlying sparse graph that allows for low complexity message passing decoding. Subsequently, Section 5.2 discusses the role of the modulation coefficients and gives an overview on the vast range of transmission scenarios possible with different choices of coefficients — among them being Gray coded PAM and QAM, different variations of LDPC coded SM as well as MIMO transmission over a fading channel. In Section 5.3, we then proceed to define a general message passing decoding algorithm and show how BP decoding is a special instance of such an algorithm. Furthermore, we discuss methods on how to evaluate the BP SN update and derive novel approximative update rules for the case of high dimensional SNs. In Section 5.4, we derive the concentration theorem for SMLDPC codes which ensures that the performance of long codes is adequately characterized by the ensemble average of the cycle-free case. In Section 5.5, we obtain results on message symmetry and LLR-consistency and show that the threshold of an SMLDPC code can be obtained via EXIT analysis. Section 5.6 then introduces irregular SMLDPC codes and Section 5.7 develops strategies to optimize the irregular degree structure of code ensembles.

Lastly, we want to remark that our original motivation to study SMLDPC codes came from the desire to extend the quantized message passing decoding scheme we developed in Chapter 4 to include a LUT mapping that produces discrete messages from quantized, discrete receive symbols — in other words, a discrete SN update. Searching the literature, we came across LDHC codes, but quickly realized that without a broader theoretical foundation and message symmetry, an extension was not possible, which eventually led us to develop those concepts ourselves.

5.1 Coded Modulation Input-Output Model

We consider superposition modulation of LDPC encoded data transmitted over a complex, additive white noise channel,

$$y = Ax + w. \tag{5.1}$$

Here, w is an iid, zero mean random vector and $x = (1-2c) \in \{-1,+1\}^N$ is the *polar* version of an LDPC codeword $c \in \mathcal{C} = \{c \mid Hc = 0\}$ with sparse parity check matrix $H \in \{0,1\}^{M\times N}$, cf. Section 2.2. The *global modulation matrix* $A \in \mathbb{C}^{Ld_y \times N}$ is sparse as well, so that for

$$y = \begin{pmatrix} y_1 \\ \vdots \\ y_L \end{pmatrix}, \tag{5.2}$$

the length-d_y subvectors $\boldsymbol{y}_l$, $l = 1,\dots,L$ only depend on a small subset $\boldsymbol{x}_{\mathcal{N}_{\mathrm{s}_l}}$ of coded bits,

$$\boldsymbol{y}_l = \sum_{n\in\mathcal{N}_{\mathrm{s}_l}} \boldsymbol{a}_{l,n} x_n + \boldsymbol{w}_l = \boldsymbol{A}_l \boldsymbol{x}_{\mathcal{N}_{\mathrm{s}_l}} + \boldsymbol{w}_l, \tag{5.3}$$

where $n \in \mathcal{N}_{\mathrm{s}_l} = \{n_1 \dots, n_{|\mathcal{N}_{\mathrm{s}_l}|}\}$ are the indices of the non-zero columns $\boldsymbol{a}_{l,n} \in \mathbb{C}^{d_y}$ in $A[l : l + d_y - 1, :]$, which constitute the *edge coefficient matrix*

$$\boldsymbol{A}_l = (\boldsymbol{a}_{l,n_1}, \dots, \boldsymbol{a}_{l,n_{|\mathcal{N}_{\mathrm{s}_l}|}}) \in \mathbb{C}^{d_y \times |\mathcal{N}_{\mathrm{s}_l}|} \tag{5.4}$$

for receive subvector $\boldsymbol{y}_l$.

Define $A_{\mathbb{1}} \in \{0,1\}^{Ld_y \times N}$ as $(A_{\mathbb{1}})_{ij} = \mathbb{1}\{a_{ij} \neq 0\}$, then due to (5.3), we can decompose

$$A_{\mathbb{1}} = \boldsymbol{\Gamma}_A \otimes \mathbf{1}_{d_y}, \tag{5.5}$$

with $\boldsymbol{\Gamma}_A \in \{0,1\}^{L\times N}$ characterizing the connectivity between symbol vectors $\boldsymbol{y}_l$, $l = 1,\dots,L$ and code bits x_n, $n = 1,\dots,N$. Let the sparse connectivity matrix $\boldsymbol{\Gamma} \in \{0,1\}^{L+M\times N}$ be defined as

$$\boldsymbol{\Gamma} \triangleq \begin{pmatrix} \boldsymbol{\Gamma}_A \\ \boldsymbol{H} \end{pmatrix}, \tag{5.6}$$

and let $G(\boldsymbol{\Gamma})$ denote the corresponding, sparse tripartite factor graph, consisting of

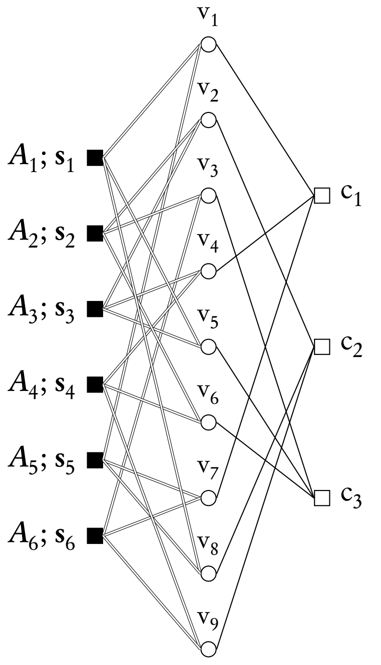
Variable nodes: Each column $n \in \{1,\dots N\}$ in $\boldsymbol{\Gamma}$ corresponds to a coded bit x_n which is represented by a VN v_n in the factor graph.

Check nodes: Each row $m \in \{1,\dots M\}$ in $\boldsymbol{H}$ corresponds to a parity check on the codeword which is represented as a CN c_m, i.e., a factor node accounting for the parity check constraint. A *CN side edge* $\mathrm{e} = \{\mathrm{v}_n, \mathrm{c}_m\}$ connects v_n and c_m iff $(\boldsymbol{H})_{mn} = 1$.

Symbol nodes: Each row $l \in \{1,\dots L\}$ in $\boldsymbol{\Gamma}_A$ corresponds to a received symbol vector $\boldsymbol{y}_l$, $l = 1,\dots,L$, cf. (5.2) and (5.3), which is represented as a symbol node (SN) $\mathbf{s}_l$. A SN is a factor node accounting for the constraints of the linear model (5.3). A *SN side edge* $\mathbf{e} = \{\mathrm{v}_n, \mathbf{s}_l\}$ connects v_n and $\mathbf{s}_l$ iff $(\boldsymbol{\Gamma}_A)_{ln} = 1$.

For the moment, assume that this graph is regular in the sense that each row of $\boldsymbol{H}$ contains d_c ones and each column contains d_{vc} nonzero elements, whereas each row of $\boldsymbol{\Gamma}_A$ contains d_s nonzero elements[1]and each column contains d_{vs} ones. We call d_c, d_s and $\boldsymbol{d}_v = (d_{vs}, d_{vc})$ the CN degree, the SN degree and the VN degree, respectively, where the VN degree is determined by the VN-to-SN degree d_{vs} and the VN-to-CN degree d_{vc}. Figure 5.1 depicts an example of a regular graph with corresponding edge coefficient matrices A_l, $l = 1,\dots,L$. The CN side of the graph characterized by $\boldsymbol{H}$ represents a sparse parity check code (cf. Section 2.2). The SN

[1] i.e., $|\mathcal{N}_{\mathrm{s}_l}| = d_s$ irrespective of l.



$$A=\begin{pmatrix} a_{1,1} & 0 & 0 & 0 & a_{1,5} & 0 & 0 & a_{1,8} & 0 \\ 0 & a_{2,2} & a_{2,3} & 0 & 0 & a_{2,6} & 0 & 0 & 0 \\ 0 & a_{3,2} & 0 & a_{3,4} & a_{3,5} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & a_{4,4} & 0 & a_{4,6} & 0 & 0 & a_{4,9} \\ a_{5,1} & 0 & 0 & 0 & 0 & 0 & a_{5,7} & a_{5,8} & 0 \\ 0 & 0 & a_{6,3} & 0 & 0 & 0 & a_{6,7} & 0 & a_{6,9} \end{pmatrix}$$

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

Figure 5.1: A regular 3-partite graph G with $d_s = 3$, $\boldsymbol{d}_v = (d_{vs}, d_{vc}) = (2,1)$ and $d_c = 3$. Furthermore, $K = 6$, $N = 9$, $M = N - K = 3$ and $L = 6$. The dimension d_y of the SNs is not visible in the graph. The rate is given by $R = \frac{1}{d_y}(1 - \frac{d_{vc}}{d_c})\frac{d_s}{d_{vs}} = \frac{1}{d_y}(1 - \frac{1}{3})\frac{3}{2} = \frac{1}{d_y}$ bit/dimension, or equivalently, $R = \frac{K}{Ld_y} = \frac{6}{6d_y} = \frac{1}{d_y}$ bit/dimension. In this example, e.g., $A_1 = (a_{1,1}, a_{1,5}, a_{1,8})$.

side corresponding to $\boldsymbol{\Gamma}_A$ on the other hand represents the mapping from code bits to symbol subvectors (rows) as well as a sparse repetition code (columns) in case $d_{vs} > 1$, since the same code bit then contributes to multiple symbol vectors.

The sparse structure of $\boldsymbol{\Gamma}$ implies that the number of both VN-to-SN as well as VN-to-CN edges is *linear* in the code word length N. Furthermore, for regular graphs, the number of VN-to-SN edges and VN-to-CN fulfill

$$Ld_s = Nd_{vs} \tag{5.7}$$

$$Md_c = Nd_{vc}, \tag{5.8}$$

and thus, the code rate (in bits per/dimension) can be expressed as a function of the node degrees only,

$$R = \frac{K}{Ld_y} = \frac{N-M}{d_y L} = \frac{1}{d_y}\left(1 - \frac{M}{N}\right)\frac{N}{L} = \frac{1}{d_y}\left(1 - \frac{d_{vc}}{d_c}\right)\frac{d_s}{d_{vs}}. \tag{5.9}$$

5.2 Modulation Coefficient Model

So far, we have only considered the connectivity of the sparse modulation matrix. In this section, we establish a probabilistic model for the non-zero entries in A and show how the input-output relation introduced in Section 5.1 can be used to model a wide variety of communication channels and modulation approaches. We assume that A is *known both at the transmitter and the receiver.*

Recall that according to (5.1) and (5.3), the code bit x_n contributes to the channel output $\boldsymbol{y}_l$ by adding or subtracting $\boldsymbol{a}_{l,n}$, assuming that $n \in \mathcal{N}(l)$, or in other words, if there is a connection between SN s_l and VN v_n. Hence, we call $\boldsymbol{a}_{l,n}$ the *edge coefficient vector* of SN edge $\{\mathrm{s}_l, \mathrm{v}_n\}$, where all edge coefficient vectors contributing to a symbol vector $\boldsymbol{y}_l$ are assembled in the *edge coefficient matrix* A_l, cf. (5.3).

5.2.1 Independence and Symmetry

Assumption 2 (Edge coefficient independence and symmetry)**.** *The discrete and/or continuously distributed edge coefficient matrices $\mathsf{A}_l \in \mathbb{C}^{d_y \times d_s}$ shall fulfill the following symmetry properties:*

1. *Coefficients of edges attached to different symbol nodes are independent:*

$$p_{\mathsf{A}_1,\ldots,\mathsf{A}_l}(A_1,\ldots,A_L) = \prod_{l=1}^{L} p_{\mathsf{A}_l}(A_l) \tag{5.10}$$

2. *All coefficient vectors are distributed equally according to a single distribution $p_{\mathsf{A}}(A)$ over $\mathbb{C}^{d_y \times d_s}$:*

$$p_{\mathsf{A}_l}(A) \equiv p_{\mathsf{A}}(A) \qquad l = 1,\ldots,L \tag{5.11}$$

3. *Column inversion invariance:*

$$p_{\mathsf{A}}(A^{\neg k}) = p_{\mathsf{A}}(A), \tag{5.12}$$

where $A^{\neg k}$ is the matrix A with the signs of the kth column inverted.

4. *Column permutation invariance: For any permutation matrix $\boldsymbol{P} \in \{0,1\}^{d_s \times d_s}$ we have*[2]

$$p_{\mathsf{A}}(A\boldsymbol{P}) = p_{\mathsf{A}}(A). \tag{5.13}$$

We denote the density of the iid local edge coefficient matrices as $p_{\mathsf{A}}(A)$, where as the density of the global matrix is denoted as $p_{\mathsf{A}_1,\ldots,\mathsf{A}_l}(A_1,\ldots,A_L)$. Furthermore, note that any

[2] Note that in contrast to the symmetry properties (5.10) to (5.12), (5.13) is not required for most results presented in this thesis. This is because the edges that connect SNs and VNs are randomly placed anyways, i.e., permutation occurs in any case from a global perspective. Thus, we can including permutation invariance into the coefficient model without loss of generality and it will be seen to be useful later on in terms of notation.

$p_{\mathbf{A}'}(A')$ can be made to fulfill (5.12) if the data is scrambled with an iid ±1 sequence, i.e., for any $\mathbf{A}' \sim p_{\mathbf{A}'}(A')$,

$$\mathbf{A} = \mathbf{A}'\operatorname{diag}\{\mathsf{s}_i\}_{i=1}^{d_s} \tag{5.14}$$

with $\mathsf{s}_i \in \{+1,-1\}$ iid and uniform fulfills (5.12). Similarly, permutation invariance (5.13) can be achieved for any $\mathbf{A}' \sim p_{\mathbf{A}'}(A')$ by randomly permuting the data,

$$\mathbf{A} = \mathbf{A}'\mathbf{P}, \tag{5.15}$$

with $\mathbf{P}$ distributed uniformly over the set $\mathcal{P}_{d_s}$ of all permutation matrices. From (5.14) and (5.15), we conclude that for any deterministic A',

$$\mathbf{A} = A'\mathbf{S}\mathbf{P}, \tag{5.16}$$

fulfills both (5.12) and (5.13) for uniform $\mathbf{S} \in \mathcal{S}_{d_s} = \left\{\operatorname{diag}\{s_i\}_{i=1}^{d_s} \;\middle|\; s_i = \pm 1\right\}$ and $\mathbf{P} \in \mathcal{P}_{d_s}$.

5.2.2 Capacity and Input Power

It follows from (5.3) and Assumption 2 that the probabilistic input-output relationship can be modeled as a fast-fading MIMO channel [101, Chapter 7], with perfect information at both transmitter and receiver. Let $\mathcal{X} = \{-1,+1\}$ and assume that the additive iid noise in (5.1) is complex Gaussian, i.e.

$$\mathbf{w} \in \mathbb{C}^{Ld_y}, \qquad \mathbf{w} \sim \mathcal{CN}(0, IN_0). \tag{5.17}$$

For any given coefficient distribution, the channel capacity C (in bits per dimension) is then given by [44]

$$\begin{aligned} d_y C &= -\mathbb{E}_{\mathbf{A}}\left[\mathbb{E}_{\mathbf{y}|\mathbf{A}}\left[\log_2\left(\frac{1}{2^{d_s}}\frac{1}{(\pi N_0)^{d_y}}\sum_{x\in\mathcal{X}^{d_s}}\exp\left(-\frac{\|\mathbf{y}-\mathbf{A}x\|^2}{N_0}\right)\right)\right]\right] - d_y\log_2(N_0\pi e) \\ &= d_s - \frac{1}{2^{d_s}}\sum_{x'\in\mathcal{X}^{d_s}}\mathbb{E}_{\mathbf{A}}\left[\mathbb{E}_{\mathbf{w}}\left[\log_2\left(\sum_{x\in\mathcal{X}^{d_s}}\exp\left(-\frac{\|\mathbf{A}x'-\mathbf{A}x+\mathbf{w}\|^2 + d_y N_0}{N_0}\right)\right)\right]\right]. \end{aligned} \tag{5.18}$$

To compare different modulation schemes, we need to ensure that the average input power is the same. Assuming iid code bits, we have $C_{\mathbf{x}} = I$, and thus, the average power (per dimension) of the channel input $\mathbf{A}\mathbf{x}$, denoted by P, can be obtained as

$$\begin{aligned} d_y P = \mathbb{E}\left[\,\|\mathbf{A}\mathbf{x}\|^2\right] = \mathbb{E}_{\mathbf{x}}\Big[\mathbf{x}^{\mathrm{T}}\overbrace{\mathbb{E}_{\mathbf{A}}\left[\mathbf{A}^{\mathrm{H}}\mathbf{A}\right]}^{\triangleq P_{\mathbf{A}}}\mathbf{x}\Big] &= \mathbb{E}_{\mathbf{x}}\left[\mathbf{x}^{\mathrm{T}}\boldsymbol{P}_{\mathbf{A}}\mathbf{x}\right] \\ &= \operatorname{tr}\{\boldsymbol{P}_{\mathbf{A}}\boldsymbol{C}_{\mathbf{x}}\} = \operatorname{tr}\{\,\boldsymbol{P}_{\mathbf{A}}\} = \sum_{k=1}^{d_s}\mathbb{E}\left[\mathbf{a}_k^{\mathrm{H}}\mathbf{a}_k\right], \end{aligned} \tag{5.19}$$

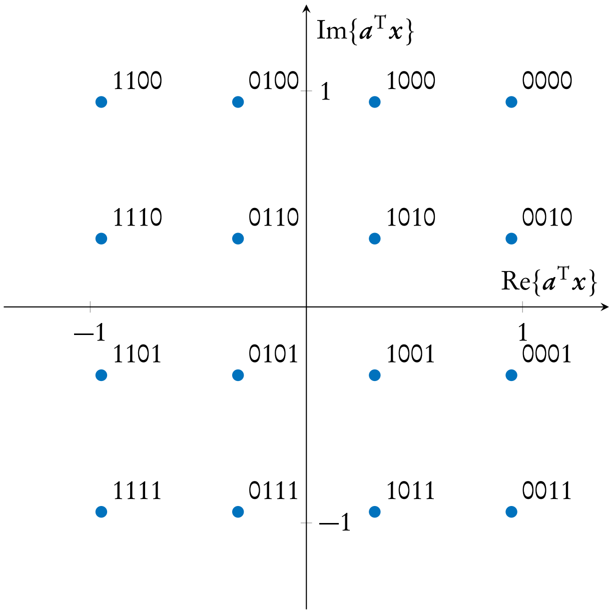


Figure 5.2: 16-QAM constellation and corresponding codebits $c = \frac{1}{2}(1-x)$ generated by $a^{\mathrm{T}} = \frac{1}{\sqrt{10}}(1,2,\mathrm{j},2\mathrm{j})$

where $\mathbf{a}_k$ are the columns of $\mathbf{A} = \left(\mathbf{a}_1,\ldots,\mathbf{a}_{d_s}\right)$, i.e., the edge coefficient vectors. Henceforth, we consider normalized edge coefficients, i.e., $\mathbb{E}\left[\,||\mathbf{A}\mathbf{x}||^2\right] = d_y$ or equivalently, $P = 1$.

In what follows, we examine different modulation and coding approaches that can be shown to be a special case of the above framework.

5.2.3 Gray-coded PAM and QAM-modulated LDPC codes

Let $d_y = 1$, i.e., the edge coefficient matrices reduce to length-d_s row vectors $\mathbf{a}_l^{\mathrm{T}}$. Let $M = 2^{d_s}$ and

$$\mathbf{a}^{\mathrm{T}} \in \mathcal{A}_{\mathrm{PAM}}^{M} \triangleq \left\{ \bar{a}^{\mathrm{T}} S P \,\middle|\, S \in \mathcal{S}_{d_s}, P \in \mathcal{P}_{d_s}, \bar{a}^{\mathrm{T}} = \sqrt{\frac{3}{M^2-1}} \begin{pmatrix} 2^0 & 2^1 & \ldots & 2^{\log_2 M-1} \end{pmatrix} \right\} \quad (5.20)$$

with uniform probability. Then for $\mathbf{x} \in \{-1,1\}^{d_s}$ with uniform probability, $\mathbf{a}^{\mathrm{T}}\mathbf{x} \in \sqrt{\frac{3}{M^2-1}}\{-(M-1),-(M-3),\ldots,(M-1),\}$ with uniform probability. Thus, we can create a M-PAM modulation, where the mapping from the bits to the symbols depends on the coefficient vector $\mathbf{a}_l^{\mathrm{T}}$ and in general, is different for any symbol $l = 1,\ldots,L$. Moreover, due to the summation mapping, it follows that the mapping from bits to symbols follows a Gray code [37].

Similarly, M-QAM constellations can be generated by letting

$$\mathbf{a}^{\mathrm{T}} \in \mathcal{A}_{\mathrm{QAM}}^{M} = \left\{ \frac{1}{\sqrt{2}} \begin{pmatrix} a_{\mathrm{R}}^{\mathrm{T}} & \mathrm{j}a_{\mathrm{I}}^{\mathrm{T}} \end{pmatrix} \,\middle|\, a_{\mathrm{R}}^{\mathrm{T}}, a_{\mathrm{I}}^{\mathrm{T}} \in \mathcal{A}_{\mathrm{PAM}}^{\sqrt{M}} \right\} \quad \text{with uniform probability.} \quad (5.21)$$

Figure 5.2 shows an example.

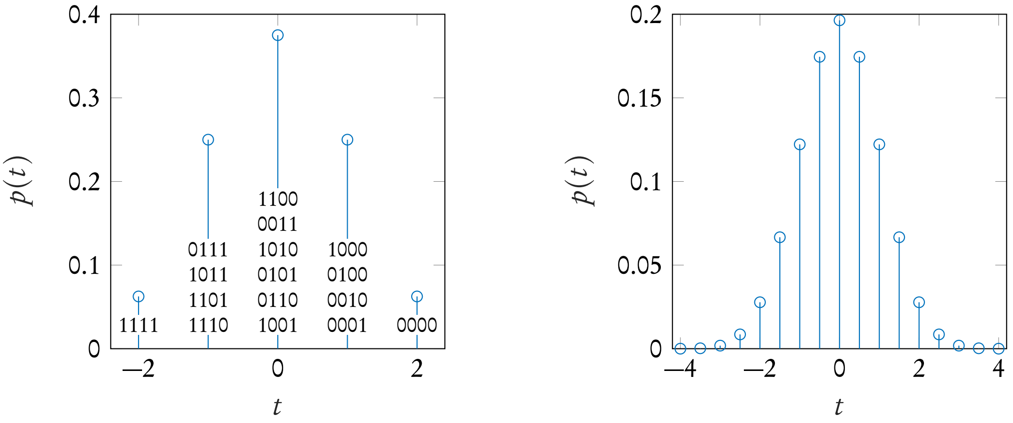
Note that for equiprobable constellations generated by (5.20) and (5.21), the capacity (5.18) becomes the coded modulation capacity [104],

$$C = \log_2 M - \frac{1}{M}\sum_{i=1}^{M}\mathbb{E}\left[\log_2\Big(\sum_{j=1}^{M}\exp\Big(\frac{|\alpha_i + \mathrm{w} - \alpha_j| - N_0}{N_0}\Big)\Big)\right]. \tag{5.22}$$

By restricting ourselves to $d_{vs} = 1$ (i.e., there is no repetition code), we can use our framework to design LDPC codes for bandwidth efficient, high order PAM and QAM constellations [48, 98, 121]. In this case, the modulation via the coefficient vectors effectively amounts to a random scrambling, interleaving and Gray mapping from bit vectors to symbols. For a practical system, both the transmitter and the receiver have to agree on the same scrambling and permutation sequence (i.e. signs and order of the edge coefficients for different symbols). Interestingly, scrambling has already been considered in [48] as a method to symmetrize the channel and simplify analysis. Later in the text, we will show that the symmetry requirements in Assumption 2 fulfill a similar purpose. The fact that regular PAM and QAM constellations can be generated by superposition is also recognized in [64], where it is referred to as "natural mapping", as well as in [46, 114], where it is referred to as a special case of "superposition modulation using unequal power allocation (SM-UPA)". However, in contrast to those works, we present a unifying framework for code design so that we can also capitalize on this observation.

5.2.4 Classic Superposition Modulation

Broadly speaking, we refer to classic SM as a set of techniques that allows the modulation symbol to be expressed as a weighted, linear combination of antipodal bits. The motivation for this approach can be traced back to [27], where it is shown that this technique bears the potential to achieve capacity on the Gaussian channel without active shaping. Depending on the choice of weighting coefficients, a wide variety of different modulation schemes is possible. For an introduction to superposition modulation and an overview on related schemes, the reader is referred to [46]. The concept of joint sparse graph for both SM demodulation and LDPC decoding has been previously established in [46, 114, 115]. More material on SM can be found in [83, 100, 114–116, 128]. In what follows, we will present several SM approaches. Note that all of them are special cases and can be described using our unifying framework introduced in Section 5.1. To meet the symmetry requirements of Assumption 2, we have to introduce random scrambling, cf. (5.14). Note that it has been observed in [114] that scrambling is necessary to avoid oscillations and error floors in iterative decoders. Furthermore, we generalize the concept to continuous modulations.



(a)

(b)

Figure 5.3: Real SM-EPA constellations for $d_s = 4$ and $d_s = 16$

SM-EPA

We again consider scalar channels, i.e., $d_y = 1$. For

$$\mathbf{a}^{\mathrm{T}} \in \mathcal{A}^{d_s}_{\mathrm{EPA},\mathbb{R}} \triangleq \left\{ \bar{a}^{\mathrm{T}} S \,\middle|\, S \in \mathcal{S}_{d_s}, \bar{a}^{\mathrm{T}} = \frac{\mathbf{1}^{\mathrm{T}}}{\sqrt{d_s}} \right\} \quad \text{with} \quad p_{\mathbf{a}}(a) = \frac{1}{2^{d_s}}, \tag{5.23}$$

we obtain real valued, scalar superposition modulation using equal power allocation (SM-EPA) [46, 114, 115]. The resulting channel input $\mathrm{t} = \mathbf{a}^{\mathrm{T}}\mathbf{x}$ has binomial distribution, cf. Figure 5.3,

$$p_{\mathrm{t}}(t) = 2^{-d_s} \binom{d_s}{(t\sqrt{d_s} + d_s)/2}, \qquad t \in \left\{ -\frac{d_s}{\sqrt{d_s}}, -\frac{d_s - 2}{\sqrt{d_s}}, \dots, \frac{d_s - 2}{\sqrt{d_s}}, \frac{d_s}{\sqrt{d_s}} \right\}. \tag{5.24}$$

For increasing d_s, the binomial distribution converges to a Gaussian distribution and therefore is potentially capable of achieving the Shannon capacity on the AWGN channel. Contrary to the schemes in Section 5.2.3, the mapping from bit vectors to symbols is not unique for SM-EPA, cf. Figure 5.3. According to [115], for non bijective modulations, it makes sense to also introduce a repetition code by choosing $d_{vs} > 1$.

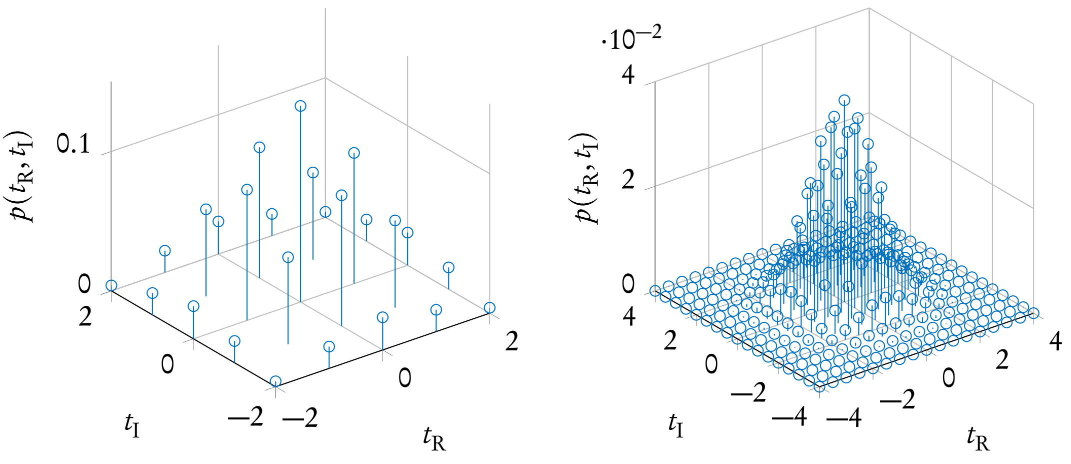
We can extend ordinary, real SM-EPA as presented above and in [46, 114, 115] in two ways: First, similarly to the expansion from PAM to QAM, we can define it over the complex domain by

$$\mathbf{a}^{\mathrm{T}} \in \mathcal{A}^{d_s}_{\mathrm{EPA},\mathbb{C}} \triangleq \left\{ \frac{1}{\sqrt{2}} \left(a_{\mathrm{R}}^{\mathrm{T}} \quad \mathrm{j}a_{\mathrm{I}}^{\mathrm{T}} \right) \,\middle|\, a_{\mathrm{R}}^{\mathrm{T}}, a_{\mathrm{I}}^{\mathrm{T}} \in \mathcal{A}^{d_s/2}_{\mathrm{EPA},\mathbb{R}} \right\} \quad \text{wih} \quad p_{\mathbf{a}}(a) = \frac{1}{2^{d_s}}. \tag{5.25}$$

For $\mathrm{t_R} + \mathrm{j}\mathrm{t_I} = \mathbf{a}^{\mathrm{T}}\mathbf{x}$, we then have

$$p_{\mathrm{t_R},\mathrm{t_I}}(t_{\mathrm{R}}, t_{\mathrm{I}}) = p_{\mathrm{t}}(t_{\mathrm{R}}) p_{\mathrm{t}}(t_{\mathrm{I}}) \tag{5.26}$$

with p_{t} as in (5.24) with $d_s/2$, cf. (5.25). Figure 5.4 shows examples of complex SM-EPA



(a) (b)

Figure 5.4: Complex SM-EPA constellations for $d_s = 8$ and $d_s = 32$

constellations.

Furthermore, it might be beneficial for demodulation if we group $d_y > 1$ symbols together and perform joint detection, i.e.,

$$\mathbf{A} \in \mathcal{A}^{d_y \times d_s}_{\mathrm{EPA},\mathbb{R}} \triangleq \left\{ \frac{1}{\sqrt{d_s}} \begin{pmatrix} \pm1 & \cdots & \pm1 \\ \vdots & \ddots & \vdots \\ \pm1 & \cdots & \pm1 \end{pmatrix} \right\} \quad \text{wih} \quad p_{\mathbf{A}}(A) = \frac{1}{2^{d_s d_y}}, \tag{5.27}$$

and

$$\mathbf{A} \in \mathcal{A}^{d_y \times d_s}_{\mathrm{EPA},\mathbb{C}} \triangleq \left\{ \frac{1}{\sqrt{2}} \begin{pmatrix} A_{\mathrm{R}} & \mathrm{j}A_{\mathrm{I}} \end{pmatrix} \,\middle|\, A_{\mathrm{R}}, A_{\mathrm{I}} \in \mathcal{A}^{d_y \times d_s/2}_{\mathrm{EPA},\mathbb{R}} \right\} \quad \text{wih} \quad p_{\mathbf{A}}(A) = \frac{1}{2^{d_s d_y}}. \tag{5.28}$$

SM-UPA and SM-GPA

SM-UPA and superposition modulation using group power allocation (SM-GPA) have been introduced in [46]. Unlike SM-EPA, SM-UPA uses coefficients of different power, i.e.,

$$\mathbf{a}^{\mathrm{T}} \in \mathcal{A}^{d_s}_{\mathrm{UPA},\mathbb{R},\rho} \triangleq \left\{ \bar{a}^{\mathrm{T}} \boldsymbol{S}\boldsymbol{P} \,\middle|\, \boldsymbol{S} \in \mathcal{S}_{d_s}, \boldsymbol{P} \in \mathcal{P}_{d_s}, \bar{a}^{\mathrm{T}} = \sqrt{\frac{1-\rho^2}{1-\rho^{2d_s}}} \begin{pmatrix} 1 & \rho & \ldots & \rho^{d_s-1} \end{pmatrix} \right\} \tag{5.29}$$

with uniform probability. Note that for $\rho = 1/2$, we obtain PAM modulation according to (5.20).

A hybrid variant of SM-UPA and SM-EPA is SM-GPA, where groups of G bits share the same SM-UPA coefficient, i.e.,

$$\mathbf{a}^{\mathrm{T}} \in \mathcal{A}^{d_s,G}_{\mathrm{GPA},\mathbb{R},\rho} \triangleq \left\{ \bar{a}^{\mathrm{T}} \otimes \frac{1_G^{\mathrm{T}}}{\sqrt{G}} \boldsymbol{S}\boldsymbol{P} \,\middle|\, \boldsymbol{S} \in \mathcal{S}_{d_s}, \boldsymbol{P} \in \mathcal{P}_{d_s}, \bar{a} \in \mathcal{A}^{d_s/G}_{\mathrm{UPA},\mathbb{R},\rho} \right\} \tag{5.30}$$

with uniform probability. Note that for $G = 1$, $\mathcal{A}^{d_s,1}_{\mathrm{GPA},\mathbb{R},\rho} = \mathcal{A}^{d_s}_{\mathrm{UPA},\mathbb{R},\rho}$ and for $G = d_s$, $\mathcal{A}^{d_s,d_s}_{\mathrm{GPA},\mathbb{R},\rho} = \mathcal{A}^{d_s}_{\mathrm{EPA},\mathbb{R}}$. In particular, for $\rho = 0.5$ and $G \geq 3$, the resulting constellation is approximately Gaussian and therefore, capable of achieving capacity.

PSM

For phase shifted superposition modulation (PSM) [116], the complex edge coefficients are given by

$$\mathbf{a}^{\mathrm{T}} \in \mathcal{A}^{d_s,G}_{\mathrm{PSM}} \triangleq \left\{ \bar{\boldsymbol{a}}^{\mathrm{T}} \boldsymbol{S} \boldsymbol{P} \,\middle|\, \boldsymbol{S} \in \mathcal{S}_{d_s}, \boldsymbol{P} \in \mathcal{P}_{d_s}, \bar{\boldsymbol{a}}^{\mathrm{T}} = \frac{1}{\sqrt{d_s}} \begin{pmatrix} 1 & e^{\mathrm{j}\pi/d_s} & \dots & e^{\mathrm{j}\pi(d_s-1)/d_s} \end{pmatrix} \right\} \tag{5.31}$$

with uniform probability.

Abstract Modulations

In the previous subsections, we put coefficient models that were previously considered in the literature into the context of our random coefficient model. Note however, that there is no apparent reason as to why we should restrict ourselves to those models. We can thus consider coefficients that are *continuously distributed* which would result in unique coefficients for each SN. It is an interesting open problem to find coefficient distributions that are both capable of achieving the Shannon capacity[3] and perform well under iterative decoding.

5.2.5 Modulated MIMO Transmission over a Fading Channel

We consider transmission of modulated symbols $\boldsymbol{A}\boldsymbol{x}$ over a linear MIMO channel [96] with channel matrix $\mathcal{H} \in \mathbb{C}^{d'_y \times d_y}$, i.e.,

$$\boldsymbol{y}' = \boldsymbol{A}'\boldsymbol{x} + \boldsymbol{w} = \mathcal{H}\boldsymbol{A}\boldsymbol{x} + \boldsymbol{w}. \tag{5.32}$$

Hence, $\mathcal{H}\boldsymbol{A} = \boldsymbol{A}' \in \mathbb{C}^{d'_y \times d_s}$ can be seen to be the transformed edge coefficient matrix.

Let $\boldsymbol{a}_k \in \mathbb{C}^{d_y}$, $k = 1, \dots, d_s$ denote the columns of $\boldsymbol{A}$ and $\boldsymbol{a}'_k \in \mathbb{C}^{d'_y}$, $k = 1, \dots, d_s$, denote the columns of $\boldsymbol{A}'$. We then have

$$\boldsymbol{a}'_k = \mathcal{H}\boldsymbol{a}_k, \tag{5.33}$$

i.e., $\boldsymbol{A}'^{\neg k} = \mathcal{H}\boldsymbol{A}^{\neg k}$, and hence, for any $\mathcal{H}$, (5.12) holds for $p_{\mathbf{A}'}$ given that it holds for $p_{\mathbf{A}}$. The same is true for the permutation invariance (5.13). Thus, we can use our model to represent MIMO transmission with multiple bits per dimension. E.g., assume we use 4-QAM modula-

[3] i.e., the distributions are such that (5.18) tends towards the AWGN capacity for increasing d_s.

tion independently on each dimension, then

$$A \in \left\{ \frac{1}{\sqrt{2}} \begin{pmatrix} \pm 1 & \pm \mathrm{j} & 0 & 0 & \cdots & 0 \\ 0 & 0 & \pm 1 & \pm \mathrm{j} & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & \cdots & \pm 1 & \pm \mathrm{j} \end{pmatrix} \right\} \tag{5.34}$$

is block diagonal and of size $d_y \times 2d_y$. Multiplication with some general $\mathcal{H}$ then introduces dependencies in between different dimension, i.e., where as $y_1 = \pm x_1 \pm \mathrm{j} x_2 + w_1$ only depends on x_1 and x_2, y_1' generally depends on all input bits $x_1, \ldots, x_{d_s}$.

5.3 Decoding

5.3.1 Message Passing Decoding Algorithm

In what follows, we describe generic message passing decoding of an SMLDPC code. Taking into account irregular codes (cf. Section 5.6), let $\bar{\jmath} \in \mathcal{D}_\mathrm{s}$, $(i,k) \in \mathcal{D}_\mathrm{v}$ and $j \in \mathcal{D}_\mathrm{c}$ denote the SN, VN and CN degrees, respectively. For the regular codes we considered so far, $\mathcal{D}_\mathrm{s} = \{d_s\}$, $\mathcal{D}_\mathrm{v} = \{(d_{vs}, d_{vc})\}$ and $\mathcal{D}_\mathrm{c} = \{d_c\}$.

Input: All receive symbol subvectors $\boldsymbol{y}_l \in \mathcal{Y}^{d_y}$ as well as the corresponding modulation coefficient matrices $\boldsymbol{A}_l \in \mathcal{A}^{d_y \times \bar{\jmath}}$ for $l = 1, \ldots, L$. Furthermore, let $\mathscr{L} > 0$ denote the maximum number of decoding iterations.

Initialization: Set $\ell = 0$. Initialize all edges in the graph with zero messages, i.e., $\mu^{(0)}_{n \to l} = \mu^{(0)}_{n \to m}$, $n = 1, \ldots, N$, $l \in \mathcal{N}_{\mathrm{v}_n|\mathrm{s}}$ and $m \in \mathcal{N}_{\mathrm{v}_n|\mathrm{c}}$, where $\mathcal{N}_{\mathrm{v}_n|\mathrm{s}}$ and $\mathcal{N}_{\mathrm{v}_n|\mathrm{c}}$ denote the indices of the non zero elements in column n of $\boldsymbol{\Gamma}_A$ and $\boldsymbol{H}$, respectively.

Iteration:

1. CN and SN update. Note that these steps can be done in parallel at the same time.

   (a) SN update: For all SNs $l = 1, \ldots, L$, calculate the SN update for all adjacent VNs $n \in \mathcal{N}_{\mathrm{s}_l}$

   $$\underline{\Phi}^{(\ell)}_{\bar{\jmath}} : \mathcal{Y}^{d_y} \times \mathcal{M}_\ell^{\bar{\jmath}-1} \times \mathcal{A}^{d_y \times \bar{\jmath}-1} \times \mathcal{A}^{d_y} \to \underline{\mathcal{M}}_\ell, \qquad \underline{\Phi}^{(\ell)}_{\bar{\jmath}}(\boldsymbol{y}_l, \boldsymbol{\mu}^{(\ell)}_{l\backslash n}, \boldsymbol{A}_{l\backslash n}, \boldsymbol{a}_{l,n}) = \underline{\mu}^{(\ell)}_{l \to n} \tag{5.35}$$

   where $\boldsymbol{\mu}^{(\ell)}_{l\backslash n} \in \mathcal{M}_\ell^{\bar{\jmath}-1}$ comprises all messages from adjacent VNs except n, cf. Figure 5.5. In (5.35), we can split $\boldsymbol{A}_l$ into $\boldsymbol{A}_{l\backslash n} \in \mathcal{A}^{d_y \times \bar{\jmath}-1}$ which contains all the columns $\mathcal{N}_{\mathrm{s}_l} \setminus n$ and the column $\boldsymbol{a}_{l,n} \in \mathcal{A}^{d_y}$ corresponding to VN n, cf. (5.4). In other words, in contrast to the VN and CN update, the SN update is *not invariant*

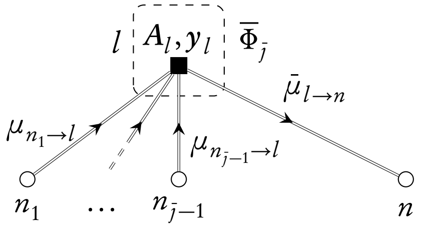


Figure 5.5: SN update for SN $\mathbf{s}_l$ with neighborhood $\mathcal{N}_{\mathbf{s}_l} = \{n_1, \dots, n_{\bar{j}-1}, n\}$

with respect to the index of the output message, as there are different edge coefficients involved with each of them and the SN update needs to take that into account.

(b) CN update. For $\ell = 0$, set $\overline{\mu}_{m\to n}^{(0)} = 0$. In case that $\ell > 0$, for all CNs $m = 1, \dots, M$ and for all adjacent VNs $n \in \mathcal{N}_{\mathrm{c}_m}$ calculate the CN update

$$\overline{\Phi}_j^{(\ell)} : \mathcal{M}_\ell^{j-1} \to \overline{\mathcal{M}}_\ell, \qquad \overline{\Phi}_j^{(\ell)}(\mu_{m\setminus n}^{(\ell)}) = \overline{\mu}_{m\to n}^{(\ell)} \tag{5.36}$$

where $\mu_{m\setminus n}^{(\ell)} \in \mathcal{M}_\ell^{j-1}$ comprises all messages from adjacent VNs except n, cf. Figure 2.3b.

2. VN update: For all VNs $n = 1, \dots, N$, calculate the APP LLRs $\mathrm{L}_n^{(\ell)} \in \mathcal{L}_\ell$ via

$$\Psi_{k,i}^{(\ell)} : \underline{\mathcal{M}}_\ell^k \times \overline{\mathcal{M}}_\ell^i \to \mathcal{L}_\ell \qquad \Psi_{k,i}^{(\ell)}(\underline{\mu}_n^{(\ell)}, \overline{\mu}_n^{(\ell)}) = \mathrm{L}_n^{(\ell)}, \tag{5.37}$$

where $\underline{\mu}_n^{(\ell)} \in \underline{\mathcal{M}}_\ell^k$ comprises all messages incident to VN n from adjacent SNs and $\overline{\mu}_n^{(\ell)} \in \overline{\mathcal{M}}_\ell^i$ comprises all messages incident to VN n from adjacent CNs.

Furthermore, for all VNs $n = 1, \dots, N$ and all adjacent SNs $l \in \mathcal{N}_{\mathrm{v}_n|\mathrm{s}}$ and CNs $m \in \mathcal{N}_{\mathrm{v}_n|\mathrm{c}}$, calculate the VN-to-SN update

$$\Phi_{k-1,i}^{(\ell)} : \underline{\mathcal{M}}_\ell^{k-1} \times \overline{\mathcal{M}}_\ell^i \to \mathcal{M}_{\ell+1} \qquad \Phi_{k-1,i}^{(\ell)}(\underline{\mu}_{n\setminus l}^{(\ell)}, \overline{\mu}_n^{(\ell)}) = \mu_{n\to l}^{(\ell+1)}, \tag{5.38}$$

as well as the VN-to-CN update

$$\Phi_{k,i-1}^{(\ell)} : \underline{\mathcal{M}}_\ell^k \times \overline{\mathcal{M}}_\ell^{i-1} \to \mathcal{M}_{\ell+1} \qquad \Phi_{k,i-1}^{(\ell)}(\underline{\mu}_n^{(\ell)}, \overline{\mu}_{n\setminus m}^{(\ell)}) = \mu_{n\to m}^{(\ell+1)}, \tag{5.39}$$

where $\underline{\mu}_{n\setminus l}^{(\ell)} \in \underline{\mathcal{M}}_\ell^{k-1}$ comprises all messages incident to VN n from adjacent SNs except l and $\overline{\mu}_{n\setminus m}^{(\ell)} \in \overline{\mathcal{M}}_\ell^{j-1}$ comprises all messages incident to VN n from adjacent CNs except m.

3. Set $\ell \mapsto \ell + 1$

4. If $\hat{x}_n = \mathrm{sign}(\mathrm{L}_n)$ fulfill the parity check equation or if $\ell = \mathscr{L}$, break and output the APP LLR and/or decoded bits $\hat{x}_n$. If not, continue with step 1.

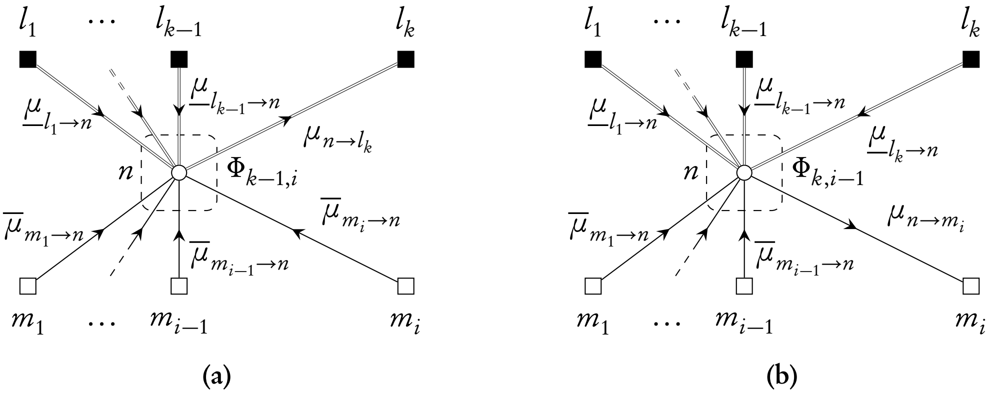

l_1
$\cdots$
l_{k-1}
l_k
$\underline{\mu}_{l_1\to n}$
$\underline{\mu}_{l_{k-1}\to n}$
$\mu_{n\to l_k}$
$\underline{\mu}_{l_k\to n}$
n
$\Phi_{k-1,i}$
$\Phi_{k,i-1}$
$\overline{\mu}_{m_1\to n}$
$\overline{\mu}_{m_{i-1}\to n}$
$\overline{\mu}_{m_i\to n}$
$\mu_{n\to m_i}$
m_1
m_{i-1}
m_i
(a)
(b)

Figure 5.6: VN-to-SN update (a) and VN-to-CN update (b) for VN v_n of degree (k,i) with SN side neighborhood $\mathcal{N}_{\mathrm{v}_n|\mathrm{s}} = \{l_1,\ldots,l_k\}$ and CN side neighborhood $\mathcal{N}_{\mathrm{v}_n|\mathrm{c}} = \{m_1,\ldots,m_i\}$.

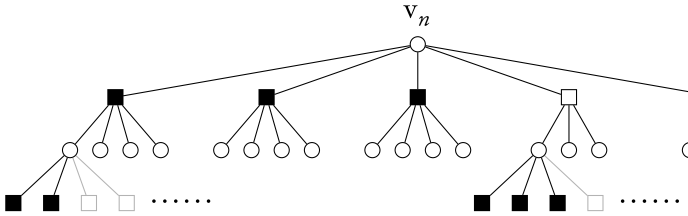


Figure 5.7: A tree-like neighborhood of a VN of depth $2\ell-1=3$, i.e., $\mathcal{N}_{\mathrm{v}_n}^{2\ell-1}$ is a tree. Here, $d_{vs}=3$, $d_{vc}=2$, $d_s=4$ and $d_c=3$. Note that the greyed-out leaf CNs do not affect the decoding process.

Note that this decoding schedule allows parallel demodulation and decoding. This can be seen by considering the corresponding decoding neighborhood of a VN. Figure 5.7 depicts a tree-like neighborhood of depth $2\ell-1=3$, covering $\ell=2$ decoder iterations. At any odd level, both SN and CN constraints are used for decoding.

In the most general case, the coefficient set $\mathcal{A}$, the set of channel outputs $\mathcal{Y}$, the SN, VN and CN message sets $\underline{\mathcal{M}}_\ell$, $\mathcal{M}_\ell$ and $\overline{\mathcal{M}}_\ell$ as well as the APP LLR set $\mathcal{L}_\ell$ are rings of arbitrary size. In what follows, we will focus on BP, where all the involved sets are equal to the real numbers $\mathbb{R}$. Since for a practical implementation, the involved quantities must be represented with finite precision, the generic definition is useful for working with finite sets and approximate BP-type algorithms, cf. Chapter 4.

5.3.2 MAP Detection and BP Decoding

Note that the global model (5.1) admits the factorization

$$p(x,y|A) = p(y|x,A)p(x) = \frac{1}{|\mathcal{C}|}\prod_{l=1}^{L} p(y_l|A_l,x_{\mathcal{N}_{\mathrm{s}_l}})\prod_{m=1}^{M}\mathbb{1}(\operatorname{par} x_{\mathcal{N}_{\mathrm{c}_m}}=+1), \tag{5.40}$$

i.e., a priori, we know that x must satisfy the parity check equations and we consider all codewords that do so equally likely. Hence, similarly to LDPC codes, the computation of $p(x_n|\, y,A)$ for MAP detection can be performed iteratively by the BP algorithm, assuming that there are no cycles. This means that when considering ℓ decoding iterations, the neighborhood of depth $2\ell-1$ of v_n, denoted by $\mathcal{N}_{\mathrm{v}_n}^{2\ell-1}$ is a tree, cf. Figure 5.7.

In particular, the VN updates[4] (5.38) and (5.39) and APP LLRs follow from the extrinsic information of a repetition code (2.58),

$$\Phi_{k,i}^{\mathrm{BP}}(\underline{\mu},\overline{\mu}) = \Psi_{k,i}^{\mathrm{BP}}(\underline{\mu},\overline{\mu}) = \sum_{l=1}^{k} \underline{\mu}_l + \sum_{m=1}^{i} \overline{\mu}_m. \tag{5.41}$$

The CN update (5.36) is identical to the one for LDPC codes (2.61).

The SN update (5.35) is characterized by the factors $p(y_l|A_l, x_{\mathcal{N}_{\mathrm{s}_l}})$ in (5.40) and corresponds to the extrinsic LLR of a SN. The a priori probabilities of the involved code bits $x_{\mathcal{N}_{\mathrm{s}_l}}$ are provided via the incoming messages using (2.55),

$$p_{\mathsf{x}_{\mathcal{N}_{\mathrm{s}_l}}}(x) = \prod_{n\in\mathcal{N}_{\mathrm{s}_l}} p_{\mathsf{x}_n}(x_n), \qquad p_{\mathsf{x}_n}(x) = \frac{e^{x\mu_{n\to l}/2}}{e^{\mu_{n\to l}/2} + e^{-\mu_{n\to l}/2}}. \tag{5.42}$$

Thus, the SN update can be calculated as

$$\underline{\Phi}_{\bar{\jmath}}^{\mathrm{BP}}(y,\mu,A_{\sim k},a_k) \overset{(5.40)}{=} \log \frac{p(y|x_k=+1,A)}{p(y|x_k=-1,A)} \tag{5.43}$$

$$\overset{(5.42)}{=} \log \frac{\sum\limits_{x'\in\mathcal{X}^{\bar{\jmath}}:x_i=+1} p(y|x',A)\exp(\mu^{\mathrm{T}}x'_{\sim k}/2)}{\sum\limits_{x'\in\mathcal{X}^{\bar{\jmath}}:x_i=-1} p(y|x',A)\exp(\mu^{\mathrm{T}}x'_{\sim k}/2)} \tag{5.44}$$

$$\overset{(5.3)}{=} \log \frac{\sum\limits_{x'\in\mathcal{X}^{\bar{\jmath}-1}} p_{\mathsf{w}}(y-A_{\sim k}x'-a_k)\exp(\mu^{\mathrm{T}}x'/2)}{\sum\limits_{x'\in\mathcal{X}^{\bar{\jmath}-1}} p_{\mathsf{w}}(y-A_{\sim k}x'+a_k)\exp(\mu^{\mathrm{T}}x'/2)} \tag{5.45}$$

$$\overset{(5.17)}{=} \log \frac{\sum\limits_{x'\in\mathcal{X}^{\bar{\jmath}-1}} \exp\left(-\frac{1}{N_0}\|y-A_{\sim k}x'-a_k\|^2 + \frac{\mu^{\mathrm{T}}x'}{2}\right)}{\sum\limits_{x'\in\mathcal{X}^{\bar{\jmath}-1}} \exp\left(-\frac{1}{N_0}\|y-A_{\sim k}x'+a_k\|^2 + \frac{\mu^{\mathrm{T}}x'}{2}\right)}. \tag{5.46}$$

In what follows, we will discuss possible strategies to evaluate (5.46).

5.3.3 Calculation of the BP SN Message Updates

For AWGN channels, we face the problem of finding a method to efficiently evaluate (5.46). Since the number of terms in (5.46) grows exponentially in the SN degree $\bar{\jmath}$, a direct application of (5.46) is limited to very small SN degrees. For higher degrees, we have to resort to

[4] Note that our choice of notation in (5.38) and (5.39) implies that the VN-to-SN update for a VN of degree (k,i) is identical to the VN-to-CN update of a VN of degree $(k-1, i+1)$. Given the repetition code interpretation of VNs, this assumption appears to be justified.

approximations or impose a special structure on the modulation edge coefficient matrix A in order to arrive at practically feasible update rules.

Jacobian Logarithm, Max-log Approximation and Soft Sphere Decoding

First, note that (5.46) is numerically challenging to evaluate since the terms in both numerator and denominator tend to be very small. For that reason, we can use the *Jacobian logarithm* [44]

$$\operatorname{jaclog}(a,b) \triangleq \log(e^a + e^b) = \max(a,b) + \underbrace{\log(1+e^{-|a-b|})}_{\triangleq r(|a-b|)}, \tag{5.47}$$

where the refinement $r(\cdot)$ can be stored in a LUT. We can then evaluate numerator and denominator in (5.46) separately by recursive application of the Jacobian logarithm because $\log(e^a + e^b + e^c) = \operatorname{jaclog}(a, \operatorname{jaclog}(b,c))$.

For a course approximation, we can neglect the refinement alltogether and arrive at the *max-log approximation*

$$\begin{aligned} \underline{\Phi}_{\bar{j}}^{\text{max-log}}(\boldsymbol{y},\mu,A_{\sim k},\boldsymbol{a}_k) = &- \min_{x\in\{-1,+1\}^{\bar{j}}} \left\{ \frac{1}{N_0} || \boldsymbol{y} - A_{\sim k}\boldsymbol{x} - \boldsymbol{a}_k ||^2 - \frac{\mu^{\mathrm{T}}\boldsymbol{x}}{2} \right\} \\ &+ \min_{x\in\{-1,+1\}^{\bar{j}}} \left\{ \frac{1}{N_0} || \boldsymbol{y} - A_{\sim k}\boldsymbol{x} + \boldsymbol{a}_k ||^2 - \frac{\mu^{\mathrm{T}}\boldsymbol{x}}{2} \right\}. \end{aligned} \tag{5.48}$$

The resulting integer-least squares problem can be solved using *sphere decoding* [31, 85, 95].

Trellis Demodulation

Depending on the coefficient distribution, the demodulation complexity can be reduced by employing the BCJR algorithm [5] for demodulation on an irregular trellis. This has first been observed in [64] for SM-EPA coefficients. The demodulation procedure exploits that for SM-EPA, only $d_s + 1$ different channel inputs are possible. For that case, the demodulation complexity of trellis demodulation is $O(d_s)$ per code bit. Interestingly, even for bijective modulation schemes where d_s bits are mapped to 2^{d_s} different symbols, this approach is capable of reducing the demodulation complexity from $O(2^{d_y})$ to $O(2^{d_y}/d_y)$ [46].

Since SM-GPA (cf. (5.30)) for $\rho = .5$ is a hybrid between bijective PAM and SM-EPA modulations, the demodulation complexity can be reduced to $O(G^2 2^{d_y/G}/d_y)$ for that case.

Gaussian Approximation

We focus on the local channel (5.3) and assume that the receive subvectors

$$\mathsf{y} = a_k x_k + A_{\sim k}\mathsf{x}_{\sim k} + \mathsf{w} = a_k x_k + \sum_{i\neq k}^{\bar{j}} a_i \mathsf{x}_i + \mathsf{w} \tag{5.49}$$

are conditionally Gaussian given x_k and A. We calculate the mean

$$\mathbb{E}\left[\mathbf{y}|x_k,A\right]=\boldsymbol{a}_k x_k+A_{\sim k}\mathbb{E}\left[\mathbf{x}_{\sim k}\right] \tag{5.50}$$

and covariance matrix

$$\operatorname{cov}\left[\mathbf{y}|x_k,A\right]=A_{\sim k}\operatorname{diag}\left\{1-\mathbb{E}\left[\mathsf{x}_i\right]^2\right\}_{\substack{i=1\\i\neq k}}^{\bar{j}}A_{\sim k}^{\mathrm{H}}+\boldsymbol{I}N_0. \tag{5.51}$$

using the expectation of the a priori bits

$$\mathbb{E}\left[\mathsf{x}_i\right]=p_{\mathsf{x}_i}(1)-p_{\mathsf{x}_i}(-1)\overset{(2.55)}{=}\tanh\frac{\mu_i}{2}. \tag{5.52}$$

Note that for $\mathbf{y}_i\sim\mathcal{CN}(\mu_i,\Sigma)$, $i=1,2$, with $\mu_1=\zeta+\eta$ and $\mu_2=\zeta-\eta$ we have

$$\begin{aligned}\log\frac{p_{\mathbf{y}_1}(y)}{p_{\mathbf{y}_2}(y)}&=(y-\mu_2)^{\mathrm{H}}\Sigma^{-1}(y-\mu_2)-(y-\mu_1)^{\mathrm{H}}\Sigma^{-1}(y-\mu_1) &(5.53)\\ &=\mu_2^{\mathrm{H}}\Sigma^{-1}\mu_2-\mu_1^{\mathrm{H}}\Sigma^{-1}\mu_1+2\operatorname{Re}\left\{(\mu_1-\mu_2)^{\mathrm{H}}\Sigma^{-1}y\right\} &(5.54)\\ &=4\operatorname{Re}\left\{\eta^{\mathrm{H}}\Sigma^{-1}(y-\zeta)\right\}. &(5.55)\end{aligned}$$

In our case, Σ is given by (5.51), where as for the mean, it follows from (5.50) and (5.52) that $\zeta=\sum_{i\neq k}^{\bar{j}}\tanh\frac{\mu_i}{2}\boldsymbol{a}_i$ and $\eta=\boldsymbol{a}_k$. Consequently, (5.53) becomes

$$\begin{aligned}&\underline{\Phi}_{\bar{j}}^{\mathrm{GAP-}\mathbb{C}}(y,\mu,A_{\sim k},\boldsymbol{a}_k)=\\ &\qquad 4\operatorname{Re}\left\{\boldsymbol{a}_k^{\mathrm{H}}\left(A_{\sim k}\operatorname{diag}\left\{1-\tanh^2\frac{\mu_i}{2}\right\}_{i\neq k}^{\bar{j}}A_{\sim k}^{\mathrm{H}}+\boldsymbol{I}N_0\right)^{-1}\left(y-\sum_{i\neq k}^{\bar{j}}\tanh\frac{\mu_i}{2}\boldsymbol{a}_i\right)\right\}.\end{aligned} \tag{5.56}$$

For real modulation and noise $\mathbf{w}\sim\mathcal{N}(\mathbf{0},\boldsymbol{I}N_0/2)$, (5.56) becomes

$$\begin{aligned}&\underline{\Phi}_{\bar{j}}^{\mathrm{GAP-}\mathbb{R}}(y,\mu,A_{\sim k},\boldsymbol{a}_k)=\\ &\qquad 2\boldsymbol{a}_k^{\mathrm{T}}\left(A_{\sim k}\operatorname{diag}\left\{1-\tanh^2\frac{\mu_i}{2}\right\}_{i\neq k}^{\bar{j}}A_{\sim k}^{\mathrm{T}}+\boldsymbol{I}N_0/2\right)^{-1}\left(y-\sum_{i\neq k}^{\bar{j}}\tanh\frac{\mu_i}{2}\boldsymbol{a}_i\right).\end{aligned} \tag{5.57}$$

The expressions (5.56) and (5.57) are similar to MMSE estimators using a priori information, cf. [103].

For $d_y=1$, (5.56) and (5.57) simplify to

$$\underline{\Phi}_{\bar{j}}^{\mathrm{GAP-}\mathbb{C}}(y,\mu,\boldsymbol{a}_{\sim k}^{\mathrm{T}},a_k)=\frac{4\operatorname{Re}\left\{a_k^*\left(y-\sum_{i\neq k}^{\bar{j}}a_i\tanh\left(\frac{\mu_i}{2}\right)\right)\right\}}{N_0+\sum_{i\neq k}^{\bar{j}}|a_i|^2\left(1-\tanh^2\left(\frac{\mu_i}{2}\right)\right)} \tag{5.58}$$

and

$$\underline{\Phi}_{\bar{j}}^{\mathrm{GAP}-\mathbb{R}}(y,\boldsymbol{\mu},\boldsymbol{a}_{\sim k}^{\mathrm{T}},a_k)=\frac{2a_k\left(y-\sum_{i\neq k}^{\bar{j}}a_i\tanh\left(\frac{\mu_i}{2}\right)\right)}{N_0/2+\sum_{i\neq k}^{\bar{j}}a_i^2\left(1-\tanh^2\left(\frac{\mu_i}{2}\right)\right)},\tag{5.59}$$

which have also been derived in [64, 116].

5.4 Random Code Ensembles and Concentration

In this section, we introduce a random code construction method which induces a uniform distribution on the set of of all tripartite graphs for any prescribed degree structure. We then show that for any random realization of code graph G, transmit sequence $\mathbf{x}$, modulation coefficient sequence $(\mathbf{A}_l)_{l=1}^{L}$ and channel noise $\mathbf{w}$, the number of incorrect APP LLRs after ℓ decoding iterations, denoted by $\mathsf{z}=\mathsf{z}\big(\mathsf{G},(\mathbf{A}_l),\mathbf{w},\mathbf{x},\ell\big)$ converges towards $NP_e^{(\ell)}$ in probability for $N\to\infty$, where $P_e^{(\ell)}$ denotes the expected fraction of incorrect APP LLRs after ℓ decoding iterations for a VN with tree-like decoding neighborhood of depth $2\ell-1$. Similar to the asymptotic theory of LDPC codes (cf. Section 2.2.2), this implies that by calculating $P_e^{(\ell)}$, we can characterize the performance of almost any code. Furthermore, note that the monotonicity for BP decoded LDPC codes extends to SMLDPC codes in a straight forward manner:

Theorem 10 (Monotonicity). *Let $p(y,A|\,x)=p(y|\,x,A)p(A)$ be the memoryless channel resulting from the local input-output relation* (5.3) *and the iid symmetric edge coefficient distribution $p(A)$ in Assumption 2. Note that since the receiver has perfect information of A, we can assume that the channel outputs pairs (y,A), i.e., A is part of the channel realization. Assume $p(y',A|\,x)$ is physically degraded with respect to $p(y,A|\,x)$, cf. Section 2.2.2. For a given code and BP decoding, let $P_e^{(\ell)}$ be the expected fraction of incorrect APP LLRs after the ℓth decoding iteration assuming tree-like neighborhoods and transmission over $p(y,A|\,x)$, and let $P_e^{(\ell)\prime}$ be the equivalent quantity for transmission over $p(y',A'|\,x)$. It then follows that $P_e^{(\ell)}\leq P_e^{(\ell)\prime}$.*

Proof: The proof follows from [78, Theorem 1] in a straightforward manner by noting that the argument of the original proof relies on the optimality of BP decoding, which is also true for our case. ■

Consequently, the threshold

$$\sigma^*=\sup\{\sigma:\lim_{\ell\to\infty}P_e^{(\ell)}(\sigma)=0\,\}\tag{5.60}$$

characterizes the highest noise variance $N_0^*=\sigma^{*2}$ of $\mathbf{w}$ for which reliable transmission is possible for *almost all SMLDPC codes* (adhering to the degree structure that is assumed when calculating $P_e^{(\ell)}$). We will discuss methods to find (5.60) in Section 5.7.1; code designs based on maximizing (5.60) are discussed in Sections 5.7.2 and 5.7.3.

For the most part, the remainder of this section is rather technical and builds the concepts required for the proof of the concentration theorem. We take a similar approach to the proof of the LDPC concentration theorem, cf. Theorem 3, [62, 78]. For readers familiar with that proof, we point out the key differences:

- Whereas [78] consider the number of incorrect VN-to-CN messages, for SMLDPC codes the choice of either VN-to-SN or VN-to-CN messages to define concentration and convergence appears arbitrary. Thus, we favoured considering the number of incorrect APP LLRs over arbitrarily choosing either the SN or CN side messages.

- Since we have 3 types of nodes in the decoding trees (cf. Figure 5.7) the trees are less homogeneous as compared to the decoding trees of LDPC codes that only contain 2 types of nodes. Hence, calculating exactly the number of a certain type of nodes in a tree becomes a tedious task. For that matter, we will use weak upper bounds by eliminating the node type with lower connectivity, cf. Figure 5.8 and Lemma 8. Consequently, the constants in the resulting concentration theorem can be improved upon by more careful examination of the decoding trees. However, in the limit $N \to \infty$, the value of the constants does not matter.

- In addition to code realizations G and channel realization $\mathbf{w}$, we need to extend the probability space by the edge coefficient realizations $(\mathbf{A}_l)_{l=1}^{L}$. Furthermore, in contrast to LDPC codes, for SMLDPC codes, given any fixed series of modulation coefficients $(A_l)_{l=1}^{L}$, z is not independent of the transmitted codeword $\mathbf{x}$. Hence, we further need to extend the probability space under consideration to include all transmit codewords.

Definition 4 (Regular Code Ensemble). *We define the regular ensemble $\mathcal{C}^N(d_s, \boldsymbol{d}_v, d_c)$ by the following construction: Assign to each SN d_s sockets and to each CN d_c sockets. The VNs are assigned d_{vs} sockets that connect to SNs and d_{vc} sockets that connect to CNs. Label the SN sockets and the VN-to-SN sockets separately with $1,\ldots,Nd_{vs}$. Similarly, label the CN sockets and the VN-to-CN sockets separately with $1,\ldots,Nd_{vc}$. Pick two permutations π_{vs} on $\{1,\ldots,Nd_{vs}\}$ and π_{vc} on $\{1,\ldots,Nd_{vc}\}$ with uniform probability from the product set of all $(Nd_{vs})!(Nd_{vc})!$ possible combinations of such permutations. The corresponding labeled tripartite graph is then defined by identifying VN-to-SN edges with pairs of sockets $\{(i,\pi_{vs}(i)), i=1,\ldots,Nd_{vs}\}$ and VN-to-CN edges with pairs of sockets $\{(i,\pi_{vc}(i)), i=1,\ldots,Nd_{vc}\}$. This induces a uniform distribution on the set of all of tripartite graphs $\mathcal{C}^N(d_s, \boldsymbol{d}_v, d_c)$.*

Definition 5 (Neighborhood of a node). *We define the neighborhood $\mathcal{N}_{\mathrm{u}}^{d}$ of depth d of a node* u *as the induced subgraph consisting of all nodes reached and edges traversed by paths of length at most d starting from* u *(including* u*).*

Definition 6 (Neighborhood of an edge). *We define the neighborhood $\mathcal{N}_{\mathrm{e}}^{d}$ of depth d of an edge* $\mathrm{e} = \{\mathrm{u}_1, \mathrm{u}_2\}$ *as*

$$\mathcal{N}_{\mathrm{e}}^{d} \triangleq \mathcal{N}_{\mathrm{u}_1}^{d} \cup \mathcal{N}_{\mathrm{u}_2}^{d} \tag{5.61}$$

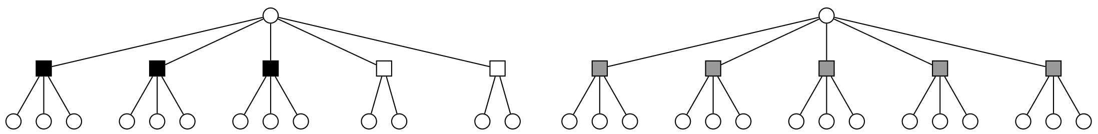


Figure 5.8: Left: Neighborhood of depth 2 of a VN for a graph with $d_{vs} = 3$, $d_{vc} = 2$, $d_s = 4$ and $d_c = 3$. Right: The same neighborhood, where the CNs and VNs have been replaced by nodes of degree $d_m = \max\{d_c, d_s\}$. Clearly, the number of nodes and edges of the right tree is greater or equal to the number of nodes and edges in the left tree.

Lemma 7 (Neighborhood Symmetry)**.** *We have*

$$\mathrm{u}_1 \in \mathcal{N}^d_{\mathrm{u}_2} \Leftrightarrow \mathrm{u}_2 \in \mathcal{N}^d_{\mathrm{u}_1}. \tag{5.62}$$

Furthermore, for any $\mathrm{e} \in \{\mathrm{u}_1, \mathrm{u}_2\}$

$$\mathrm{e} \in \mathcal{N}^d_{\mathrm{u}} \Leftrightarrow \mathrm{u} \in \mathcal{N}^{d-1}_{\mathrm{e}}. \tag{5.63}$$

Proof: (5.62) follows directly from the symmetry of the distance function.

To prove (5.63), assume $\mathrm{e} \in \mathcal{N}^d_{\mathrm{u}}$, which implies that either u_1 or u_2 can be reached by a path of length $d-1$ from u. It follows from the symmetry relation (5.62) that either $\mathrm{u} \in \mathcal{N}^{d-1}_{\mathrm{u}_1}$ or $\mathrm{u} \in \mathcal{N}^{d-1}_{\mathrm{u}_2}$ and hence, by (5.61) $\mathrm{u} \in \mathcal{N}^{d-1}_{\mathrm{e}}$.

Now assume $\mathrm{u} \in \mathcal{N}^{d-1}_{\mathrm{e}}$. It follows that there exists a path starting from either u_1 or u_2 and ending at u of length at most $d-1$. Reversing the orientation of the directed edges we obtain a path of length at most $d-1$ from u to either u_1 or u_2 to u. Thus, there exists a path of length at most d starting from u containing e, and hence it follows that $\mathrm{e} \in \mathcal{N}^d_{\mathrm{u}}$. ■

Lemma 8 (Bounds on the size of neighborhoods)**.** *The number of (distinct) nodes as well as the number of (distinct) edges in a neighborhood of a node or an edge of depth* 2ℓ *is upper-bounded by* $2(d_m d_v)^\ell$*, where* $d_m = \max\{d_c, d_s\}$ *and* $d_v = d_{vs} + d_{vc}$*.*

Proof: Note that the number of nodes and edges for any tree consisting of variable-, symbol- and check nodes with degrees $d_v = d_{vs} + d_{vc}$, d_s and d_c is less or equal to a tree containing only two types of nodes with degree d_v and $d_m = \max\{d_c, d_s\}$, cf. Figure 5.8. As stated in [78], Section II.B, for a graph consisting only of two types of nodes with degrees d_1 and d_2, the number of (distinct) nodes and edges in a neighborhood of a node or an edge of depth 2ℓ is upper-bounded by $2(d_1 d_2)^\ell$. The claim then follows by associating $d_1 \mapsto d_v$, $d_1 \mapsto d_m$. ■

Theorem 11 (Concentration Theorem)**.** *Over the probability space of all graphs* $\mathsf{G} \in \mathcal{C}^N(d_s, d_v, d_c)$*, iid edge coefficient matrices* $(\mathbf{A}_l)_{l=1}^L \in \mathcal{C}^{d_y \times d_s \times L}$*, iid channel noise* $\mathbf{w} \in \mathbb{C}^{d_y L}$ *and transmitted codewords* $\mathbf{x} \in \{1-2c \mid \mathbf{H}c = \mathbf{0}\}$ *with* $\mathbf{H}$ *induced by* G*, let* $\mathrm{z} = \mathrm{z}\big(\mathsf{G}, (\mathbf{A}_l), \mathbf{w}, \mathbf{x}, \ell\big)$ *be the number of incorrect APP LLRs after iteration* ℓ*. Let* $P_e^{(\ell)}$ *be the expected fraction of incorrect APP LLRs of a VN with a tree-like neighborhood of depth at least* $2\ell - 1$*. Then there exist positive constants* $\beta = \beta(d_s, d_{vs}, d_{vc}, d_c, \ell)$ *and* $\gamma = \gamma(d_s, d_{vs}, d_{vc}, d_c, \ell)$ *such that the following holds:*

Concentration around expected value: *For any $\epsilon > 0$, we have*

$$\mathrm{P}\{|\mathrm{z}-\mathbb{E}[\mathrm{z}]|>N\epsilon/2\}\leq 2e^{-\beta\epsilon^2 N}. \tag{5.64}$$

Convergence to cycle-free case: *For any $\epsilon > 0$ and $N > \frac{2\gamma}{\epsilon}$, we have*

$$|\mathbb{E}[\mathrm{z}]-NP_e^{(\ell)}|<N\epsilon/2. \tag{5.65}$$

Concentration around cycle-free case: *For any $\epsilon > 0$ and $N > \frac{2\gamma}{\epsilon}$, we have*

$$\mathrm{P}\left\{|\mathrm{z}-NP_e^{(\ell)}|>N\epsilon\right\}\leq 2e^{-\beta\epsilon^2 N}. \tag{5.66}$$

In order to prove Theorem 11, we first need to show the following

Lemma 9 (Probability of tree-like neighborhood)**.** *Over the probability space of all graphs $\mathrm{G} \in \mathcal{C}^N(d_s, d_v, d_c)$, we can bound the probability that the neighborhood of depth $2\ell - 1$ of some VN $\mathrm{v} \in \mathrm{G}$ is not tree-like as*

$$\mathrm{P}\left\{\mathcal{N}_{\mathrm{v}}^{2\ell-1}\ \textit{is not tree-like}\right\}\leq\frac{\gamma}{N}, \tag{5.67}$$

where γ depends on d_v, d_{vs}, d_{vc}, d_s and ℓ but not on N.

Proof of Lemma 9: First, note that the leaf CNs in a tree-like $\mathcal{N}_{\mathrm{v}}^{2\ell-1}$ do not affect the APP LLR at VN v at iteration ℓ, cf. Figure 5.7. However, it follows that if $\mathcal{N}_{\mathrm{v}}^{2\ell-1}$ is tree-like, then so is a version without leaf CNs. To have a higher degree of homogeneity throughout the tree, we include the leaf CNs in what follows.

Note that according to Lemma 8, we can bound the number of SNs S_ℓ, the number of VNs V_ℓ and the number of CNs C_ℓ in $\mathcal{N}_{\mathrm{v}}^{2\ell-1}$ as

$$V_\ell\leq 2(d_m d_v)^\ell,\qquad S_\ell\leq 2(d_m d_v)^\ell,\qquad C_\ell\leq 2(d_m d_v)^\ell \tag{5.68}$$

assuming it is tree-like.

Recall that M denotes the number of CNs and L denotes the number of SNs in the full graph. Fix ℓ and let $\ell' < \ell$. Assuming that $\mathcal{N}_{\mathrm{v}}^{2\ell'-1}$ is tree-like, we ask what is the probability that $\mathcal{N}_{\mathrm{v}}^{2\ell'}$ is tree-like? We obtain a bound by revealing the outgoing edges of the SN leaves of the tree given by $\mathcal{N}_{\mathrm{v}}^{2\ell'}$ one at a time and bounding the probability that this creates a loop. Assume that k additional VN-to-SN edges have been revealed at this stage without creating a loop; then the probability that the next revealed edge does not create a loop is

$$\frac{(N-V_{\ell'}-k)d_{vs}}{Nd_{vs}-V_{\ell'}d_{vs}-k}=1-\frac{k(1-\frac{1}{d_{vs}})}{N-(V_{\ell'}+\frac{k}{d_{vs}})}, \tag{5.69}$$

which is computed as the fraction of the number of unoccupied VN-to-SN sockets that are not yet part of the tree[5] over the total number of unoccupied VN-to-SN sockets. Note that

$$\frac{a}{N-b} \leq \frac{c}{N} \qquad a,b,c>0, \quad a<c \qquad \text{for } N \geq \frac{bc}{c-a}. \tag{5.70}$$

Furthermore, note that the number of edges revealed when going from $\mathcal{N}_{\mathrm{v}}^{2\ell'-1}$ to $\mathcal{N}_{\mathrm{v}}^{2\ell'}$ can be bounded by

$$k \leq d_v(d_m-1)^{\ell'+1}(d_v-1)^{\ell'} \leq d_v(d_m-1)^{\ell+1}(d_v-1)^{\ell} \tag{5.71}$$

and thus,

$$\frac{(N-V_{\ell'}-k)d_{vs}}{Nd_{vs}-V_{\ell'}d_{vs}-k} \geq 1-\frac{d_v(d_m-1)^{\ell+1}(d_v-1)^{\ell}}{N} \tag{5.72}$$

for sufficiently large N. Similarly, the probability that the revelation of a VN-to-CN edge creates a loop given that $\mathcal{N}_{\mathrm{v}}^{2\ell'-1}$ is tree-like can be bounded by (5.72). Thus, the probability that $\mathcal{N}_{\mathrm{v}}^{2\ell'}$ is tree-like, given that $\mathcal{N}_{\mathrm{v}}^{2\ell'-1}$ is tree-like, is lower-bounded by

$$\left(1-\frac{d_v(d_m-1)^{\ell+1}(d_v-1)^{\ell}}{N}\right)^{2(V_{\ell'+1}-V_{\ell'})}. \tag{5.73}$$

Assume now that $\mathcal{N}_{\mathrm{v}}^{2\ell'}$ is tree-like. As above, we reveal the outgoing SN side edges of the VN leafs of $\mathcal{N}_{\mathrm{v}}^{2\ell'}$ one at a time. Assuming that k SNs have been revealed without creating a loop, then the probability that the next revealed VN-to-SN edge does not create a loop can be bounded as

$$\frac{(L-S_{\ell'}-k)d_s}{Ld_s-S_{\ell'}d_s-k} = 1-\frac{k(1-\frac{1}{d_s})}{L-(S_{\ell'}+\frac{k}{d_s})} \geq 1-\frac{d_v(d_m-1)^{\ell+1}(d_v-1)^{\ell+1}}{L}. \tag{5.74}$$

Similarly, for the revelation of the CN side edges, the probability that the next revealed VN-to-CN edge does not create a loop is lower-bounded by

$$\frac{(M-C_{\ell'}-k)d_c}{Md_c-C_{\ell'}d_c-k} = 1-\frac{k(1-\frac{1}{d_c})}{M-(C_{\ell'}+\frac{k}{d_c})} \geq 1-\frac{d_v(d_m-1)^{\ell+1}(d_v-1)^{\ell+1}}{M}. \tag{5.75}$$

Thus, the probability that $\mathcal{N}_{\mathrm{v}}^{2\ell'+1}$ is tree-like, given that $\mathcal{N}_{\mathrm{v}}^{2\ell'}$ is tree-like, is lower-bounded by

$$\left(1-\frac{d_v(d_m-1)^{\ell+1}(d_v-1)^{\ell+1}}{L}\right)^{S_{\ell'+1}-S_{\ell'}}\left(1-\frac{d_v(d_m-1)^{\ell+1}(d_v-1)^{\ell+1}}{M}\right)^{C_{\ell'+1}-C_{\ell'}}. \tag{5.76}$$

By iteratively applying (5.73) and (5.76), the probability that $\mathcal{N}_{\mathrm{v}}^{2\ell-1}$ is tree-like is lower-bounded

[5] A socket is part of the tree when its corresponding node is part of the tree. Because all the sockets of non-leaf nodes are occupied, a cycle can only be created by connecting two leaf nodes.

by

$$\left(1-\frac{\overbrace{d_v(d_m-1)^{\ell+1}(d_v-1)^{\ell}}^{A}}{N}\right)^{2V_\ell}\left(1-\frac{\overbrace{d_v(d_m-1)^{\ell+1}(d_v-1)^{\ell+1}}^{B}}{L}\right)^{S_\ell}\left(1-\frac{B}{M}\right)^{C_\ell}. \tag{5.77}$$

Applying the binomial formula we calculate

$$\begin{aligned}
\mathrm{P}\left\{\mathcal{N}_{\mathrm{v}}^{2\ell-1}\text{ is not tree-like}\right\} &\leq 1-\left[\left(1-\frac{A}{N}\right)^{2V_\ell}\left(1-\frac{B}{L}\right)^{S_\ell}\left(1-\frac{B}{M}\right)^{C_\ell}\right] &(5.78)\\
&=1-\left[\sum_{i=1}^{2V_\ell}\sum_{j=1}^{S_\ell}\sum_{k=1}^{C_\ell}\binom{2V_\ell}{i}\binom{S_\ell}{j}\binom{C_\ell}{k}\left(\frac{A}{N}\right)^{i}\left(\frac{B}{L}\right)^{j}\left(\frac{B}{M}\right)^{k}(-1)^{i+j+k}\right]\\
&\leq\frac{2AV_\ell}{N}+\frac{BS_\ell}{L}+\frac{BC_\ell}{M} &(5.79)\\
&=\frac{2AV_\ell+\frac{d_s}{d_{vs}}BS_\ell+\frac{d_c}{d_{vc}}BC_\ell}{N} &(5.80)
\end{aligned}$$

where (5.79) holds for sufficiently large N and the claim follows from (5.80) due to (5.68) and (5.77). ■

We are now ready to prove the concentration theorem for SMLDPC codes.

Proof of Theorem 11: First note that it is sufficient to show (5.64) and (5.65), as (5.66) follows immediately from (5.64) and (5.65).

We start by proving (5.65). Let $\mathbb{E}[z_n], i\in\{1,\ldots,N\}$, be the expected fraction of incorrect APP LLRs at VN v_n, where the average is over all graphs, channel noise, edge coefficient matrices and codewords. Then, by linearity of expectation, by symmetry and by the iid property of the channel and the edge coefficients,

$$\mathbb{E}[z]=\sum_{n=1}^{N}\mathbb{E}[z_n]=N\mathbb{E}[z_1]. \tag{5.81}$$

Furthermore

$$\begin{aligned}
\mathbb{E}[z_1]=\mathbb{E}\left[z_1|\mathcal{N}_{\mathrm{v}_1}^{2\ell-1}\text{ is tree-like}\right]\mathrm{P}\left\{\mathcal{N}_{\mathrm{v}_1}^{2\ell-1}\text{ is tree-like}\right\}&\\
+\mathbb{E}\left[z_1|\mathcal{N}_{\mathrm{v}_1}^{2\ell-1}\text{is not tree-like}\right]\mathrm{P}\left\{\mathcal{N}_{\mathrm{v}_1}^{2\ell-1}\text{ is not tree-like}\right\}& &(5.82)
\end{aligned}$$

From Lemma 9,

$$\mathrm{P}\left\{\mathcal{N}_{\mathrm{v}_1}^{2\ell}\text{ is not tree-like}\right\}\leq\frac{\gamma}{N} \tag{5.83}$$

for some $\gamma>0$. Furthermore, we have

$$\mathbb{E}\left[z_1|\mathcal{N}_{\mathrm{v}_1}^{2\ell-1}\text{ is tree-like}\right]=P_e^{(\ell)} \tag{5.84}$$

by definition and

$$\left|\mathbb{E}\left[\mathsf{z}_1|\mathcal{N}_{\mathsf{v}_1}^{2\ell-1}\text{ is not tree-like}\right]\right| \leq 1 \tag{5.85}$$

trivially, hence

$$NP_e^{(\ell)}\left(1-\frac{\gamma}{N}\right) \leq \mathbb{E}[\mathsf{z}] \leq N\left(P_e^{(\ell)}+\frac{\gamma}{N}\right) \tag{5.86}$$

and

$$|\mathbb{E}[\mathsf{z}]-NP_e^{(\ell)}| < \gamma. \tag{5.87}$$

It follows that for $N > \frac{2\gamma}{\epsilon}$,

$$|\mathbb{E}[\mathsf{z}]-NP_e^{(\ell)}| < N\epsilon/2. \tag{5.88}$$

It remains to prove (5.64). Recall that z denotes the number of incorrect APP LLRs among all N VNs at the end of iteration ℓ for a particular $\left(\mathsf{G},(\mathbf{A}_l),\mathbf{w},\mathsf{x}\right) \in \Omega$, where Ω denotes the probability space. We show (5.64) by the method of bounded differences [71] invoking the Azuma-Hoeffding inequality:

Theorem 12 (Azuma-Hoeffding Inequality [4, 45]). *Let $\mathsf{z}_0, \mathsf{z}_1 \ldots, \mathsf{z}_q$ be a martingale sequence such that for $0 \leq i < q$*

$$|\mathsf{z}_{i+1}-\mathsf{z}_i| \leq \zeta_i$$

where the constant ζ_i may depend on i. Then, for all $q \geq 1$ and any $\epsilon > 0$

$$\mathrm{P}\left\{|\mathsf{z}_q-\mathsf{z}_0| \geq \epsilon\right\} \leq 2\exp\left(-\frac{\epsilon^2}{2\sum_{i=1}^{q}\zeta_i^2}\right). \tag{5.89}$$

Hence, we proceed to define a martingale such that $\mathsf{z}_0 = \mathbb{E}[\mathsf{z}]$ and $\mathsf{z}_q = \mathsf{z}$ and show that the difference between any two successive sequence elements is bounded by constants that are independent of N. For the definition of the martingale, consider a sequence $=_i$, $0 \leq i \leq q \triangleq (d_{vs}+d_{vc})N+L+N$ of equivalence relations on Ω by refinement, i.e., $\left(\mathsf{G}',(\mathbf{A}_l'),\mathbf{w}',\mathsf{x}'\right) =_i \left(\mathsf{G}'',(\mathbf{A}_l''),\mathbf{w}'',\mathsf{x}''\right)$ implies $\left(\mathsf{G}',(\mathbf{A}_l'),\mathbf{w}',\mathsf{x}'\right) =_{i-1} \left(\mathsf{G}'',(\mathbf{A}_l''),\mathbf{w}'',\mathsf{x}''\right)$. These equivalence classes are defined by partial equalities. In particular, suppose we expose the $d_{vs}N$ SN edges one at a time, i.e., at step $i \leq d_{vs}N$ we expose the particular SN socket $\pi_{vs}(i)$ which is connected to the ith VN socket. Similarly, for $d_{vs}N < i \leq (d_{vs}+d_{vc})N$, we expose the CN sockets one at a time. For $(d_{vs}+d_{vc})N < i \leq (d_{vs}+d_{vc})N+L$, we expose the L coefficient matrices $\mathbf{A}_l$ along with the corresponding noise component $\mathbf{w}_l$. Eventually, for $(d_{vs}+d_{vc})N+L < i \leq (d_{vs}+d_{vc})N+L+N$, we expose bits of the codeword x_n one at a time. Then we have $\left(\mathsf{G}',(\mathbf{A}_l'),\mathbf{w}',\mathsf{x}'\right) =_i \left(\mathsf{G}'',(\mathbf{A}_l''),\mathbf{w}'',\mathsf{x}''\right)$ iff the information revealed in the first i steps for both tuples is the same.

Now, define $\mathsf{z}_0, \mathsf{z}_1, \ldots, \mathsf{z}_p$ by

$$\mathsf{z}_i\left(\mathsf{G},(\mathbf{A}_l),\mathbf{w},\mathsf{x}\right) \triangleq \mathbb{E}\left[\mathsf{z}\left(\mathsf{G}',(\mathbf{A}_l'),\mathbf{w}',\mathsf{x}'\right)\,\middle|\,\left(\mathsf{G}',(\mathbf{A}_l'),\mathbf{w}',\mathsf{x}'\right) =_i \left(\mathsf{G},(\mathbf{A}_l),\mathbf{w},\mathsf{x}\right)\right]. \tag{5.90}$$

By construction, $\mathsf{z}_0, \mathsf{z}_1, \ldots, \mathsf{z}_q$ is a *Doob's Martingale Process*, where, as required, $\mathsf{z}_0 = \mathbb{E}[\mathsf{z}]$ and

$\mathsf{z}_q=\mathsf{z}$. Thus, we need to bound

$$\Big|\mathsf{z}_{i+1}\big(\mathsf{G},(\mathbf{A}_l),\mathbf{w},\mathbf{x}\big)-\mathsf{z}_i\big(\mathsf{G},(\mathbf{A}_l),\mathbf{w},\mathbf{x}\big)\Big|\le\zeta_i,\qquad i=0,\ldots,q-1 \tag{5.91}$$

with some suitable constants ζ_i which may depend on d_{vs},d_{vc},d_s,d_c and ℓ, but not on N.

We start out to prove (5.91) for $i=0,1,\ldots,d_{vs}N$, i.e., for the steps where we expose the SN edges. Recall that $\pi_{vs}(i)=j$ means that the ith VN socket is connected to the jth SN socket. Let $\mathcal{G}(\mathsf{G},i)\subset\mathcal{C}^N(d_s,d_{vs},d_{vc},d_c)$ be such that the first i edges are equal to the edges in G, i.e.,

$$\mathcal{G}(\mathsf{G},i)\triangleq\Big\{\mathsf{G}'\;\Big|\;\big(\mathsf{G}',(\mathbf{A}'_l),\mathbf{w}',\mathbf{x}'\big)=_i\big(\mathsf{G}',(\mathbf{A}'_l),\mathbf{w}',\mathbf{x}'\big)\Big\}. \tag{5.92}$$

Let $\mathcal{G}_j(\mathsf{G},i)$ be the subset of $\mathcal{G}(\mathsf{G},i)$ consisting of those graphs for which $\pi_{vs}(i+1)=j$. Thus, $\mathcal{G}(\mathsf{G},i)=\bigcup_j\mathcal{G}_j(\mathsf{G},i)$. Consequently, we can decompose

$$\begin{aligned}\mathsf{z}_i&=\mathbb{E}\big[\mathsf{z}\big(\mathsf{G}',(\mathbf{A}'_l),\mathbf{w}',\mathbf{x}'\big)\mid\mathsf{G}'\in\mathcal{G}(\mathsf{G},i)\big]\\&=\sum_{j=1}^{d_{vs}N}\mathbb{E}\big[\mathsf{z}\big(\mathsf{G}',(\mathbf{A}'_l),\mathbf{w}',\mathbf{x}'\big)\mid\mathsf{G}'\in\mathcal{G}_j(\mathsf{G},i)\big]\mathrm{P}\big\{\mathsf{G}'\in\mathcal{G}_j(\mathsf{G},i)|\mathsf{G}'\in\mathcal{G}(\mathsf{G},i)\big\}.\end{aligned} \tag{5.93}$$

Now note that $\mathsf{z}_{i+1}\big(\mathsf{G},(\mathbf{A}_l),\mathbf{w},\mathbf{x}\big)$ is equal to $\mathbb{E}\big[\mathsf{z}\big(\mathsf{G}',(\mathbf{A}'_l),\mathbf{w}',\mathbf{x}'\big)\mid\mathsf{G}'\in\mathcal{G}_j(\mathsf{G},i)\big]$ for some $j\in\Pi_i$ where $\Pi_i\subset\{1,\ldots,Nd_{vs}\}$ denotes the set of unoccupied SN sockets after revealing i edges of G, i.e., $\Pi_i=\{1,\ldots,Nd_{vs}\}\setminus\{\pi_{vs}(k):k\le i\}$. Hence

$$\begin{aligned}&\big|\mathsf{z}_{i+1}\big(\mathsf{G},(\mathbf{A}_l),\mathbf{w},\mathbf{x}\big)-\mathsf{z}_i\big(\mathsf{G},(\mathbf{A}_l),\mathbf{w},\mathbf{x}\big)\big|\\&\le\max_{j\in\Pi_i}\Big|\mathbb{E}\Big[\mathsf{z}\big(\mathsf{G}',(\mathbf{A}'_l),\mathbf{w}',\mathbf{x}'\big)\mid\mathsf{G}'\in\mathcal{G}_j(\mathsf{G},i)\Big]-\mathsf{z}_i\big(\mathsf{G},(\mathbf{A}_l),\mathbf{w},\mathbf{x}\big)\Big| &(5.94)\\&\le\max_{j,k\in\Pi_i}\left|\mathbb{E}\Big[\mathsf{z}\big(\mathsf{G}',(\mathbf{A}'_l),\mathbf{w}',\mathbf{x}'\big)\,\Big|\,\mathsf{G}'\in\mathcal{G}_j(\mathsf{G},i)\Big]-\mathbb{E}\Big[\mathsf{z}\big(\mathsf{G}',(\mathbf{A}'_l),\mathbf{w}',\mathbf{x}'\big)\,\Big|\,\mathsf{G}'\in\mathcal{G}_k(\mathsf{G},i)\Big]\right| &(5.95)\\&=\max_{j,k\in\Pi_i}\left|\mathbb{E}\Big[\mathsf{z}\big(\mathsf{G}',(\mathbf{A}'_l),\mathbf{w}',\mathbf{x}'\big)-\mathsf{z}\big(\phi_{j,k}(\mathsf{G}'),(\mathbf{A}'_l),\mathbf{w}',\mathbf{x}'\big)\,\Big|\,\mathsf{G}'\in\mathcal{G}_j(\mathsf{G},i)\Big]\right| &(5.96)\\&\le\max_{j,k\in\Pi_i}\mathbb{E}\Big[\big|\mathsf{z}\big(\mathsf{G}',(\mathbf{A}'_l),\mathbf{w}',\mathbf{x}'\big)-\mathsf{z}\big(\phi_{j,k}(\mathsf{G}'),(\mathbf{A}'_l),\mathbf{w}',\mathbf{x}'\big)\big|\,\Big|\,\mathsf{G}'\in\mathcal{G}_j(\mathsf{G},i)\Big] &(5.97)\\&\le\max_{j,k\in\Pi_i,\mathsf{G}',(\mathbf{A}'_l),\mathbf{w}',\mathbf{x}'}\Big|\mathsf{z}\big(\mathsf{G}',(\mathbf{A}'_l),\mathbf{w}',\mathbf{x}'\big)-\mathsf{z}\big(\phi_{j,k}(\mathsf{G}'),(\mathbf{A}'_l),\mathbf{w}',\mathbf{x}'\big)\Big| &(5.98)\\&\le 8(d_m d_v)^{\ell-1}. &(5.99)\end{aligned}$$

Here, (5.95) follows from the decomposition (5.93) and the fact that the expectation of a random variable is always larger or equal to its minimum value. To show (5.96), define a map $\phi_{j,k}:\mathcal{G}_j(\mathsf{G},i)\to\mathcal{G}_k(\mathsf{G},i)$ as follows. Let π be the permutation defining the symbol edge assignment for a given graph $\mathsf{H}\in\mathcal{G}_j(\mathsf{G},i)$ and let $i'=\pi^{-1}(k)$. Define a permutation $\pi'=\pi$ except that $\pi'(i+1)=k$ and $\pi'(i')=j$. Let H' denote the resulting graph, i.e., $\mathsf{H}'=\phi_{j,k}(\mathsf{H})$ is the same graph as H, except that the edge connected to VN socket $i+1$ now is attached to SN socket k instead of j and the edge incident to SN socket k is attached to SN socket j. Clearly, $\phi_{j,k}$ is a bijection and, since graphs in the ensemble have uniform probability, such a bijection preserves probabilities and thus

$$\mathbb{E}\Big[\mathsf{z}\big(\mathsf{G}',(\mathbf{A}'_l),\mathbf{w}',\mathbf{x}'\big)\,\Big|\,\mathsf{G}'\in\mathcal{G}_k(\mathsf{G},i)\Big]=\mathbb{E}\Big[\mathsf{z}\big(\phi_{j,k}(\mathsf{G}'),(\mathbf{A}'_l),\mathbf{w}',\mathbf{x}'\big)\,\Big|\,\mathsf{G}'\in\mathcal{G}_j(\mathsf{G},i)\Big]. \tag{5.100}$$

The steps (5.97) and (5.98) follow because $|\mathbb{E}[\mathsf{x}]|\leq\mathbb{E}[|\mathsf{x}|]$ and the fact that the expectation of a random variable is less or equal to its maximum value. Finally, to see that (5.99) holds, note that for any VN v, the APP LLR at the end of iteration ℓ is only a function of the decoding neighborhood $\mathcal{N}_{\mathrm{v}}^{2\ell-1}$. Therefore, the LLR is only affected by an exchange of the endpoints of two edges if one or both of the edges are in $\mathcal{N}_{\mathrm{v}}^{2\ell-1}$, i.e., if $\mathrm{e}_i\in\mathcal{N}_{\mathrm{v}}^{2\ell-1}$, where e_i, $i=1,\ldots,4$ denotes the two edges attached to the sockets j and k before and after the change of sockets by $\phi_{j,k}$. By the symmetry relation (5.63), this is equivalent to $\mathrm{v}\in\mathcal{N}_{\mathrm{e}_i}^{2\ell-2}$ and hence, the difference in APP LLR errors in (5.98) can be bounded by 4 times the number of nodes in $\mathcal{N}_{\mathrm{e}_i}^{2\ell-2}$, which, according to Lemma 8 is equal to $8(d_m d_v)^{\ell-1}$. This proves (5.91) with $\zeta_i=8(d_m d_v)^{\ell-1}$ for $0=1,\ldots,Nd_{v,s}$.

The proof of (5.91) for $Nd_{vs}<i\leq N(d_{vs}+d_{vc})$, i.e., for the steps where the CN edges are revealed follows the very same argument as for the SN edges where again $\zeta_i=8(d_m d_v)^{\ell-1}$ in (5.91).

For $N(d_{vs}+d_{vc})<i\leq N(d_{vs}+d_{vc})+L$, corresponding to $l=i-N(d_{vs}+d_{vc})\in\{1,\ldots,L\}$, we reveal the edge coefficient matrices $\mathbf{A}_l$ and noise $\mathbf{w}_l$ one at a time. Let $\mathcal{A}_l\subset\Omega$ be defined as

$$\mathcal{A}_l\triangleq\{(\mathsf{G}',(\mathbf{A}'_k),\mathbf{w}',\mathbf{x}')\,|\,(\mathsf{G}',(\mathbf{A}'_k),\mathbf{w}',\mathbf{x}')=_i(\mathsf{G},(\mathbf{A}_k),\mathbf{w},\mathbf{x})\}, \tag{5.101}$$

and let z' be shorthand for $\mathsf{z}'=\mathsf{z}\big(\mathsf{G}',(\mathbf{A}'_l),\mathbf{w}',\mathbf{x}'\big)$ then for $N(d_{vs}+d_{vc})<i\leq N(d_{vs}+d_{vc})+L$, we can decompose

$$\mathsf{z}_i=\mathbb{E}\big[\mathsf{z}'\,|\,\mathcal{A}_l\big]=\mathbb{E}_{\mathbf{A}'_{l+1},\mathbf{w}'_{l+1}}\Big[\mathbb{E}\big[\mathsf{z}'\,|\,\mathcal{A}_l,\mathbf{A}'_{l+1},\mathbf{w}'_{l+1}\big]\,\Big|\,\mathcal{A}_l\Big]. \tag{5.102}$$

Thus, similar to (5.95) to (5.99), we can bound

$$\begin{aligned}
&\big|\mathsf{z}_{i+1}\big(\mathsf{G},(\mathbf{A}_l),\mathbf{w},\mathbf{x}\big)-\mathsf{z}_i\big(\mathsf{G},(\mathbf{A}_l),\mathbf{w},\mathbf{x}\big)\big|\\
&\leq\max_{\substack{(w_1,A_1)\\(w_2,A_2)}}\Big|\mathbb{E}\big[\mathsf{z}'\,|\,\mathcal{A}_l,\mathbf{A}'_{l+1}=A_1,\mathbf{w}'_{l+1}=w_1\big]-\mathbb{E}\big[\mathsf{z}'\,|\,\mathcal{A}_l,\mathbf{A}'_{l+1}=A_2,\mathbf{w}'_{l+1}=w_2\big]\Big| &(5.103)\\
&=\max_{(w_1,A_1),(w_2,A_2)}\Big|\mathbb{E}_{(\mathbf{A}_k)_{k=l+2}^{L},(\mathbf{w}_k)_{k=l+2}^{L},\mathbf{x}'}\big[\mathsf{z}(A_1,w_1,l+1)-\mathsf{z}(A_2,w_2,l+1)\big]\Big| &(5.104)\\
&\leq\max_{(w_1,A_1),(w_2,A_2),(\mathsf{G}',(\mathbf{A}'_k),\mathbf{w}',\mathbf{x}')\in\mathcal{A}_l}\big|\mathsf{z}(A_1,w_1,l+1)-\mathsf{z}(A_2,w_2,l+1)\big| &(5.105)\\
&\leq 2(d_v d_m)^{\ell}. &(5.106)
\end{aligned}$$

Here, (5.103) follows from the decomposition (5.102) and the fact that $\mathsf{z}_{i+1}=\mathbb{E}\big[\mathsf{z}'\,|\,\mathcal{A}_l,\mathbf{A}'_{l+1},\mathbf{w}'_{l+1}\big]$

for some $(\mathbf{A}'_{l+1}, \mathbf{w}'_{l+1})$. To show (5.104) let

$$\mathsf{z}(A, w, l) \triangleq \mathsf{z}\big(\mathsf{G}, (\mathbf{A}_k)_{k=1}^{l-1}, A, (\mathbf{A}'_k)_{k=l+1}^{L}), (\mathbf{w}_k)_{k=1}^{l-1}, w, (\mathbf{w}'_k)_{k=l+1}^{L}), \mathbf{x}'\big), \tag{5.107}$$

i.e., the number of APP LLR errors assuming that for the lth SN, the edge coefficient and noise realization are equal to A and w, respectively. (5.104) follows by noting that both the edge coefficient matrices and noise components are iid which allows for both terms to be collected under the same expectation operator. To see why (5.106) holds, note that $\max\big|\mathsf{z}(A_1, w_1, l+1) - \mathsf{z}(A_2, w_2, l+1)\big|$ is the maximum difference in APP LLR errors if for the single SN $\mathbf{s}_{l+1}$, the edge coefficient and noise realizations are changed from (A_1, w_1) to (A_2, w_2) or vice versa. To find an upper bound for that difference, note that the APP LLR of a particular VN v is only affected by the change if $\mathbf{s}_{l+1} \in \mathcal{N}_{\mathrm{v}}^{2\ell-1}$, which by Lemma 7, is equivalent to $\mathrm{v} \in \mathcal{N}_{\mathbf{s}_{l+1}}^{2\ell-1}$. By Lemma 8, the number of affected VN can thus be bounded by $2(d_v d_m)^\ell$, which proves (5.91) with $\zeta_i = 2(d_v d_m)^\ell$ for $N(d_{vs} + d_{vc}) < i \leq N(d_{vs} + d_{vc}) + L$.

For the last N steps, $N(d_{vs} + d_{vc}) + L < i \leq N(d_{vs} + d_{vc}) + L + N$ correponding to $n = i - N(d_{vs} + d_{vc}) - L \in \{1, \ldots, N\}$, the code bits x_n are revealed one at a time. Let $\mathsf{z}' = \mathsf{z}\big(\mathsf{G}, (\mathbf{a}_l), \mathbf{w}, \mathbf{x}'\big)$ and $\mathcal{X}_n = \{\mathbf{x}' = (\mathsf{x}'_1, \ldots \mathsf{x}'_N) \,|\, (\mathsf{x}'_k)_{k=1}^n = (\mathsf{x}_k)_{k=1}^n\}$. Then

$$\begin{aligned}
&\big|\mathsf{z}_{i+1}\big(\mathsf{G}, (\mathbf{A}_l), \mathbf{w}, \mathbf{x}\big) - \mathsf{z}_i\big(\mathsf{G}, (\mathbf{A}_l), \mathbf{w}, \mathbf{x}\big)\big| \\
&\quad \leq \max_{x_1, x_2} \Big|\mathbb{E}_{\mathbf{x}'}\big[\mathsf{z}' \mid x' \in \mathcal{X}_n, \mathsf{x}'_{n+1} = x_1\big] - \mathbb{E}_{\mathbf{x}'}\big[\mathsf{z}' \mid \mathbf{x}' \in \mathcal{X}_n, \mathsf{x}'_{n+1} = x_2\big]\Big| && (5.108) \\
&\quad = \Big|\mathbb{E}_{\mathbf{x}'}\big[\mathsf{z}' \mid \mathbf{x}' \in \mathcal{X}_n, \mathsf{x}'_{n+1} = +1\big] - \mathbb{E}_{\mathbf{x}'}\big[\mathsf{z}' \mid \mathbf{x}' \in \mathcal{X}_n, \mathsf{x}'_{n+1} = -1\big]\Big| && (5.109) \\
&\quad = \Big|\mathbb{E}_{\mathbf{x}'}\big[\mathsf{z}\big(\mathsf{G}, (\mathbf{A}_l), \mathbf{w}, \mathbf{x}'\big) - \mathsf{z}\big(\mathsf{G}, (\mathbf{A}_l), \mathbf{w}, \mathbf{x}'^{\neg n+1}\big) \mid \mathbf{x}' \in \mathcal{X}_n, \mathsf{x}'_{n+1} = +1\big]\Big| && (5.110) \\
&\quad \leq \max_{\mathbf{x}' \in \mathcal{X}_n} \Big|\mathsf{z}\big(\mathsf{G}, (\mathbf{A}_l), \mathbf{w}, \mathbf{x}'\big) - \mathsf{z}\big(\mathsf{G}, (\mathbf{A}_l), \mathbf{w}, \mathbf{x}'^{\neg n+1}\big)\Big| && (5.111) \\
&\quad \leq 2(d_v d_m)^\ell && (5.112)
\end{aligned}$$

Once again, we start out by conditioning on the information revealed in the next step $i+1$ and bound the maximum difference to arrive at (5.108). (5.109) then follows since the code bits are binary and the maximum difference is attained for different bits. Next, (5.110) follows because

$$\begin{aligned}
&\mathbb{E}_{\mathbf{x}'}\Big[\mathsf{z}\big(\mathsf{G}, (\mathbf{A}_l), \mathbf{w}, \mathbf{x}'\big) \,\Big|\, \mathbf{x}' \in \mathcal{X}_n, \mathsf{x}_{n+1} = -1\Big] \\
&\qquad\qquad = \mathbb{E}_{\mathbf{x}'}\Big[\mathsf{z}\big(\mathsf{G}, (\mathbf{A}_l), \mathbf{w}, \mathbf{x}'^{\neg n+1}\big) \,\Big|\, \mathbf{x}' \in \mathcal{X}_n, \mathsf{x}_{n+1} = +1\Big], && (5.113)
\end{aligned}$$

thus allowing for both terms to be collected under the same expectation operator. Eventually, (5.112) bounds the maximum change in APP LLR errors between two codewords that only differ at digit $n+1$. That is, the APP LLR of a particular VN v is only affected if $\mathrm{v}_{n+1} \in \mathcal{N}_{\mathrm{v}}^{2\ell-1} \Leftrightarrow \mathrm{v} \in \mathcal{N}_{\mathrm{v}_{n+1}}^{2\ell-1}$, where the numer of VNs in $\mathcal{N}_{\mathrm{v}_{n+1}}^{2\ell-1}$ can be upper bounded due to Lemma 8. This proves (5.91) with $\zeta_i = 2(d_v d_m)^\ell$ for $N(d_{vs} + d_{vc}) + L < i \leq N(d_{vs} + d_{vc}) + L + N$.

Thus, we have shown (5.64), where according to (5.89), (5.99), (5.106) and (5.112), for the constant β in (5.64),

$$\frac{1}{\beta} = 8(d_v d_m)^{2\ell-2}\left(16 d_v + \frac{d_v^2 d_m^2 d_{vc}}{d_s} + d_v^2 d_m^2\right), \tag{5.114}$$

which concludes the proof of Theorem 11. ∎

5.5 Analysis of the Cycle-Free Case

In the following, we will establish fundamental results that help to determine the threshold as defined in (5.60). First, we show that for tree-like decoding neighborhoods and fairly general symmetry constraints, the messages passed along the edges of the graph are conditionally iid and symmetric. Whereas for LDPC codes this is true in general, for SMLDPC codes such a statement only holds when we consider the average over edge coefficient realizations. The symmetry of the densities then implies that the fraction of APP LLR errors conditioned on a specific codeword x, denoted by $P_e^{(\ell)}(x)$, is independent of the codeword. Specifically, this means that we can obtain $P_e^{(\ell)}$ as defined in Theorem 11 without averaging over codewords by simply conditioning on an arbitrary codeword, $P_e^{(\ell)} = P_e^{(\ell)}(x)$. Consistent with the theory for LDPC codes, we choose to condition on the all-one word $x = +1$. Under this assumption, the messages at any iteration are characterized by scalar, real probability densities. We then proceed to show that for BP decoding, on top of being symmetric, these message densities are also LLR-consistent, cf. (2.85) and (2.86). This result allows for an equivalent reformulation of the threshold (5.60) in terms of the mutual information between the code bits and the APP LLRs. Thus, under the assumption that we can calculate this mutual information exactly using EXIT analysis (cf. Section 2.3), we are guaranteed to obtain the same threshold as we would get from density evolution and exact calculation of $P_e^{(\ell)}$. This is crucial since in contrast to LDPC codes, an exact recursion of densities in the fashion of (2.83) is difficult to obtain due to the complexity of the SN updates. Lastly, we outline an accurate, low complexity method to obtain the EXIT function of an arbitrary LLR-consistent SN update.

5.5.1 Symmetric Message Passing

Assumption 3 (Symmetric message passing decoding for SMLDPC codes)**.**

- ***Check node symmetry:*** *for all ℓ, the CN updates* (5.36) *fulfill the symmetry property* (2.72).
- ***Variable node symmetry:*** *Sign inversion invariance of VN message updates* (5.38) *and* (5.39)

$$\Phi_{k,i}^{(\ell)}(-\underline{\mu}, -\overline{\mu}) = -\Phi_{k,i}^{(\ell)}(\underline{\mu}, \overline{\mu}), \tag{5.115}$$

and APP LLR updates (5.37)

$$\Psi_{k,i}^{(\ell)}(-\underline{\mu}, -\overline{\mu}) = -\Psi_{k,i}^{(\ell)}(\underline{\mu}, \overline{\mu}). \tag{5.116}$$

- ***Symbol node symmetry:*** *The SN updates* (5.35) *fulfill*

$$\underline{\Phi}_{\bar{j}}^{(\ell)}(y,\mu,A_{\sim k},-a_k) = -\underline{\Phi}_{\bar{j}}^{(\ell)}(y,\mu,A_{\sim k},a_k), \tag{5.117}$$

$$\underline{\Phi}_{\bar{j}}^{(\ell)}(y,\mu^{\neg i},A_{\sim k}^{\neg i},a_k) = \underline{\Phi}_{\bar{j}}^{(\ell)}(y,\mu,A_{\sim k},a_k). \tag{5.118}$$

The VN and CN symmetry properties are well known from LDPC decoding, cf. Assumption 1. The SN symmetries (5.117) and (5.118) are fulfilled by both BP message updates (5.45) and Gaussian approximation updates (5.56) and (5.57). Note that in case of symmetric noise densities, i.e., $p_{\mathsf{w}}(-w) = p_{\mathsf{w}}(w)$, another symmetry property can be shown to hold for the BP update (5.45),

$$\underline{\Phi}_{\bar{j}}^{(\ell)}(-y,0,A_{\sim k},a_k) = -\underline{\Phi}_{\bar{j}}^{(\ell)}(y,0,A_{\sim k},a_k). \tag{5.119}$$

Since we mainly consider AWGN channels, we could in principal assume that (5.119) holds. However, (5.119) is not required for the following result and thus, symmetric noise is not a prerequisite for symmetric message passing.

Lemma 10. *Consider the output of a SN update fulfilling the symmetry conditions* (5.117) *and* (5.118) *and let*

$$\overline{\mathsf{\mu}}_k = \underline{\Phi}_{\bar{j}}(\mathbf{\mathsf{y}},\mathsf{\mu},\mathbf{\mathsf{A}}_{\sim k},\mathbf{\mathsf{a}}_k), \tag{5.120}$$

where the local SN constraint (5.3) *holds according to*

$$\mathbf{\mathsf{y}} = \mathbf{\mathsf{A}}_{\sim k}x_{\sim k} + \mathbf{\mathsf{a}}_k x_k + \mathbf{\mathsf{w}}, \tag{5.121}$$

with $x \in \{-1,+1\}^{d_s}$ *and* $x_{\sim k} \in \{-1,+1\}^{d_s-1}$ *being the subvector of* x *not containing* x_k*. Let* $\mathbf{\mathsf{A}}$ *be distributed such that the symmetry relations* (5.12) *and* (5.13) *hold. Furthermore, for the conditional density of the input messages, we assume*

$$p_{\mathsf{\mu}|\mathsf{x}}(\mu^{\neg i}|x) = p_{\mathsf{\mu}|\mathsf{x}}(\mu|x^{\neg i}). \tag{5.122}$$

Then, the following statements hold.

1. *The density of the output message* $\overline{\mathsf{\mu}}_k$ *given the kth bit is conditionally independent of all other* $d_s - 1$ *code bits involved in the SN constraint,*

$$p_{\overline{\mathsf{\mu}}_k|\mathsf{x}}(\overline{\mu}|x) = p_{\overline{\mathsf{\mu}}_k|\mathsf{x}_k}(\overline{\mu}|x). \tag{5.123}$$

2. *The density of the output message* $\overline{\mathsf{\mu}}_k$ *conditioned on the corresponding code bit is symmetric in the sense that*

$$p_{\overline{\mathsf{\mu}}_k|\mathsf{x}_k}(-\overline{\mu}|x) = p_{\overline{\mathsf{\mu}}_k|\mathsf{x}_k}(\overline{\mu}|-x). \tag{5.124}$$

3. *The density $p_{\overline{\mu}_k|\mathsf{x}_k}$ is independent of k, i.e., for any $i,k \in \{1,\dots,\bar{j}\}$,*

$$p_{\overline{\mu}_k|\mathsf{x}_k}(\overline{\mu}|x) = p_{\overline{\mu}_i|\mathsf{x}_i}(\overline{\mu}|x) = p_{\overline{\mu}|\mathsf{x}}(\overline{\mu}|x). \tag{5.125}$$

Proof: We start out by proving (5.123). For all $i \neq k$ and all $\mathcal{M} \subseteq \mathbb{R}$

$$\begin{aligned}
\mathrm{P}\{\overline{\mu}_k \in \mathcal{M}|\mathbf{x}=x, \mathbf{A}=A\} &= \mathrm{P}\left\{(A\mathbf{x}+\mathbf{w}, \mu, A_{\sim k}, a_k) \in \underline{\Phi}_{\bar{j}}^{-1}(\mathcal{M})|\mathbf{x}=x\right\} \\
&\overset{(5.118)}{=} \mathrm{P}\left\{(A^{\neg i}\mathbf{x}^{\neg i}+\mathbf{w}, \mu^{\neg i}, A^{\neg i}_{\sim k}, a_k) \in \underline{\Phi}_{\bar{j}}^{-1}(\mathcal{M})|\mathbf{x}=x\right\} \\
&\overset{(5.122)}{=} \mathrm{P}\left\{(A^{\neg i}\mathbf{x}+\mathbf{w}, \mu, A^{\neg i}_{\sim k}, a_k) \in \underline{\Phi}_{\bar{j}}^{-1}(\mathcal{M})|\mathbf{x}=x^{\neg i}\right\} \\
&= \mathrm{P}\left\{\overline{\mu}_k \in \mathcal{M}|\mathbf{x}=x^{\neg i}, \mathbf{A}=A^{\neg i}\right\}.
\end{aligned}$$

Consequently, by the law of total probability,

$$\begin{aligned}
\mathrm{P}\{\overline{\mu}_k \in \mathcal{M}|\mathbf{x}=x\} &= \int_{\mathbb{C}^{d_y \times \bar{j}}} \mathrm{P}\{\overline{\mu}_k \in \mathcal{M}|\mathbf{x}=x, \mathbf{A}=A\}\, p_{\mathbf{A}}(A)\mathrm{d}A \\
&\overset{(5.12)}{=} \int_{\mathbb{C}^{d_y \times \bar{j}}} \mathrm{P}\left\{\overline{\mu}_k \in \mathcal{M}|\mathbf{x}=x^{\neg i}, \mathbf{A}=A^{\neg i}\right\} p_{\mathbf{A}}(A^{\neg i})\mathrm{d}A \\
&= \mathrm{P}\left\{\overline{\mu}_k \in \mathcal{M}|\mathbf{x}=x^{\neg i}\right\} \qquad \forall \mathcal{M}, i \neq k.
\end{aligned}$$

Since x is binary, (5.123) follows. To prove (5.124), note that for any $\mathcal{M} \subset \mathbb{R}$

$$\begin{aligned}
&\mathrm{P}\{-\overline{\mu}_k \in \mathcal{M}|\mathsf{x}_k = x_k\} \\
&= \int_{\mathbb{C}^{d_y \times \bar{j}}} \mathrm{P}\{-\overline{\mu}_k \in \mathcal{M}|\mathsf{x}_k = x_k, \mathbf{A}=A\}\, p_{\mathbf{A}}(A)\mathrm{d}A \\
&\overset{(5.117)}{=} \int_{\mathbb{C}^{d_y \times \bar{j}}} \mathrm{P}\left\{(\mathbf{y}, \mu, \mathbf{A}_{\sim k}, -\mathbf{a}_k) \in \underline{\Phi}_{\bar{j}}^{-1}(\mathcal{M})|\mathsf{x}_k = x_k, \mathbf{A}=A\right\} p_{\mathbf{A}}(A)\mathrm{d}A \\
&\overset{(5.121)}{=} \int_{\mathbb{C}^{d_y \times \bar{j}}} \mathrm{P}\left\{(\mathbf{A}_{\sim k}\mathbf{x}_{\sim k} + \mathbf{a}_k\mathsf{x}_k + \mathbf{w}, \mu, \mathbf{A}_{\sim k}, -\mathbf{a}_k) \in \underline{\Phi}_{\bar{j}}^{-1}(\mathcal{M})|\mathsf{x}_k = x_k, \mathbf{A}=A\right\} p_{\mathbf{A}}(A)\mathrm{d}A \\
&\overset{(5.12)}{=} \int_{\mathbb{C}^{d_y \times \bar{j}}} \mathrm{P}\left\{(\mathbf{A}_{\sim k}\mathbf{x}_{\sim k} + \mathbf{a}_k\mathsf{x}_k + \mathbf{w}, \mu, \mathbf{A}_{\sim k}, \mathbf{a}_k) \in \underline{\Phi}_{\bar{j}}^{-1}(\mathcal{M})|\mathsf{x}_k = -x_k, \mathbf{A}=A^{\neg k}\right\} p_{\mathbf{A}}(A^{\neg k})\mathrm{d}A \\
&= \int_{\mathbb{C}^{d_y \times \bar{j}}} \mathrm{P}\left\{\overline{\mu}_k \in \mathcal{M}|\mathsf{x}_k = -x_k, \mathbf{A}=A^{\neg k}\right\} p_{\mathbf{A}}(A^{\neg k})\mathrm{d}A \\
&= \mathrm{P}\{\overline{\mu}_k \in \mathcal{M}|\mathsf{x}_k = -x_k\},
\end{aligned}$$

which proves (5.124). Lastly, we show (5.125). Let $A_{i\pi k}$ be the matrix A with columns i and k

exchanged. Then

$$\begin{aligned}
&\mathrm{P}\left\{\overline{\mu}_k \in \mathcal{M} | \mathsf{x}_k = x\right\} \\
&\overset{(5.123)}{=} \mathrm{P}\left\{\overline{\mu}_k \in \mathcal{M} | \mathsf{x}_k = x, \mathsf{x}_i = x'\right\} \\
&= \int_{\mathbb{C}^{d_y \times \bar{j}}} \mathrm{P}\left\{(\mathbf{A}_{\sim ik}\mathbf{x}_{\sim ik} + \mathsf{x}_i \mathbf{a}_i + \mathsf{x}_k \mathbf{a}_k + \mathbf{w}, \mu, \mathbf{A}_{\sim k}, \mathbf{a}_k) \in \underline{\Phi}_{\bar{j}}^{-1}(\mathcal{M}) | \mathsf{x}_k = x, \mathsf{x}_i = x', \mathbf{A} = A\right\} p_{\mathbf{A}}(A)\mathrm{d}A \\
&= \int_{\mathbb{C}^{d_y \times \bar{j}}} \mathrm{P}\left\{(\mathbf{A}_{\sim ik}\mathbf{x}_{\sim ik} + \mathsf{x}_i \mathbf{a}_k + \mathsf{x}_k \mathbf{a}_i + \mathbf{w}, \mu, \mathbf{A}_{\sim i}, \mathbf{a}_i) \in \underline{\Phi}_{\bar{j}}^{-1}(\mathcal{M}) | \mathsf{x}_k = x, \mathsf{x}_i = x', \mathbf{A} = A_{i\pi k}\right\} p_{\mathbf{A}}(A)\mathrm{d}A \\
&\overset{(5.13)}{=} \int_{\mathbb{C}^{d_y \times \bar{j}}} \mathrm{P}\left\{(\mathbf{y}, \mu, \mathbf{A}_{\sim i}, \mathbf{a}_i) \in \underline{\Phi}_{\bar{j}}^{-1}(\mathcal{M}) | \mathsf{x}_i = x, \mathsf{x}_k = x', \mathbf{A} = A_{i\pi k}\right\} p_{\mathbf{A}}(A_{i\pi k})\mathrm{d}A \\
&= \mathrm{P}\left\{\overline{\mu}_i \in \mathcal{M} | \mathsf{x}_i = x, \mathsf{x}_k = x'\right\} \\
&= \mathrm{P}\left\{\overline{\mu}_i \in \mathcal{M} | \mathsf{x}_i = x\right\}. \qquad \blacksquare
\end{aligned}$$

Note that (5.125) is a consequence of the permutation invariance (5.13). If we did not assume (5.13) we would need to average over the edge index to obtain the conditional distribution of the SN output messages for a randomly chosen edge,

$$p_{\underline{\mu}|\mathsf{x}}(\underline{\mu}|x) = \frac{1}{\bar{j}} \sum_{k=1}^{\bar{j}} p_{\underline{\mu}_k|\mathsf{x}}(\underline{\mu}|x). \tag{5.126}$$

Based on Lemma 10, we can now derive fundamental symmetry and independence properties for the densities of messages passed at iteration ℓ on a code where the decoding neighborhood of depth $2\ell - 1$ of any VN is tree-like.

Theorem 13 (Conditional independence and symmetry of message densities).
Let $G \in C^N(d_s, \boldsymbol{d}_v, d_c)$ be a tripartite graph representing a SMLDPC code where the decoding neighborhood of depth $2\ell - 1$ of any VN $\mathsf{v} \in G$ is tree-like. For a given symmetric message passing algorithm (cf. Assumption 3) and symmetric edge coefficient distribution (cf. Assumption 2) let

$$\begin{aligned}
p^{(\ell)}_{\underline{\mu}|x} &: \quad \underline{\mathcal{M}}_\ell^{N d_{vs}} \rightarrow \mathbb{R}^+, \\
p^{(\ell)}_{\mu|x} &: \quad \mathcal{M}_\ell^{N(d_{vs}+d_{vc})} \rightarrow \mathbb{R}^+, \\
p^{(\ell)}_{\overline{\mu}|x} &: \quad \overline{\mathcal{M}}_\ell^{N d_{vc}} \rightarrow \mathbb{R}^+,
\end{aligned}$$

denote the joint VN- SN and CN message densities at iteration ℓ, respectively, assuming a particular codeword $\mathbf{x}$ has been transmitted. Then the following statements hold.

- *The messages are conditionally iid, i.e.,*

$$p^{(\ell)}_{\underline{\boldsymbol{\mu}}|\boldsymbol{x}}(\underline{\boldsymbol{\mu}}|\boldsymbol{x}) = \prod_{\mathrm{e}_s \in \mathcal{E}_s} p^{(\ell)}_{\underline{\mu}|\mathrm{x}}(\underline{\mu}_{\mathrm{e}_c}|x_{n(\mathrm{e}_c)}), \tag{5.127}$$

$$p^{(\ell)}_{\boldsymbol{\mu}|\boldsymbol{x}}(\boldsymbol{\mu}|\boldsymbol{x}) = \prod_{\mathrm{e}_s \in \mathcal{E}_s} p^{(\ell)}_{\mu_s|\mathrm{x}}(\mu_{\mathrm{e}_s}|x_{n(\mathrm{e}_s)}) \prod_{\mathrm{e}_c \in \mathcal{E}_c} p^{(\ell)}_{\mu_c|\mathrm{x}}(\mu_{\mathrm{e}_c}|x_{n(\mathrm{e}_c)}), \tag{5.128}$$

$$p^{(\ell)}_{\overline{\boldsymbol{\mu}}|\boldsymbol{x}}(\overline{\boldsymbol{\mu}}|\boldsymbol{x}) = \prod_{\mathrm{e}_c \in \mathcal{E}_c} p^{(\ell)}_{\overline{\mu}|\mathrm{x}}(\overline{\mu}_{\mathrm{e}_c}|x_{n(\mathrm{e}_c)}), \tag{5.129}$$

where $\mathcal{E}_s$ and $\mathcal{E}_c$ denote the SN- and CN side edge sets of G *and for an edge* e, $n(\mathrm{e})$ *denotes the index of the VN attached to* e.

- *The individual message densities in* (5.127) *to* (5.129) *are symmetric,*

$$p^{(\ell)}_{\underline{\mu}|\mathrm{x}}(\underline{\mu}|x) = p^{(\ell)}_{\underline{\mu}|\mathrm{x}}(-\underline{\mu}|-x), \tag{5.130}$$

$$p^{(\ell)}_{\mu_s|\mathrm{x}}(\mu|x) = p^{(\ell)}_{\mu_s|\mathrm{x}}(-\mu|-x), \tag{5.131}$$

$$p^{(\ell)}_{\mu_c|\mathrm{x}}(\mu|x) = p^{(\ell)}_{\mu_c|\mathrm{x}}(-\mu|-x), \tag{5.132}$$

$$p^{(\ell)}_{\overline{\mu}|\mathrm{x}}(\overline{\mu}|x) = p^{(\ell)}_{\overline{\mu}|\mathrm{x}}(-\overline{\mu}|-x), \tag{5.133}$$

where $p_{\mu_s|\mathrm{x}}$ and $p_{\mu_c|\mathrm{x}}$ are the VN-to-SN and VN-to-CN message densities and $p_{\underline{\mu}|\mathrm{x}}$ $p_{\overline{\mu}|\mathrm{x}}$ are the SN-to-VN and CN-to-VN message densities, respectively.

Proof: We show the statement by induction. First note that for iteration 0, all VN and CN messages are zero, i.e., $p^{(0)}_{\overline{\mu}|\mathrm{x}}(\overline{\mu}|x) = p^{(0)}_{\mu_s|\mathrm{x}}(\mu|x) = p^{(0)}_{\mu_c|\mathrm{x}}(\mu|x) = \Delta_0$ which is symmetric. For the SN update, first note that (5.122) holds for the zero inputs $p^{(0)}_{\mu_s|\mathrm{x}}(\mu|x) = \Delta_0$ and thus Lemma 10 applies, i.e. $p^{(0)}_{\underline{\mu}|\mathrm{x}}(-\underline{\mu}|-x) = p^{(0)}_{\underline{\mu}|\mathrm{x}}(\underline{\mu}|x)$ is symmetric. The iid properties (5.127) to (5.129) follow due to the tree-like decoding neighborhood.

Now assume that (5.128), (5.131) and (5.132) hold for iteration $\ell' < \ell$. We show that this implies (5.127), (5.129), (5.130) and (5.133) for the same iteration ℓ' as well as (5.128), (5.131) and (5.132) for the next iteration $\ell' + 1$; i.e., the symmetry and independence is preserved throughout one decoding iteration, which proves the claim. We start with the SN update for iteration ℓ'. Once again, due to the symmetry of the VN-to-SN messages at iteration ℓ', Lemma 10 applies and hence, (5.127) and (5.130) hold for iteration ℓ'. For the CN updates, note that since the incoming VN-to-CN messages are symmetric, we can write the message on edge $\mathrm{e} \in \mathcal{E}_c$ as $\mu^{(\ell')}_{\mathrm{e}} = x_{n(\mathrm{e})}\mu^{(\ell')}_{\mathrm{e}|+1}$ with $\mu^{(\ell')}_{\mathrm{e}|+1} \sim p^{(\ell')}_{\mu|\mathrm{x}}(\mu|+1)$. Hence, due to the CN symmetry

(2.72), we have for the message on the CN-to-VN message on edge e_j

$$\overline{\mu}_{\mathrm{e}_j}^{(\ell')} = \overline{\Phi}_j^{(\ell')}\big(\mu_{\mathrm{e}_1}^{(\ell')},\ldots,\mu_{\mathrm{e}_{j-1}}^{(\ell')}\big) = \left(\prod_{i=1}^{j-1} x_{n(\mathrm{e}_i)}\right)\overline{\Phi}_j^{(\ell')}\big(\mu_{\mathrm{e}_1|+1}^{(\ell')},\ldots,\mu_{\mathrm{e}_{j-1}|+1}^{(\ell')}\big) \tag{5.134}$$

$$= x_{n(\mathrm{e}_j)}\Phi_j^{(\ell')}\big(\mu_{\mathrm{e}_1|+1}^{(\ell')},\ldots,\mu_{\mathrm{e}_{j-1}|+1}^{(\ell')}\big) \tag{5.135}$$

$$= x_{n(\mathrm{e}_j)}\overline{\mu}_{\mathrm{e}_j|+1}^{(\ell')}, \tag{5.136}$$

where (5.135) follows from $\prod_{i=1}^{j} x_{n(\mathrm{e}_i)} = +1$ which holds since $\{\mathrm{e}_i\}$ are the edges of a CN and $\boldsymbol{x}$ is a codeword. This shows (5.129) and (5.133) for iteration ℓ'.

Now for the VN updates, let $\{\mathrm{e}_{s,1},\ldots,\mathrm{e}_{s,k}\} \subset \mathcal{E}_s$ and $\{\mathrm{e}_{c,1},\ldots,\mathrm{e}_{c,i}\} \subset \mathcal{E}_c$ be the SN and VN side edges connected to some VN v_n. Since all input messages are symmetric, we can perform the same factorization as for the CN update, i.e., $\underline{\mu}_{\mathrm{e}_{s,k'}}^{(\ell')} = x_n \underline{\mu}_{\mathrm{e}_{s,k'}|+1}^{(\ell')}$ with $\underline{\mu}_{\mathrm{e}_{s,k'}|+1}^{(\ell')} \sim p_{\underline{\mu}|\mathsf{x}}^{(\ell')}(\underline{\mu}|+1)$ for $k' = 1,\ldots,k$ and $\overline{\mu}_{\mathrm{e}_{c,i'}}^{(\ell')} = x_n \overline{\mu}_{\mathrm{e}_{c,i'}|+1}^{(\ell')}$ with $\overline{\mu}_{\mathrm{e}_{c,i'}|+1}^{(\ell')} \sim p_{\overline{\mu}|\mathsf{x}}^{(\ell')}(\overline{\mu}| + 1)$ for $i' = 1,\ldots,i$. Thus, due to the VN symmetry (5.115),

$$\mu_{\mathrm{e}_{s,k}}^{(\ell'+1)} = \Phi_{k-1,i}^{(\ell')}\big(\underline{\mu}_{\mathrm{e}_{s,1}}^{(\ell')},\ldots,\underline{\mu}_{\mathrm{e}_{s,k-1}}^{(\ell')},\overline{\mu}_{\mathrm{e}_{c,1}}^{(\ell')},\ldots,\overline{\mu}_{\mathrm{e}_{c,i}}^{(\ell')}\big) \tag{5.137}$$

$$= x_n\Phi_{k-1,i}^{(\ell')}\big(\underline{\mu}_{\mathrm{e}_{s,1|+1}}^{(\ell')},\ldots,\underline{\mu}_{\mathrm{e}_{s,k-1|+1}}^{(\ell')},\overline{\mu}_{\mathrm{e}_{c,1|+1}}^{(\ell')},\ldots,\overline{\mu}_{\mathrm{e}_{c,i|+1}}^{(\ell')}\big) \tag{5.138}$$

$$= x_n\mu_{\mathrm{e}_{s,k|+1}}^{(\ell'+1)}, \tag{5.139}$$

$$\mu_{\mathrm{e}_{c,i}}^{(\ell'+1)} = \Phi_{k,i-1}^{(\ell')}\big(\underline{\mu}_{\mathrm{e}_{s,1}}^{(\ell')},\ldots,\underline{\mu}_{\mathrm{e}_{s,k-1}}^{(\ell')},\overline{\mu}_{\mathrm{e}_{c,1}}^{(\ell')},\ldots,\overline{\mu}_{\mathrm{e}_{c,i}}^{(\ell')}\big) \tag{5.140}$$

$$= x_n\Phi_{k,i-1}^{(\ell')}\big(\underline{\mu}_{\mathrm{e}_{s,1|+1}}^{(\ell')},\ldots,\underline{\mu}_{\mathrm{e}_{s,k|+1}}^{(\ell')},\overline{\mu}_{\mathrm{e}_{c,1|+1}}^{(\ell')},\ldots,\overline{\mu}_{\mathrm{e}_{c,i-1|+1}}^{(\ell')}\big) \tag{5.141}$$

$$= x_n\mu_{\mathrm{e}_{c,i|+1}}^{(\ell'+1)}, \tag{5.142}$$

which proves (5.128), (5.131) and (5.132) for iteration $\ell' + 1$ and thus completes the induction proof. ■

Note that Theorem 13 implies the following:

Corollary 1 (Conditional independence of error probability under symmetry)**.** *Let $P_e^{(\ell)}$ be defined as in Theorem 11 and let $P_e^{(\ell)}(\boldsymbol{x})$ be the corresponding conditional probability assuming codeword $\boldsymbol{x}$ was transmitted, i.e. $P_e^{(\ell)} = \sum_{\boldsymbol{x}\in\mathcal{C}} P_e^{(\ell)}(\boldsymbol{x})p(\boldsymbol{x})$. Under the assumptions of Theorem 13, $P_e^{(\ell)}(\boldsymbol{x})$ is independent of $\boldsymbol{x}$ and hence $P_e^{(\ell)} = P_e^{(\ell)}(\boldsymbol{x})$ for any $\boldsymbol{x} \in \mathcal{C}$.*

Thus, to simplify the calculation of $P_e^{(\ell)}$, we can assume that $\boldsymbol{x} = +\mathbf{1}$ was transmitted, similarly to the case of LDPC codes (cf. Lemma 1). Hence, from now on we write e.g., $p_{\mu_s|\mathsf{x}}^{(\ell)}(\mu) = p_{\mu_s|\mathsf{x}}^{(\ell)}(\mu| + 1)$, implicitly assuming that $\boldsymbol{x} = +\mathbf{1}$ was transmitted. We follow the same convention for the other densities (5.130), (5.132) and (5.133).

5.5.2 LLR-Consistency of Message Densities

Lemma 11. *Consider the BP SN update* (5.43),

$$\underline{\mathsf{\mu}}_k = \underline{\Phi}(\mathbf{y}, \boldsymbol{\mu}, \mathbf{A}_{\sim k}, \mathbf{a}_k) = \log \frac{p(\mathbf{y}|x_k = +1, \mathbf{A}, \boldsymbol{\mu})}{p(\mathbf{y}|x_k = -1, \mathbf{A}, \boldsymbol{\mu})} \tag{5.143}$$

under the assumption of symmetric input messages (5.122) *and symmetric edge coefficients (cf. Assumption 2) with SN constraint* $\mathbf{y} = \mathbf{A}x + \mathbf{w}$. *Then the output density* $p_{\underline{\mathsf{\mu}}_k|\mathsf{x}_k}(\underline{\mu}) = p_{\underline{\mathsf{\mu}}_k|\mathsf{x}_k}(\underline{\mu}| + 1)$ *is LLR-consistent according to* (2.85) *and* (2.86).

Proof: We show that

$$\int_{\mathbb{R}} h(\underline{\mu}) p_{\underline{\mathsf{\mu}}_k|\mathsf{x}_k}(\underline{\mu}) \mathrm{d}\underline{\mu} = \int_{\mathbb{R}} e^{-\underline{\mu}} h(-\underline{\mu}) p_{\underline{\mathsf{\mu}}_k|\mathsf{x}_k}(\underline{\mu}) \mathrm{d}\underline{\mu} \tag{5.144}$$

for any function h such that the above integral exists, which implies (2.85), cf. [77]. Note that the BP SN update is symmetric according to (5.117) and (5.118) and hence, Lemma 10 applies. Thus

$$\begin{aligned}
\mathbb{E}_{\underline{\mathsf{\mu}}_k}\left[e^{-\underline{\mathsf{\mu}}_k} h(-\underline{\mathsf{\mu}}_k) \,|\, \mathsf{x}_k = +1\right] &= \mathbb{E}_{\underline{\mathsf{\mu}}_k}\left[e^{\underline{\mathsf{\mu}}_k} h(\underline{\mathsf{\mu}}_k) \,|\, \mathsf{x}_k = -1\right] \\
&= \mathbb{E}_{\boldsymbol{\mu},\mathbf{A}}\left[\mathbb{E}_{\mathbf{y}}\left[\frac{p(\mathbf{y}|\mathbf{A}, \boldsymbol{\mu}, \mathsf{x}_k = +1)}{p(\mathbf{y}|\mathbf{A}, \boldsymbol{\mu}, \mathsf{x}_k = -1)} h\big(\underline{\Phi}_{\bar{j}}(\mathbf{y}, \boldsymbol{\mu}, \mathbf{A})\big) \,\middle|\, \mathbf{A}, \boldsymbol{\mu}, \mathsf{x}_k = -1\right]\right] \\
&= \mathbb{E}_{\boldsymbol{\mu},\mathbf{A}}\left[\mathbb{E}_{\mathbf{y}}\left[h\big(\underline{\Phi}_{\bar{j}}(\mathbf{y}, \boldsymbol{\mu}, \mathbf{A})\big) \,\middle|\, \mathbf{A}, \boldsymbol{\mu}, \mathsf{x}_k = +1\right]\right] \\
&= \mathbb{E}_{\underline{\mathsf{\mu}}_k}\left[h(\underline{\mathsf{\mu}}_k) \,|\, \mathsf{x}_k = +1\right] \qquad \blacksquare
\end{aligned}$$

Theorem 14 (LLR-consistency of message densities under BP decoding)**.** *For BP decoding and a symmetric edge coefficient distribution (cf. Assumption 2), all message densities* (5.130) *to* (5.133) *as well as the APP LLR density* $p^{(\ell)}_{\mathsf{L}|\mathsf{x}}$ *are LLR-consistent and symmetric.*

Proof: The proposition follows by induction from Lemma 11 and Theorem 13 as well as the fact that for BP, both the VN updates (5.41) and the CN updates (2.61) preserve LLR-consistency, cf. [77]. ■

5.5.3 Threshold Calculation for SMLDPC Codes

In Section 2.2.3, we have seen that for BP, the VN and CN updates (2.59) and (2.61), the densities transform according to (2.78) and (2.82). Generally, speaking, let

$$^{*}\underline{\Phi}_{\bar{j}}^{(\ell)}\big(p_{\mathbf{w}|d_y}, p_{\mathbf{A}|d_y,\bar{j}}, p_{\mu_s|\mathsf{x}}^{(\ell)}\big) = p_{\underline{\mu}|\mathsf{x},d_y,\bar{j}}^{(\ell)}, \tag{5.145}$$

$$^{*}\overline{\Phi}_{j}^{(\ell)}\big(p_{\mu_c|\mathsf{x}}^{(\ell)}\big) = p_{\overline{\mu}|\mathsf{x},j}^{(\ell)}, \tag{5.146}$$

$$^{*}\Phi_{k-1,i}^{(\ell)}\big(p_{\underline{\mu}|\mathsf{x}}^{(\ell)}, p_{\overline{\mu}|\mathsf{x}}^{(\ell)}\big) = p_{\mu_s|\mathsf{x},k,i}^{(\ell+1)}, \tag{5.147}$$

$$^{*}\Phi_{k,i-1}^{(\ell)}\big(p_{\underline{\mu}|\mathsf{x}}^{(\ell)}, p_{\overline{\mu}|\mathsf{x}}^{(\ell)}\big) = p_{\mu_c|\mathsf{x},k,i}^{(\ell+1)}, \tag{5.148}$$

denote the transformation of densities corresponding to the generic, symmetric (cf. Assumption 3) update rules (5.35), (5.36), (5.38) and (5.39), where we included the respective node degrees and channel dimension in the conditioning of the densities. Due to the similarities with the BP updates of LDPC codes, we have

$$^{*}\overline{\Phi}_{j}^{\mathrm{BP}}\big(p_{\mu_c|\mathsf{x}}^{(\ell)}\big) = \Gamma^{-1}\Big(\Gamma\Big(p_{\mu_c|\mathsf{x}}^{(\ell)}\Big)^{\otimes j-1}\Big), \tag{5.149}$$

$$^{*}\Phi_{k,i}^{\mathrm{BP}}\big(p_{\underline{\mu}|\mathsf{x}}^{(\ell)}, p_{\overline{\mu}|\mathsf{x}}^{(\ell)}\big) = p_{\underline{\mu}|\mathsf{x}}^{\otimes k} \otimes p_{\overline{\mu}|\mathsf{x}}^{\otimes i}, \tag{5.150}$$

cf. (2.78) and (2.82). Unfortunately, the SN update $\underline{\Phi}_{\bar{j}}^{\mathrm{BP}}$ (5.43) appears to be too complex to obtain a corresponding $^{*}\underline{\Phi}_{\bar{j}}^{\mathrm{BP}}$. Thus, without $^{*}\underline{\Phi}_{\bar{j}}^{\mathrm{BP}}$, an explicit recursion of densities as has been obtained for LDPC codes in (2.83) is not possible. However, by the following theorem, it is not necessary to explicitly calculate $P_e^{(\ell)}$ to obtain the threshold (5.60).

Theorem 15. *For LLR-consistent densities, $P_e^{(\ell)}$ converges to zero iff $I(\mathsf{L}^{(\ell)}, \mathsf{x})$ converges to one.*

Proof: According to [77, Corollary 1], LLR-consistency implies $P_e^{(\ell)} \to 0$ iff $p_{\mathsf{L}|\mathsf{x}}^{(\ell)} \to \Delta_\infty$. Furthermore, it is shown in [102] that for a density with these properties, the mutual information $I(\mathsf{L};\mathsf{x})$ can be computed as

$$I(\mathsf{L};\mathsf{x}) = 1 - \int_{\mathbb{R}} p_{\mathsf{L}|\mathsf{x}}(\mathrm{L}) \log_2\big(1 + e^{-\mathrm{L}}\big) \mathrm{d}\mathrm{L}, \tag{5.151}$$

where $I(\mathsf{L};\mathsf{x}) \le 1$, with equality iff $p_{\mathsf{L}|\mathsf{x}} = \Delta_\infty$, which proves the claim. ■

Note that Theorem 15 covers the important special case of BP, which is the optimal decoding algorithm in the MAP sense. Consequently we can rewrite (5.60) as

$$\sigma^{*} = \sup\{\sigma : \lim_{\ell\to\infty} P_e^{(\ell)}(\sigma) = 0\} = \sup\{\sigma : \lim_{\ell\to\infty} I^{(\ell)}(\sigma) = 1\}, \tag{5.152}$$

with $I^{(\ell)} = I(\mathsf{L}^{(\ell)};\mathsf{x})$. Thus, under the assumption that we can calculate the mutual information exactly, the threshold determination can be based on EXIT analysis rather than density evolution.

Next, we show that the results on symmetry and LLR-consistency we derived in Sections 5.5.1 and 5.5.2 can be exploited to establish a very accurate characterization of the SN update in terms of its EXIT function, cf. Section 2.3. The idea behind that is as follows:

- The input messages to the SN updates are the outputs of the VNs, i.e., the sum of independent random variables. Thus for increasing VN degrees, the input messages to the SN updates become Gaussian by the central limit theorem. Furthermore, simulations show that the densities of the VN messages match a Gaussian distribution fairly closely, cf. [19]. Thus, it is reasonable to assume that the input messages to the SN update are Gaussian and due to the LLR-consistency, are distributed according to

$$\mathrm{\mu}|\mathrm{x}=+1 \sim p_{\mathrm{\mu}|\mathrm{x}}=\mathcal{N}(\sigma_A^2/2,\sigma_A^2), \tag{5.153}$$

  where σ_A is related to the a priori information according to $I_A = I(\mathrm{\mu};\mathrm{x}) = J(\sigma_A)$, cf. (2.87).

- The SN output density $p_{\underline{\mathrm{\mu}}|\mathrm{x}}$ is symmetric and LLR-consistent. Hence, by (5.151), we can calculate

$$\begin{aligned}
I(\underline{\mathrm{\mu}};\mathrm{x}) &= 1-\mathbb{E}_{\underline{\mathrm{\mu}}}\left[\log_2\left(1+e^{-\underline{\mathrm{\mu}}}\right)\,\middle|\,\mathrm{x}=1\right]\\
&= 1-\mathbb{E}_{\mathbf{w},\boldsymbol{\mu},\mathbf{A}}\left[\log_2\left(1+e^{-\underline{\Phi}_{\bar{j}}(\mathbf{A}\mathbf{x}+\mathbf{w},\boldsymbol{\mu},\mathbf{A})}\right)\,\middle|\,\mathbf{x}=\mathbf{1}\right] &(5.154)\\
&\approx 1-\frac{1}{L_s}\sum_{l=1}^{L_s}\log_2\left(1+e^{-\underline{\Phi}_{\bar{j}}(\mathbf{A}_l\mathbf{1}+\mathbf{w}_l,\boldsymbol{\mu}_{l|1},\mathbf{A}_l)}\right), &(5.155)
\end{aligned}$$

  where (5.155) becomes exact for an infinite amount $L_s \to \infty$ of iid samples $(\mathbf{w}_l,\boldsymbol{\mu}_{l|1},\mathbf{A}_l)$ distributed according to

$$(\mathbf{w}_l,\boldsymbol{\mu}_{l|1},\mathbf{A}_l) \sim p_{\mathbf{w}}(w)p_{\mathbf{A}}(A)\prod_{i=1}^{\bar{j}-1}p_{\mathrm{\mu}|\mathrm{x}}(\mu_i). \tag{5.156}$$

- Using (5.155) and (5.156), we can approximate the EXIT function of the SN update arbitrary closely, and establish a functional relationship in the form of

$$I_E = I_E(I_A, N_0, p_{\mathbf{A}}, \underline{\Phi}). \tag{5.157}$$

  Here, the EXIT curve is parametrized by the channel noise variance N_0, the edge coefficient distribution $p_{\mathbf{A}}$ as well as the update rule $\underline{\Phi}$, which is required to preserve LLR-consistency for the estimation (5.155) to hold.

5.6 Irregular Graphs

5.6.1 General Definition and Code Rate

In the following, let S_{ijk} denote the number of SNs with degree j, channel dimension i, and type k. In this context, the introduction of a SN type allows for mixtures of SNs that share the same dimension and degree but use different modulation schemes and/or update rules. Similarly, let V_{ij} denote the number of VNs with VN-to-SN degree i and VN-to-CN degree j and C_j the number of CNs of degree j. Hence,

$$L = \sum_i \sum_j \sum_k S_{ijk} \qquad N = \sum_i \sum_j V_{ij}, \qquad M = \sum_j C_j. \tag{5.158}$$

We define the *node perspective degree distributions* as

$$\Sigma_{ijk} \triangleq \frac{S_{ijk}}{L}, \qquad \Lambda_{ij} \triangleq \frac{V_{ij}}{N}, \qquad P_j \triangleq \frac{C_j}{M}, \tag{5.159}$$

cf. Table 5.1 for an overview of related quantities.

Due to the constraints on the number of SN side and CN side edges E_s and E_c,

$$E_s = \sum_j j \sum_i \sum_k S_{ijk} = \sum_i i \sum_j V_{ij}, \qquad E_c = \sum_j j C_j = \sum_j j \sum_i V_{ij} \iff \tag{5.160}$$

$$L \sum_j j \sum_i \sum_k \Sigma_{ijk} = N \sum_i i \sum_j \Lambda_{ij}, \qquad M \sum_j i P_j = N \sum_j j \sum_i \Lambda_{ij} \tag{5.161}$$

we can express the code rate solely in terms of the degree distributions,

$$\begin{aligned} R = \frac{N-M}{\bar{d}_y L} = \left(1 - \frac{M}{N}\right) \frac{N}{\bar{d}_y L} &= \left(1 - \frac{\sum_j j \sum_i \Lambda_{ij}}{\sum_i i P_i}\right) \frac{\sum_j j \sum_i \sum_k \Sigma_{ijk}}{\left(\sum_i i \sum_j \Lambda_{ij}\right)\left(\sum_j \sum_i i \sum_k \Sigma_{ijk}\right)} \\ &= \frac{\left[1 - \left(\sum_j \frac{\rho_j}{j}\right)\left(\sum_j j \sum_i \Lambda_{ij}\right)\right]\left(\sum_i \sum_j j \sum_k \Sigma_{ijk}\right)}{\left(\sum_i i \sum_j \Lambda_{ij}\right)\left(\sum_i i \sum_j \sum_k \Sigma_{ijk}\right)} \qquad (5.162) \\ &= \frac{\left[1 - \left(\sum_j \frac{\rho_j}{j}\right)\left(\sum_j j \Lambda^s_j\right)\right]\left(\sum_j j \Sigma^s_j\right)}{\left(\sum_i i \Lambda^c_i\right)\left(\sum_i i \Sigma^d_i\right)}. \qquad (5.163) \end{aligned}$$

It is interesting to note that R only depends on the marginal distributions Σ^c_i, Σ^s_j, Λ^s_i and Λ^c_j (cf. Table 5.1), whereas the actual convergence behaviour of the iterative message passing decoder is determined by the joint distributions Σ_{ijk} and Λ_{ij}. That is, the mixture densities of an

Quantity	Symbol	in terms of base quantity
number of SNs with channel dimension i, SN degree j and type k	S_{ijk}	$L\Sigma_{ijk}$
fraction of SNs with channel dimension i, SN degree j and type k	Σ_{ijk}	Σ_{ijk}
fraction of SNs with channel dimension i	Σ_i^c	$\displaystyle\sum_j\sum_k \Sigma_{ijk}$
fraction of SNs with degree j	Σ_j^s	$\displaystyle\sum_i\sum_k \Sigma_{ijk}$
average channel dimension	$\bar{d}_y$	$\displaystyle\sum_j\sum_i i\sum_k \Sigma_{ijk} = \sum_i i\Sigma_i^c$
fraction of SN side edges connected to a SN of dimension i, degree j and type k	σ_{ijk}	$\dfrac{j\Sigma_{ijk}}{\sum_j j\sum_i\sum_k \Sigma_{ijk}}$
fraction of SN side edges connected to a SN of degree j	σ_j	$\dfrac{j\sum_i\sum_k \Sigma_{ijk}}{\sum_j j\sum_i\sum_k \Sigma_{ijk}} = \displaystyle\sum_i\sum_k \sigma_{ijk}$
number of VNs with VN-to-SN degree i and VN-to-CN degree degree j	V_{ij}	$N\Lambda_{ij}$
fraction of VNs with VN-to-SN degree i and VN-to-CN degree degree j	Λ_{ij}	Λ_{ij}
fraction of VNs with VN-to-SN degree i	Λ_i^s	$\displaystyle\sum_j \Lambda_{ij}$
fraction of VNs with VN-to-CN degree j	Λ_j^c	$\displaystyle\sum_i \Lambda_{ij}$
fraction of SN side edges connected to VN with VN-to-SN degree i and VN-to-SN degree j	λ_{ij}^s	$\dfrac{i\Lambda_{ij}}{\sum_i i\sum_j \Lambda_{ij}} = \dfrac{i\Lambda_{ij}}{\sum_i i\Lambda_i^s}$
fraction of CN side edges connected to VN with VN-to-SN degree i and VN-to-CN degree j	λ_{ij}^c	$\dfrac{j\Lambda_{ij}}{\sum_j j\sum_i \Lambda_{ij}} = \dfrac{j\Lambda_{ij}}{\sum_j j\Lambda_j^c}$
number of CNs with degree j	C_j	MP_j
fraction of CNs with degree j	P_j	P_j
fraction of CN side edges connected to CN with degree j	ρ_j	$\dfrac{jP_j}{\sum_j jP_j}$

Table 5.1: Quantities specifying the connectivity of an irregular SMLDPC code.

irregular code read

$$p^{(\ell)}_{\underline{\mu}|x} = \sum_i \sum_j \sigma_{ijk} p^{(\ell)}_{\underline{\mu}|x,i,j,k}, \tag{5.164}$$

$$p^{(\ell)}_{\overline{\mu}|x} = \sum_j \rho_j p^{(\ell)}_{\overline{\mu}|x,j}, \tag{5.165}$$

$$p^{(\ell+1)}_{\mu_s|x} = \sum_i \sum_j \lambda^s_{ij} p^{(\ell+1)}_{\mu_s|x,i,j}, \tag{5.166}$$

$$p^{(\ell+1)}_{\mu_c|x} = \sum_i \sum_j \lambda^c_{ij} p^{(\ell+1)}_{\mu_c|x,i,j}, \tag{5.167}$$

$$p^{(\ell)}_{L|x} = \sum_i \sum_j \Lambda_{ij} p^{(\ell)}_{L|x,i,j}, \tag{5.168}$$

where the densities for a particular combination of degrees and/or channel dimension are determined by the transformations (5.145) to (5.148), e.g., for BP,

$$p^{(\ell+1)}_{\mu_s|x,i,j} = {}^*\Phi^{\mathrm{BP}}_{i-1,j}\big(p^{(\ell)}_{\underline{\mu}|x}, p^{(\ell)}_{\overline{\mu}|x}\big) = p^{\otimes i-1}_{\underline{\mu}|x} \otimes p^{\otimes j}_{\overline{\mu}|x}.$$

While the mixture densities (5.164) to (5.167) are characterized by the edge perspective degree distributions, the code rate (5.163) can only be accurately expressed via the node perspective distributions, assuming that there are VNs with either VN-to-SN or VN-to-SN degree zero. Furthermore, note the rate is independent of the type distribution.

Thus, in general, the irregular structure of a graph is best described by the degree distribution triple (Σ, Λ, ρ). We could theoretically introduce polynomials (over a two dimensional domain) to express the SN and VN degree distributions as it is commonly done for LDPC codes. For SMLDPC codes however, we lack an explicit recursion based on the polynomial notation (cf. (2.83)), so the notational usefulness would be limited and we directly work with degree distributions.

The above considerations illustrate very well the difference between our analysis of SMLDPC codes which is based on joint distributions Λ and Σ and the approach taken in [114, 115], which merely has access to the marginals Λ^s and Λ^c.

5.6.2 Generalization of Results to Irregular Codes

In this section, we will briefly discuss how the results we obtained previously for regular codes generalize to the irregular case.

Channel Model: In case of multiple different channel dimensions i, SN degrees j and node types k, we have to specify one edge coefficient distribution per triple (i, j, k). That is, we generalize the distribution of edge coefficient matrices to $p_{\mathsf{A}|i,j,k}$, where for any (i, j, k), $p_{\mathsf{A}|i,j,k}$ fulfills Assumption 2. Let C_{ijk} and P_{ijk} be the capacity and channel input power for a particular tuple (i, j, k) according to (5.18) and (5.19) with $i = d_y$ and

$j = d_s$ and corresponding coefficient type. Then the capacity and input power of the irregular mixture are given by

$$C = \sum_i \sum_j \sum_k \Sigma_{ijk} C_{ijk}, \tag{5.169}$$

$$P = \sum_i \sum_j \sum_k \Sigma_{ijk} P_{ijk}. \tag{5.170}$$

Note that $P_{ijk} = 1$ implies $P = 1$.

Decoding: The general message passing decoding procedure described in Section 5.3 applies unchanged to the irregular case, where the update rules depend on the degree of the nodes. For SNs of different dimension, alternative decoding schedules could be of interest. The reasoning here is that the updates for high-dimensional SNs are more costly and take longer to evaluate.

Concentration and Symmetry: All results derived in Section 5.4 for regular codes also apply to irregular codes. In particular, Theorem 11 holds with adjusted constants β and γ. Furthermore, the symmetry relations obtained in Section 5.5 also apply to irregular codes, assuming that for each degree, the nodes adhere to the symmetry constraints in Assumption 3.

5.6.3 Discussion and Potential of Irregular Codes

One might ask what the benefit is of introducing irregular SMLDPC codes. For the VNs, the motivation for irregularity is similar as for LDPC codes: While low-degree VNs only receive little extrinsic information and are therefore more difficult to decode, they do increase the overall code rate. Conversely, high-degree VNs provide reliability due to redundancy but reduce the code rate. For irregular LDPC codes, it turned out that mixing VNs of different degrees can greatly improve the performance of a code for a given rate. That is, while the addition of low-degree VNs increases the rate, the high-degree VNs provide a lot of extrinsic information, which through the iterative message passing decoding procedure, benefits also the low-degree VN, thereby achieving good performance at high rates close to the channel capacity.

Another aspect that calls for VN irregularity is the two dimensional nature of the VN degree for SMLDPC codes. A priori, it is unclear which combinations of VN-to-SN and VN-to-CN degrees lead to good performance and it appears intuitive that a single degree pair (d_{vs}, d_{vc}) does not fully exploit the potential for iteratively decoded SMLDPC codes.

For the SNs the motivation is similar. By taking into account irregular SN degrees, we can model a mixture of different modulation types, e.g. different modulation orders. For example, let $i = 1$ and $j \in \{2, 4\}$ for the coefficient model (5.21), which leads to a mix of 4-QAM and 16-QAM constellations. By introducing multiple types, we can even mix different coefficient models of the same degree. We could e.g., mix real SM-EPA and PAM of the same symbol node degree. A reason for doing this might be that both modulations have desirable properties

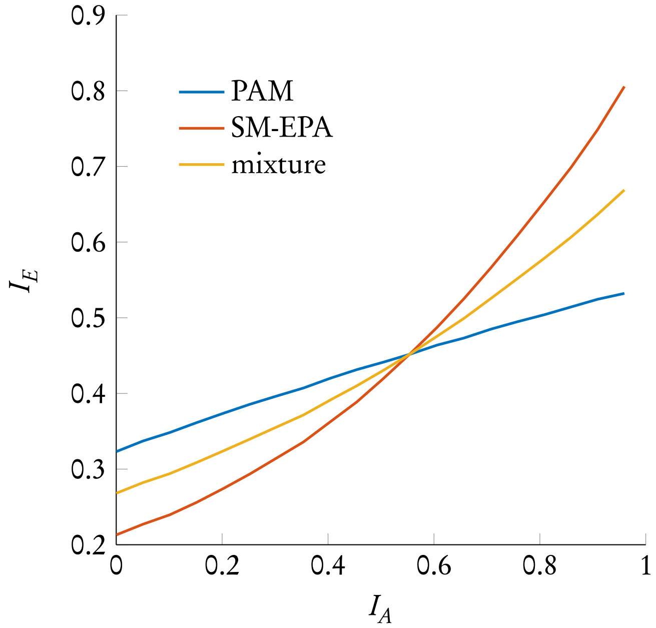


Figure 5.9: EXIT function for PAM, SM-EPA and a 50 — 50 mix thereof for a SN degree of 4 and $N_0 = 0.6$

and a mixture of those could potentially enhance the iterative decoding procedure. Whereas (Gray coded) PAM provides high extrinsic information without any a priori information, it does not benefit much from iterative demodulation due to its flat EXIT function. Conversely, the EXIT function of SM-EPA is rather steep, meaning that this modulation is not optimal at the beginning of iterative decoding due to lack of a priori information but produces messages much more reliable than PAM later in the decoding process. Note that due to the symmetry and consistency of SN messages (cf. Sections 5.5.1 and 5.5.2) the mutual information is linear in the densities, cf. (5.151). Hence, it follows from (5.164), that the EXIT function of SN mixtures is the linear combination of the individual EXIT functions,

$$I_E(I_A) = \sum_i \sum_j \sum_k \sigma_{ijk} I_{E|i,j,k}(I_A). \tag{5.171}$$

Figure 5.9 illustrates the above. Using a mixture of 50% SM-EPA nodes and 50% PAM nodes, we can see that this leads to a combination of the characteristics of both EXIT functions. Consequently, for irregular SMLDPC codes, we have very granular control over the average EXIT functions of the SN updates, which helps to prevent rate loss and therefore approaching capacity [3].

By that logic, we can also define different SN types based on using different update rules[6] (5.35). Generally speaking, the parameter space of irregular SMLDPC codes is vast and the fact that special cases of this framework such as SN-regular codes [114] or conventional modulated

[6] We note that unlike edge coefficients and graph structure, the update rules are *not* part of the code itself. However, from a system design perspective, it makes sense to include them in the SN type.

LDPC codes without iterative demodulation [48, 98, 121] were shown to come close to the channel capacity, lead us to believe that SMLDPC codes are capable of achieving the Shannon capacity in the bandwidth limited regime. In particular, for a prescribed code rate R, we strive to find combinations of degree distributions, coefficient distributions as well as SN update rules that allow decoding close to the capacity in linear time, i.e.,

$$\sigma^* = \sup\left\{\sigma_\mathrm{w} \mid \lim_{\ell\to\infty} I^{(\ell)}(\sigma_\mathrm{w}, p_{\mathbf{A}|i,j,k}, \Sigma, \Lambda, \rho) \to 1\right\}, \tag{5.172}$$

such that $|\sigma_{\max,R} - \sigma^*|$ is small. Here, $I^{(\ell)}$ denotes the mutual information between APP LLRs and code bits after iteration ℓ and $\sigma_{\max,R} = 1/\sqrt{2^R - 1}$ is the maximum noise level of an AWGN channel for which reliable transmission of rate R is possible according to the channel capacity, assuming normalized transmit power $P = 1$. Having found such a tuple $\left(p_{\mathbf{A}|i,j,k}, \Sigma, \Lambda, \rho\right)$, the concentration theorem (Theorem 11) then ensures that for large N, a code randomly generated according to $\left(p_{\mathbf{A}|i,j,k}, \Sigma, \Lambda, \rho\right)$ will have performance close to the ensemble average and therefore perform close to capacity.

5.7 EXIT Evolution and Degree Distribution Optimization

5.7.1 EXIT based Threshold Computation

In what follows, we will describe the decoding process (cf. Section 5.3.1) of an SMLDPC code in terms of EXIT analysis (cf. Section 2.3). We will refer to this procedure as *EXIT evolution*. Let $\{\underline{\Phi}_{i,j,k}\}_{(i,j,k)\in\mathcal{D}_\mathrm{s}}$ be a set of SN update rules of the form (5.35), where the tuples (i,j,k) consists of channel dimensions, SN degrees and SN types and $\mathcal{D}_\mathrm{s}$ is the set of such combinations. Furthermore, for the moment, we assume that any $\underline{\Phi}_{i,j,k}$ is independent of the iteration index, however, the procedure described below also works for that case with some slight adjustments, as we will explain later. Furthermore, for the tuple (i,j,k), $p_{\mathbf{A}|i,j,k}$ denotes the corresponding edge coefficient distribution.

Input: Channel noise standard deviation σ_w, degree, dimension and type structure $\mathcal{D}_\mathrm{s}$, $\mathcal{D}_\mathrm{c}$, $\mathcal{D}_\mathrm{v}$ and corresponding distributions (Σ, Λ, ρ), coefficient distribution(s) $\{p_{\mathbf{A}|i,j,k}\}$, tolerance $\epsilon > 0$, update rule(s) $\{\underline{\Phi}_{i,j,k}\}$ that preserve symmetry and LLR-consistency.

Initialization: For a given σ_w and all tuples $(i,j,k) \in \mathcal{D}_\mathrm{s}$, calculate the EXIT function

$$I_{E|i,j,k}(I_A, \sigma_\mathrm{w}^2, p_{\mathbf{A}|i,j,k}, \underline{\Phi}_{i,j,k}), \tag{5.173}$$

using the procedure described in Section 5.5.3. Usually, it is sufficient to calculate I_E for something like 100 supporting values of I_A evenly spaced between 0 and 1 and interpolate in between those values. Furthermore, set $\ell = 0$ and $I^{(0)}_{\mathrm{v}\to\mathrm{s}} = I^{(0)}_{\mathrm{v}\to\mathrm{c}} = I^{(0)}_{\mathrm{c}\to\mathrm{v}} = 0$.

Iteration:

1. SN and CN EXIT updates:

   (a) SN update: Calculate the SN EXIT update according to the pre-generated EXIT functions (5.173),

   $$I^{(\ell)}_{\mathrm{s}\to\mathrm{v}|i,j,k} = I_E(I^{(\ell)}_{\mathrm{v}\to\mathrm{s}}, \sigma^2_{\mathrm{w}}, p_{\mathbf{A}|i,j,k}, \underline{\Phi}_{i,j,k}). \tag{5.174}$$

   Then, due to (5.151) and (5.164) and Theorem 14 and, we obtain

   $$I^{(\ell)}_{\mathrm{s}\to\mathrm{v}} = \sum_{(i,j,k)\in\mathcal{D}_{\mathrm{s}}} \sigma_{ijk} I^{(\ell)}_{\mathrm{s}\to\mathrm{v}|i,j,k}. \tag{5.175}$$

   The same argument applies to the averaging of CN and VN updates in the following.

   (b) CN update: Calculate the CN EXIT update according to the EXIT function (2.92),

   $$I^{(\ell)}_{\mathrm{c}\to\mathrm{v}|j} = 1 - J\big(\sqrt{j-1}\,J^{-1}(1 - I^{(\ell)}_{\mathrm{v}\to\mathrm{c}})\big), \tag{5.176}$$

   followed by calculating the average

   $$I^{(\ell)}_{\mathrm{c}\to\mathrm{v}} = \sum_{j\in\mathcal{D}_{\mathrm{c}}} \rho_j I^{(\ell)}_{\mathrm{c}\to\mathrm{v}|j}. \tag{5.177}$$

2. VN and APP LLR update: Calculate the VN EXIT updates according to the EXIT function (2.90),

   $$I^{(\ell+1)}_{\mathrm{v}\to\mathrm{s}|i,j} = J\Big(\sqrt{(i-1)\big(J^{-1}(I^{(\ell)}_{\mathrm{s}\to\mathrm{v}})\big)^2 + j\big(J^{-1}(I^{(\ell)}_{\mathrm{c}\to\mathrm{v}})\big)^2}\Big), \tag{5.178}$$

   $$I^{(\ell+1)}_{\mathrm{v}\to\mathrm{c}|i,j} = J\Big(\sqrt{i\big(J^{-1}(I^{(\ell)}_{\mathrm{s}\to\mathrm{v}})\big)^2 + (j-1)\big(J^{-1}(I^{(\ell)}_{\mathrm{c}\to\mathrm{v}})\big)^2}\Big), \tag{5.179}$$

   $$I^{(\ell+1)}_{\mathrm{v}\to\mathrm{s}} = \sum_{(i,j)\in\mathcal{D}_{\mathrm{v}}} \lambda^{s}_{ij} I^{(\ell+1)}_{\mathrm{v}\to\mathrm{s}|i,j}, \tag{5.180}$$

   $$I^{(\ell+1)}_{\mathrm{v}\to\mathrm{c}} = \sum_{(i,j)\in\mathcal{D}_{\mathrm{v}}} \lambda^{c}_{ij} I^{(\ell+1)}_{\mathrm{v}\to\mathrm{c}|i,j}. \tag{5.181}$$

   Calculate the APP mutual information

   $$I^{(\ell)}_{ij} = J\Big(\sqrt{(i)\big(J^{-1}(I^{(\ell)}_{\mathrm{s}\to\mathrm{v}})\big)^2 + j\big(J^{-1}(I^{(\ell)}_{\mathrm{c}\to\mathrm{v}})\big)^2}\Big), \tag{5.182}$$

   $$I^{(\ell)} = \sum_{(i,j)\in\mathcal{D}_{\mathrm{v}}} \Lambda_{ij} I^{(\ell)}_{ij}. \tag{5.183}$$

3. Set $\ell \mapsto \ell+1$.

4. If $1-I^{(\ell)} < \epsilon$ break and declare success. If not and $\ell < \mathscr{L}$, continue with step 1. Otherwise, declare failure.

Output: Success or failure of decoding. Furthermore, we output the *EXIT trajectories* $I^{(\ell')}_{\mathrm{s}\to\mathrm{v}|i,j,k}$, $I^{(\ell')}_{\mathrm{c}\to\mathrm{v}|j}$, $I^{(\ell')}_{\mathrm{v}\to\mathrm{s}|i,j}$, $I^{(\ell')}_{\mathrm{v}\to\mathrm{c}|i,j}$ and $I^{(\ell')}$ for all $\ell' < \ell$ and all degrees.

Hence, using the above EXIT evolution procedure, we can determine whether for a particular ensemble, iterative decoding converges for a channel noise parameter σ_{w}. According to the monotonicity theorem (Theorem 10), it follows that if the procedure is not successful for a particular σ_{w}, then it won't be successful for any $\sigma'_{\mathrm{w}} > \sigma_{\mathrm{w}}$. Consequently, we can perform a bisection search to obtain σ^*_{w}, where for a code of rate R, the initial search interval is given by $[0, \sigma_{\max,R}]$.

In case that the SN updates depend on the iteration index, we either have to precalculate (5.173) for any of those update rules or run the Monte Carlo simulation at any iteration for the current a priori information $I^{(\ell)}_{\mathrm{v}\to\mathrm{s}}$. Furthermore, note that for the bisection search, it might pay off in the long run to also precompute (5.173) for a range of supporting values of σ_{w}, thereby obtaining *EXIT surfaces* over $[0,1]\times[0,\sigma_{\max,R}]$, where again we can use (2D) interpolation in between the supporting values.

5.7.2 Local Optimization of Degree Distributions

We start this section by giving a brief overview of degree distribution optimization for conventional LDPC codes, i.e., we consider $\Sigma_{ijk} = \delta_{i-1}\delta_{j-1}\delta_{k-1}$ and $\Lambda_{ij} = \delta_{i-1}\Lambda^c_j$. For that case, the code rate (5.162) becomes

$$R = 1 - \frac{\sum_j \frac{\rho_j}{j}}{\sum_j \frac{\lambda^c_j}{j}}, \tag{5.184}$$

where $\lambda^c_j = \frac{j\Lambda^c_j}{\sum_j j\Lambda^c_j}$ is the edge perspective VN degree distribution of the CN side edges, cf. (2.49). A strategy that turned out to be very successful in producing good degree distributions is to optimize the code rate (5.184) under suitable linear constraints, cf. [18, 77, 79] and Section 4.3.3. Specifically, as discussed in Section 4.3.3, (5.184) can be maximized by minimizing $\sum_j \rho_j/j$ for constant λ^c or maximizing $\sum_j \lambda^c_j/j$ for a constant ρ, both of which are linear objective functions. Now consider the case where we want to optimize the VN distribution for a given CN distribution. Suitable linear constraints can be derived from the recursion (2.95), i.e., let $\mathcal{I} = [0,1]$ and define the functions

$$f_j : \mathcal{I} \to \mathcal{I},\ I^{(\ell)}_{\mathrm{v}\to\mathrm{c}} \mapsto I^{(\ell+1)}_{\mathrm{v}\to\mathrm{c}|j} \qquad \text{and} \qquad f(I) = \sum_j \lambda^c_j f_j(I) \tag{5.185}$$

according to the recursion (2.95). We then require that $f(I) \geq I$ for all $I \in \mathcal{I}$, which is a linear constraint in λ^c. Furthermore, $\sum_j \lambda^c_j = 1$ and $\lambda^c_j \geq 0$. This yields a linear program for optimizing the VN degree distribution. A similar linear program can be formulated for the CN degree distribution. Now alternating between VN and CN optimization, we quickly converge to very good degree distributions (λ^c, ρ).

For general SMLDPC codes, we can derive a similar optimization procedure, however, there are two aspects that complicate the extension:

1. The rate for a general code (5.162) is not linear in Σ or Λ, given all other degree distributions. However, (5.162) is a *linear-fractional function* of Σ given Λ and ρ, or in terms of Λ, a linear-fractional function of Λ given Σ and ρ. Thus, given that we can find suitable linear constraints, we can formulate *linear-fractional programs* that can be transformed to linear programs and solved efficiently [10, Section 4.3.2].

2. For conventional LDPC codes, the EXIT trajectories are one-dimensional for both VNs and CNs. Thus, the constraint $f(I) \geq I$ for all $I \in \mathcal{I}$ does not entail any loss of generality. For SMLDPC codes however, $I^{(\ell+1)}_{\mathrm{v}\to\mathrm{c}}$ does not only depend on $I^{(\ell)}_{\mathrm{v}\to\mathrm{c}}$ but also on $I^{(\ell)}_{\mathrm{v}\to\mathrm{s}}$, i.e., denoting $\boldsymbol{i}^{(\ell)}_{\mathrm{v}} = \begin{pmatrix} I^{(\ell)}_{\mathrm{v}\to\mathrm{s}} & I^{(\ell)}_{\mathrm{v}\to\mathrm{c}} \end{pmatrix}^{\mathrm{T}}$ the recursion takes the form

$$\left.\begin{aligned} f^s_{ij} &: \mathcal{I}^2 \to \mathcal{I},\ \boldsymbol{i}^{(\ell)}_{\mathrm{v}} \mapsto I^{(\ell+1)}_{\mathrm{v}\to\mathrm{s}|i,j} \\ f^c_{ij} &: \mathcal{I}^2 \to \mathcal{I},\ \boldsymbol{i}^{(\ell)}_{\mathrm{v}} \mapsto I^{(\ell+1)}_{\mathrm{v}\to\mathrm{c}|i,j} \end{aligned}\right\} \quad \boldsymbol{f} : \mathcal{I}^2 \to \mathcal{I}^2,\ \boldsymbol{i}^{(\ell)}_{\mathrm{v}} \mapsto \begin{pmatrix} \sum_i \sum_j \lambda^s_{ij} f^s_{ij}\big(\boldsymbol{i}^{(\ell)}_{\mathrm{v}}\big) \\ \sum_i \sum_j \lambda^c_{ij} f^c_{ij}\big(\boldsymbol{i}^{(\ell)}_{\mathrm{v}}\big) \end{pmatrix}. \tag{5.186}$$

Here, the two-dimensional *EXIT vector field* $\boldsymbol{f}$ and the component functions f^s_{ij} and f^c_{ij} are obtained by combining (5.174) to (5.181). Consequently, $\boldsymbol{f}$ is parametrized by σ_{w}, (Σ, Λ, ρ) as well as $p_{\mathsf{A}|i,j,k}$ and $\underline{\Phi}_{i,j,k}$. We could thus formulate linear constraints by requiring that $\boldsymbol{f}(\boldsymbol{i}) \succcurlyeq \boldsymbol{i}$ for all $\boldsymbol{i} \in \mathcal{I}^2$. This condition is however prohibitively restrictive and prevents good solutions because in the two dimensional case, $\boldsymbol{f}(\boldsymbol{i}^{(\ell)}_{\mathrm{v}}) \succcurlyeq \boldsymbol{i}^{(\ell)}_{\mathrm{v}}$ merely along the one dimensional EXIT trajectory $\boldsymbol{i}^{(\ell)}_{\mathrm{v}}$. This problem has also been observed in [121] for the optimization of degree distributions for multi-edge-type LDPC codes where an approach based on the *backward difference vector field*

$$\nabla \boldsymbol{f} :\ \mathcal{I}^2 \to \mathcal{I}^2, \quad \boldsymbol{i} \mapsto \boldsymbol{f}(\boldsymbol{i}) - \boldsymbol{i} \tag{5.187}$$

is suggested. Specifically, it is required that $\nabla \boldsymbol{f}(\boldsymbol{i}) \neq 0$ for all $\boldsymbol{i} \in \mathcal{I}^2 \setminus \mathbf{1}$, i.e., there are no stationary points other than $\mathbf{1}$, which can be shown to imply convergence. In practice, this condition is modeled as $||\nabla \boldsymbol{f}(\boldsymbol{i})||^2 \geq \epsilon$ for small ϵ and a finite, sufficiently dense grid of supporting values $\boldsymbol{i} \in \hat{\mathcal{I}}^2 \subset \mathcal{I}^2$, which translates into $|\hat{\mathcal{I}}^2|$ quadratic constraints. Figure 5.10 depicts an example of a backward difference vector field and a decoding trajectory $\boldsymbol{i}^{(\ell)}_{\mathrm{v}}$ for an optimized, irregular SMLDPC code of rate 2 based on real SM-EPA coefficients, cf. (5.23). Observe that $\boldsymbol{f}(\boldsymbol{i}^{(\ell)}_{\mathrm{v}}) \succcurlyeq \boldsymbol{i}^{(\ell)}_{\mathrm{v}}$ along the decoding trajectory but not in general.

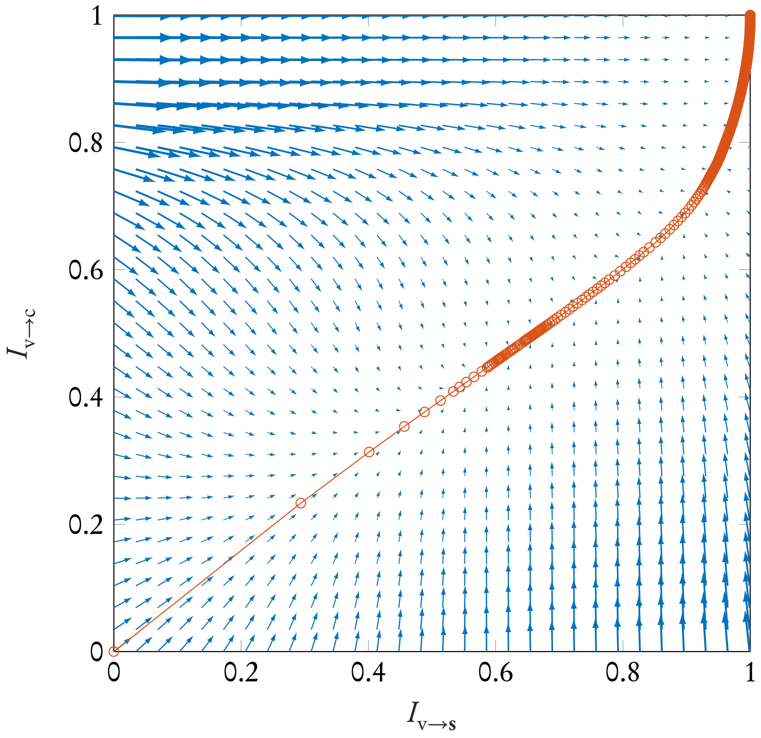


Figure 5.10: Backward difference vector field (scaled by a factor 0.1 for better visibility) and decoding trajectory using a supporting set $\hat{\mathcal{I}}^2$ with 30 points per dimension. As we can see, along the decoding trajectory, the vector field almost vanishes. In particular, around the area of the point (.65, .5), the field is close to being stationary. This can also be seen by examining the point density of the decoding trajectory, which indicates that it took many iteration to cross that region.

For multi-edge-type LDPC codes, the objective is linear and thus the resulting optimization problem is linear, non convex and quadratically constrained and due to its well behaved objective, can be solved using general purpose interior point methods. In our case however, the objective is not linear and a transformation to a linear problem as described in item 1 is only possible for linear constraints. We tried general purpose solvers as suggested in [121] but could not obtain satisfactory results for our objective using random initializations for optimization. However, using "good" initializations obtained from the optimization approach we present in the following, this method is capable of producing good degree distributions.

In what follows, we will derive a strategy that works around those limitations. The idea is that we assume that for small changes to the degree distribution, decoding still follows essentially the same EXIT trajectory. This is an assumption commonly made for changes to the degree distribution and resulting densities of conventional LDPC codes [18, 77].

We start with an initial degree distribution Λ' and obtain the corresponding EXIT trajectories for the highest possible σ_{w} for which EXIT evolution still converges. Denote those trajectories as $I^{(\ell)}_{\text{v}\to\text{s}|i,j}$, $I^{(\ell)}_{\text{v}\to\text{c}|i,j}$ and $I^{(\ell)}_{\text{v}\to\text{s}} = \sum_{i,j} \lambda^{s\prime}_{ij} I^{(\ell)}_{\text{v}\to\text{s}|i,j}$, $I^{(\ell)}_{\text{v}\to\text{c}} = \sum_{i,j} \lambda^{c\prime}_{ij} I^{(\ell)}_{\text{v}\to\text{c}|i,j}$. Then for any point along that trajectory, we consider one more iteration using a new, slightly different degree distribution Λ that maximizes the code rate under the constraint that

1. the new degree distribution is a proper pmf,

2. the new degree distribution is close to the original one,

3. the trajectory of the new degree distribution improves upon the old one.

This translates into the following optimization problem for the VN degree distribution Λ

$$\max_{\Lambda} \quad \frac{C_\Sigma^s\big(1 - C_\rho \sum_{(i,j)\in\mathcal{D}_\mathrm{v}} j\Lambda_{ij}\big)}{C_\Sigma^c \sum_{(i,j)\in\mathcal{D}_\mathrm{v}} i\Lambda_{ij}} \tag{5.188}$$

subject to

$$\sum_{(i,j)\in\mathcal{D}_\mathrm{v}} \Lambda_{ij} = 1, \tag{5.189}$$

$$\Lambda_{ij} \geq 0, \quad (i,j)\in\mathcal{D}_\mathrm{v}, \tag{5.190}$$

$$\left| I_{\mathrm{v}\to\mathrm{s}}^{(\ell)} - \sum_{(i,j)\in\mathcal{D}_\mathrm{v}} I_{\mathrm{v}\to\mathrm{s}|i,j}^{(\ell)} \lambda_{ij}^s \right| \leq \delta \left[I_{\mathrm{v}\to\mathrm{s}}^{(\ell)} - I_{\mathrm{v}\to\mathrm{s}}^{(\ell-1)} \right]^+, \tag{5.191}$$

$$\left| I_{\mathrm{v}\to\mathrm{c}}^{(\ell)} - \sum_{(i,j)\in\mathcal{D}_\mathrm{v}} I_{\mathrm{v}\to\mathrm{c}|i,j}^{(\ell)} \lambda_{ij}^c \right| \leq \delta \left[I_{\mathrm{v}\to\mathrm{c}}^{(\ell)} - I_{\mathrm{v}\to\mathrm{c}}^{(\ell-1)} \right]^+, \tag{5.192}$$

$$\sum_{(i,j)\in\mathcal{D}_\mathrm{v}} I_{\mathrm{v}\to\mathrm{s}|i,j}^{(\ell)} \lambda_{ij}^s \geq I_{\mathrm{v}\to\mathrm{s}}^{(\ell-1)} + \epsilon, \tag{5.193}$$

$$\sum_{(i,j)\in\mathcal{D}_\mathrm{v}} I_{\mathrm{v}\to\mathrm{c}|i,j}^{(\ell)} \lambda_{ij}^c \geq I_{\mathrm{v}\to\mathrm{c}}^{(\ell-1)} + \epsilon, \tag{5.194}$$

where item 1 corresponds to the constraints (5.189) and (5.190), item 2 corresponds to (5.191) and (5.192), item 3 corresponds to (5.193) and (5.194), $\epsilon \geq 0$ and $\delta > 0$ are positive constants that control closeness and improvement and the constants $C_\Sigma^s = \sum_{(i,j,k)\in\mathcal{D}_\mathrm{s}} j\Sigma_{ijk}$, $C_\Sigma^c = \sum_{(i,j,k)\in\mathcal{D}_\mathrm{s}} i\Sigma_{ijk}$ and $C_\rho = \sum_{j\in\mathcal{D}_\mathrm{c}} \frac{\rho_j}{j}$ are independent of Λ.

Applying the Charnes-Cooper transformation [15], we introduce a new variable

$$t \triangleq \frac{1}{C_\Sigma^c \sum_{(i,j)\in\mathcal{D}_\mathrm{v}} i\Lambda_{ij}} > 0 \tag{5.195}$$

and transform $x_{ij} = t\Lambda_{ij}$ to arrive at

$$\begin{aligned}
&\max_{x,t} \quad C_{\Sigma}^{s}\big(t - C_{\rho} \sum_{(i,j)\in\mathcal{D}_{\mathrm{v}}} j x_{ij}\big) \\
&\text{subject to} \\
&\quad \sum_{(i,j)\in\mathcal{D}_{\mathrm{v}}} x_{ij} = t, \\
&\quad x_{ij} \geq 0, \\
&\quad t \geq 0, \\
&\quad C_{\Sigma}^{c} \sum_{(i,j)\in\mathcal{D}_{\mathrm{v}}} i x_{ij} = 1, \\
&\quad \sum_{(i,j)\in\mathcal{D}_{\mathrm{v}}} \Big(i I_{\mathrm{v}\to\mathbf{s}}^{(\ell)} - i I_{\mathrm{v}\to\mathbf{s}|i,j}^{(\ell)} - i\delta \big[I_{\mathrm{v}\to\mathbf{s}}^{(\ell)} - I_{\mathrm{v}\to\mathbf{s}}^{(\ell-1)} \big]^{+} \Big) x_{ij} \leq 0, \\
&\quad \sum_{(i,j)\in\mathcal{D}_{\mathrm{v}}} \Big(-i I_{\mathrm{v}\to\mathbf{s}}^{(\ell)} + i I_{\mathrm{v}\to\mathbf{s}|i,j}^{(\ell)} - i\delta \big[I_{\mathrm{v}\to\mathbf{s}}^{(\ell)} - I_{\mathrm{v}\to\mathbf{s}}^{(\ell-1)} \big]^{+} \Big) x_{ij} \leq 0, \\
&\quad \sum_{(i,j)\in\mathcal{D}_{\mathrm{v}}} \Big(j I_{\mathrm{v}\to\mathrm{c}}^{(\ell)} - j I_{\mathrm{v}\to\mathrm{c}|i,j}^{(\ell)} - j\delta \big[I_{\mathrm{v}\to\mathrm{c}}^{(\ell)} - I_{\mathrm{v}\to\mathrm{c}}^{(\ell-1)} \big]^{+} \Big) x_{ij} \leq 0, \\
&\quad \sum_{(i,j)\in\mathcal{D}_{\mathrm{v}}} \Big(-j I_{\mathrm{v}\to\mathrm{c}}^{(\ell)} + j I_{\mathrm{v}\to\mathrm{c}|i,j}^{(\ell)} - j\delta \big[I_{\mathrm{v}\to\mathrm{c}}^{(\ell)} - I_{\mathrm{v}\to\mathrm{c}}^{(\ell-1)} \big]^{+} \Big) x_{ij} \leq 0, \\
&\quad \sum_{(i,j)\in\mathcal{D}_{\mathrm{v}}} \Big(i I_{\mathrm{v}\to\mathbf{s}}^{(\ell-1)} + i\epsilon - i I_{\mathrm{v}\to\mathbf{s}|i,j}^{(\ell)} \Big) x_{ij} \leq 0, \\
&\quad \sum_{(i,j)\in\mathcal{D}_{\mathrm{v}}} \Big(j I_{\mathrm{v}\to\mathrm{c}}^{(\ell-1)} + j\epsilon - j I_{\mathrm{v}\to\mathrm{c}s|i,j}^{(\ell)} \Big) x_{ij} \leq 0.
\end{aligned}$$

Then the transformed problem is linear and the solution x_{ij} and t yields the solution of the original problem by inverting the transformation, $\Lambda_{ij} = x_{ij}/t$. A similar program can be formulated in terms of the SN degree distribution Σ_{ijk}. Specifically, if Σ has a fixed, regular dimension d_y, i.e., $\Sigma_{ijk} = \delta_{i-d_y}\Sigma_{jk}$ the rate expression takes the form

$$R = \frac{1 - \big(\sum_{j\in\mathcal{D}_{\mathrm{c}}} \frac{\rho_j}{j}\big)\big(\sum_{(i,j)\in\mathcal{D}_{\mathrm{v}}} j\Lambda_{ij}\big)}{\big(\sum_{(i,j)\in\mathcal{D}_{\mathrm{v}}} i\Lambda_{ij}\big)\big(d_y \sum_{j\in\mathcal{D}_{\mathrm{s}}^{s}} \frac{\sigma_j}{j}\big)}, \tag{5.196}$$

which allows for joint optimization of the SN and CN degree distribution since (5.196) is a linear-fractional function in (σ, ρ). If Σ does not have regular dimension, then we have to optimize Σ assuming constant ρ and Λ. In that case, ρ has to be optimized separately using an analogous linear program, assuming fixed Λ and Σ.

Thus, to sum up the degree distribution optimization procdedure, we start with an initial triple (Σ, Λ, ρ) and run an EXIT evolution based bisection search to obtain the highest σ_{w} for which decoding still converges and save the corresponding EXIT traces. Subsequently, we consider linearization around the so obtained trajectories and iteratively solve linear programs optimizing the code rate, where we alternate between optimizing VN, SN and CN degree distributions. If the SN degree distribution has regular channel dimension, we alternate between optimizing the VN degree distribution and joint optimization of both SN and CN degree distribution. Since we allow only small changes to the degree distributions, the EXIT trajectories of the new degree distributions still essentially follow the old trajectories. Thus, after solving the optimization problems, the new trajectories are obtained from the old ones by running one more iteration using the optimized degree distributions. We repeat the above procedure until there is no more improvement in the objective.

5.7.3 Global Optimization of Degree Distributions

In the previous section, we presented an optimization method for degree distributions of irregular SMLDPC ensembles. The procedure is a local gradient type optimization algorithm and therefore requires an initial degree distribution to start with. Originally, the degree distribution optimization of conventional LDPC codes faced the same problem [77]. However, it was quickly realized that the global optimization step can be omitted since empirical results of degree distribution optimization indicated that

- optimal CN degree distributions tend to concentrate around a single degree
- optimal VN degree distributions are sparse, feature high fractions of low degree nodes with degrees $2,3,\dots,10$ as well as the maximum allowed VN degree and only a few degrees in between

Thus, in the face of the above results, it turned out to be sufficient to omit global optimization altogether and start with random initializations that adhere to the above structure [18].

For SMLDPC codes, the situation is different in that we do not yet have an understanding as to which degree distributions perform well for a given edge coefficient distribution. Furthermore, since the VN degree is two dimensional, we need to identify which combinations of VN-to-SN and VN-to-CN degrees yield good performance at high rates. For the SN degree distribution, we face a similar problem.

For that matter, we use *differential evolution* [94], a method that has already been employed successfully to find good LDPC codes [77, 89]. Our main objective here is to identify good degree combinations. The exact probability of that combinations can be optimized later on using the local optimization approach described in Section 5.7.2. Since differential evolution works best over continuous domains of not too high dimension, we consider tuples

$$(\Sigma_{ijk}, i, j) \in [0,1] \times [1, d_{y,\max}] \times [1, d_{s,\max}] \subset \mathbb{R}^3 \quad \forall k, \tag{5.197}$$

$$(\Lambda_{ij}, i, j) \in [0,1] \times [0, d_{vs,\max}] \times [0, d_{vc,\max}] \subset \mathbb{R}^3, \tag{5.198}$$

$$(\rho_j, j) \in [0,1] \times [1, d_{c,\max}] \subset \mathbb{R}^2, \tag{5.199}$$

i.e. continuous node degrees. Note that we did not introduce a continuous node type as the numeric index of the type is irrelevant to the rate and furthermore, we cannot assume continuity when going from type k to the next type $k+1$ and therefore, a linear combination as performed by the differential evolution algorithm would not be meaningful for the node type. Consequently, the node types are not part of the differential evolution search space and must be regarded as fixed parameters set in advance.

Let $\lceil i \rceil \in \mathbb{N}$ denote the integer closest to $i \in \mathbb{R}$ and $\lfloor i \rfloor \neq \lceil i \rceil$ denote the second closest integer to i. Then from the continuous VN tuple we obtain

$$(\Lambda_{ij}, i, j) \Rightarrow \begin{cases} (\Lambda_{\lceil i \rceil \lceil j \rceil}, \lceil i \rceil, \lceil j \rceil) \\ (\Lambda_{\lfloor i \rfloor \lceil j \rceil}, \lfloor i \rfloor, \lceil j \rceil) \\ (\Lambda_{\lceil i \rceil \lfloor j \rfloor}, \lceil i \rceil, \lfloor j \rfloor) \end{cases} \quad \text{with} \quad \begin{pmatrix} i\Lambda_{ij} \\ j\Lambda_{ij} \\ \Lambda_{ij} \end{pmatrix} = \begin{pmatrix} \lceil i \rceil & \lfloor i \rfloor & \lceil i \rceil \\ \lceil j \rceil & \lceil j \rceil & \lfloor j \rfloor \\ 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} \Lambda_{\lceil i \rceil \lceil j \rceil} \\ \Lambda_{\lfloor i \rfloor \lceil j \rceil} \\ \Lambda_{\lceil i \rceil \lfloor j \rfloor} \end{pmatrix}, \tag{5.200}$$

i.e., a continuous triple splits into three different integer degree pairs with corresponding weights that are uniquely determined by requiring that the code rate as well as the sum probability remain unaffected by the split. Similarly, for the SN and CN tuples,

$$(\Sigma_{ijk}, i, j) \Rightarrow \begin{cases} (\Sigma_{\lceil i \rceil \lceil j \rceil k}, \lceil i \rceil, \lceil j \rceil) \\ (\Sigma_{\lfloor i \rfloor \lceil j \rceil k}, \lfloor i \rfloor, \lceil j \rceil) \\ (\Sigma_{\lceil i \rceil \lfloor j \rfloor k}, \lceil i \rceil, \lfloor j \rfloor) \end{cases} \quad \text{with} \quad \begin{pmatrix} i\Sigma_{ijk} \\ j\Sigma_{ijk} \\ \Sigma_{ijk} \end{pmatrix} = \begin{pmatrix} \lceil i \rceil & \lfloor i \rfloor & \lceil i \rceil \\ \lceil j \rceil & \lceil j \rceil & \lfloor j \rfloor \\ 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} \Sigma_{\lceil i \rceil \lceil j \rceil k} \\ \Sigma_{\lfloor i \rfloor \lceil j \rceil k} \\ \Sigma_{\lceil i \rceil \lfloor j \rfloor k} \end{pmatrix}, \tag{5.201}$$

$$(\rho_j, j) \Rightarrow \begin{cases} (\rho_{\lceil j \rceil}, \lceil j \rceil) \\ (\rho_{\lfloor j \rfloor}, \lfloor j \rfloor) \end{cases} \quad \text{with} \quad \begin{pmatrix} \rho_j / j \\ \rho_j \end{pmatrix} = \begin{pmatrix} 1/\lceil j \rceil & 1/\lfloor j \rfloor \\ 1 & 1 \end{pmatrix} \begin{pmatrix} \rho_{\lceil j \rceil} \\ \rho_{\lfloor j \rfloor} \end{pmatrix}. \tag{5.202}$$

Prior to running differential evolution, we specify the number of *active degrees* per node type, i.e., how many continuous tuples in the form of (5.197) to (5.199) to consider. Let $d_{s,\text{act}} = |\mathcal{D}_\text{s}|$, $d_{v,\text{act}} = |\mathcal{D}_\text{v}|$ and $d_{c,\text{act}} = |\mathcal{D}_\text{c}|$ denote the number of active SN, VN and CN degrees (i.e. the number of degrees with non zero probability). The dimension of the search space is then given by

$$3d_{s,\text{act}} + 3d_{v,\text{act}} + 2d_{c,\text{act}}, \tag{5.203}$$

assuming that we search over SN, VN as well as CN degree distributions. For the objective function, we once again use the code rate (5.162).

In Table 5.2 we show a degree distribution that we obtained using differential evolution followed by local optimization for fixed $d_y = 1$, $d_s = 8$ and $d_c = 10$ for SM-EPA coefficients. The code rate of the ensemble is $R = 1.9$, corresponding to the maximum noise standard deviation $\sigma_{\text{max},R} = 0.2781$. Observe how the optimization procedure yields a VN degree distribution with correlated VN-to-SN and VN-to-CN degrees. Consequently, the joint design of VN-to-SN and VN-to-CN degree structure is crucial for the design of good SMLDPC codes. In

i	3	0	1	0	8	2	3
j	11	2	2	3	20	3	3
Λ_{ij}	0.014497	0.224958	0.039526	0.134598	0.125895	0.363348	0.097177

Table 5.2: VN degree distribution obtained by first running differential evolution over the search space of VN degree distributions with $d_{v,\mathrm{act}} = 3$ continuous active degrees and maximum degrees $d_{vs,\max} = d_{vc,\max} = 20$ for a regular fixed SN degree of $d_s = 8$ and regular fixed CN degree of $d_c = 10$. We used real SM-EPA coefficients (cf. (5.23)) and the optimal MAP SN update (5.46). After the differential evolution search, we employed the local optimization algorithm presented in Section 5.7.2 with the differential evolution solution as starting point.

particular, it is interesting to note that the optimized distribution contains degree zero nodes, i.e., punctured VNs that are only involved in parity checks but are not explicitly present in the modulated symbols. Similarly to LDPC codes, there is a high amount of low degree VNs as well as VNs with the highest allowed degrees.

Using bisection search, we obtained the ensemble threshold $\sigma_{\mathrm{w}}^* = 0.2657$, which is 0.908 dB from capacity. Based on this ensemble, we also created an SMLDPC code of length $N = 10^5$ and rate $R = 1.9$ and simulated transmission over an AWGN channel, cf. Figure 5.11. The code was obtained using the random code construction method proposed in [65] for LDPC codes. That is, starting with the lowest degree VNs, edges are placed randomly by setting elements of Γ to one on a per-column basis, cf. (5.6). Upon insertion of an edge, we check if a short cycle was created and if so, we remove the edge and place it else where.

As we can see from Figure 5.11, this code already comes fairly close to capacity, even though we did not perform any SN or CN degree distribution optimization. Specifically, we chose $d_s = 8$ and $d_c = 10$ to compare the resulting optimized code to the performance of the heuristically designed rate 2 code presented in [115]. Whereas we obtain similar performance in the water fall region, in contrast to [115], we did not observe any flattening or error floor using our design.

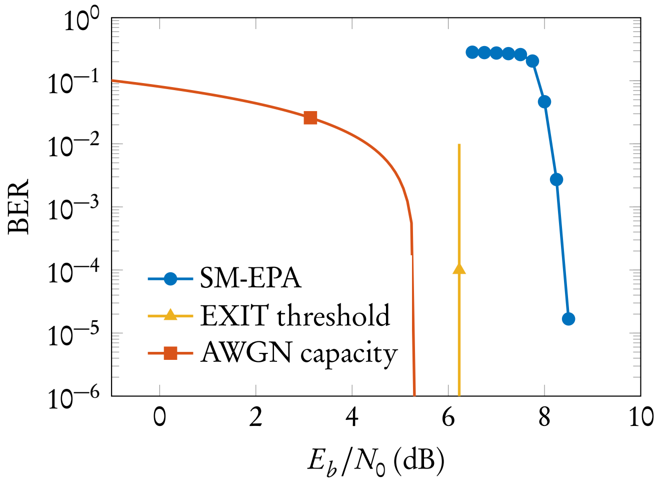


Figure 5.11: Bit error rate and threshold for an optimzized code using SM-EPA coefficients.

Chapter 6
Conclusion and Outlook

In this concluding chapter, we first provide a chapter-by-chapter summary of our most relevant findings. Subsequently, we discuss potential extensions and topics for future work.

6.1 Summary of Contributions

Chapter 3: Rate Information Coding for Gaussian Signals

- We analyzed the optimal rate information tradeoff for jointly Gaussian observation and relevance in a communication context. This formulation highlights the effects of the GIB information-optimal projection and yields closed-form expressions for the rate information tradeoff in terms of the mode SNRs.

- Based on the communication model, we showed that the GIB optimal projection is closely related to the associated Wiener filter for estimating the transformed relevance from the channel output.

- We derived and analyzed the rate information tradeoff for linearly prefiltered MSE-optimal quantization and showed that the information-optimal tradeoff as characterized by the GIB is obtained by choosing the square root of the Wiener filter, as opposed to direct Wiener filtering, which results into the MSE-optimal tradeoff. This result establishes a fundamental relation between MSE-optimal and information-optimal quantization for the Gaussian case.

- We established a coding theorem for information-optimal quantization of a sequence of Gaussian vectors and discussed the extension to stationary ergodic sources. Furthermore, we proved the existence of information-optimal quantizers derived from MSE-optimal quantizers based on the equivalence of the optimal probabilistic mapping $p(z|y)$ for both cases.

- We presented simulation results indicating that MSE-optimal quantization of square-root-Wiener-filtered signals indeed shows the best rate information performance amongst comparable filters.

Chapter 4: Quantized Message Passing LDPC Decoding

- We formulated a novel symmetry concept for LUT decoding based on involutions and the pmfs of discrete messages in density evolution. It has been previously observed in [60] that without symmetry of LUT updates, the resulting decoder performed poorly, so symmetry was introduced there based on empirical observations of decoder performance. Our results provide a theoretical backing to that observation.

- We showed that LUT decoding is essentially a form of quantized BP, where LUTs take the role of iteration-dependent quantizers. This result establishes a connection between LUT decoding and BP and allows concepts that have been originally developed for BP to be applied to LUT decoding. Furthermore, we established a duality between the discrete LUT labels and the associated LLR-values that are obtained from density evolution pmfs. This serves to create a rudimentary algebraic structure on the message labels which allows for a combination of LUT based and algebraic node updates.

- We derived an information-optimal symmetric LUT decoding algorithm for LDPC codes. Due to the algebraic structure on the labels, LUT inputs can be preprocessed which leads to a reduction in LUT size and complexity. Moreover, we proposed a hybrid min-LUT algorithm that combines the algebraic min-sum (MS) CN update rule with a VN LUT update.

- We proposed a variety of different measures to further reduce the complexity of LUT decoders and investigated their impact on decoder performance. In particular, we demonstrated that LUTs can be arranged into binary trees of maximum width with negligible impact on decoder performance. Furthermore, we showed that a structured reuse of LUTs for multiple decoding iterations can reduce implementation complexity by up to 80% with minor performance penalties. We found that downsizing the LUT resolution over the course of several decoding iterations decreases the LUT size and causes only an insignificant performance degradation.

- We derived the first extension of LUT decoding to irregular LDPC codes. By taking into account degree distributions, we proposed an information-optimal joint design for LUTs of different node degrees and demonstrated the far superior performance of this approach as compared to the individual design. Moreover, based on the relation we found between LUT and BP decoding, we showed that LUT decoders fail for BP-optimized codes due to a mismatch in the degree structure. In particular, we identified that this is due to an excess in degree two VNs and we quantified the necessary corrections by means of asymptotic stability analysis. Considering the bound on the fraction of degree two VNs obtained that way, we optimized degree distributions for LUT decoding and showed that LUT decoders can in fact take advantage of irregular codes.

- We presented extensive numeric simulation results comparing different LUT decoders and conventional BP and MS decoders. Specifically, our results show that for regular codes, a LUT resolution of only 3 bit is sufficient for the low complexity min-LUT decoder to reach the performance of conventional MS decoders operating at floating point precision, while outperforming MS decoders with 4 bit message resolution by 0.3 dB and more in the error floor region. Using a 4 bit LUT decoder, we come within 0.1 dB of BP performance. For irregular codes, the requirements for the LUT resolution are higher than for regular codes. Nonetheless, we demonstrated using a standard compliant irregular code that a min-LUT decoder with 4 bit resolution can outperform a floating point MS decoder by 0.5 dB.

Chapter 5: SMLDPC Codes

- We proposed SMLDPC codes — a novel type of potentially capacity approaching codes for continuous input additive white noise channels that allow for joint parallel decoding and demodulation with complexity linear in code word length. Formally, an SMLDPC code is completely determined by a sparse tripartite graph $G(\Gamma)$ and a series of edge coefficient matrices $(A_l)_{l=1}^{L}$. Using different edge coefficients, a wide variety of well-known modulation schemes can be adopted by SMLDPC codes, including the widely used regular QAM constellations.

- We presented a generic message passing decoding approach for SMLDPC codes and derived the BP decoding algorithm. Due to the structure of the underlying graph, SMLDPC codes allow for joint parallel demodulation and decoding in linear time — even MIMO detection can be incorporated into the decoding algorithm. The most complex operation of the decoding algorithm was identified as the SN update, for which various approaches were proposed. In particular, we derived a low complexity approximate SN update that is similar to a linear MMSE estimator with a priori information.

- We considered a random construction of SMLDPC codes based on random sampling of the edge coefficient matrices $(\mathbf{A}_l)_{l=1}^{L}$ and construction of a random graph $\mathsf{G}(\mathbf{\Gamma})$. We showed that for any particular realization of graph, edge coefficients, channel realization and transmit message, the fraction of bit errors under message passing decoding converges to the average of the cycle-free case as the length of the code approaches infinity. Then due to the monotonicity of the BP decoding algorithm, this concentration result gives rise to a thresholding phenomenon, i.e., for the AWGN channel with noise variance σ^2, there exists σ^* such that for any $\sigma < \sigma^*$ decoding of any codeword from any code disturbed by any noise realization will be successful as $L \to \infty$, whereas for $\sigma > \sigma^*$ the opposite is true. Furthermore, we demonstrated that for BP decoding, σ^* is computable exactly by means of EXIT analysis. More precisely, we showed that for BP, the messages have iid symmetric and LLR-consistent densities which gives rise to a very accurate method for obtaining the EXIT function of the SN updates.

- Based on the concentration theorem and the resulting thresholding phenomenon, we proposed to search for ensembles of irregular of SMLDPC codes that exhibit high thresholds σ^* close to $\sigma_{\max,R}$, which is the maximum noise parameter for a rate R code according to the AWGN capacity. That is, we searched for degree- and coefficient distribution tuples $(\Sigma, \Lambda, \rho, p_{\mathsf{A}|i,j,k})$ such that the corresponding threshold $\sigma^*(\Sigma, \Lambda, \rho, p_{\mathsf{A}|i,j,k})$ is close to $\sigma_{\max,R}$. Towards this end, we derived an EXIT evolution procedure to accurately determine σ^* for prescribed $(\Sigma, \Lambda, \rho, p_{\mathsf{A}|i,j,k})$. Moreover, we suggested global and local optimization techniques to find ensembles $(\Sigma, \Lambda, \rho, p_{\mathsf{A}|i,j,k})$ with high rate for any prescribed noise parameter σ. Once we found a good ensemble, the concentration theorem guarantees that any random long code sampled from the ensemble will have performance close to the predicted threshold. We exemplified our optimization strategy by optimizing Λ for fixed Σ, ρ and $p_{\mathsf{A}|i,j,k}$, obtaining an ensemble with threshold within 0.9 dB from capacity and corresponding randomly generated code.

6.2 Open Problems for Future Research

Chapter 3: Rate Information Coding for Gaussian Signals

- In Section 3.2.3, we calculated the information rate tradeoff for the limit of infinitely long vectors, cf. (3.67) and (3.68). While we were able to state a coding theorem for a sequence of iid Gaussian vectors in Section 3.3.1, extending this result to the case of stationary ergodic sequences is still an open problem.

- We discovered in Section 3.3.2 that the rate information tradeoff is not suited to characterize the capacity under channel output quantization. For scalar quantizers, we found that the capacity is characterized by the double optimization problem (3.81). Attempts have been made to solve this problem for symmetric quantizers in [90]. However, it has been shown in [54] that, somewhat surprisingly, symmetric quantizers are note necessarily optimal for that setting. Thus, finding the solution to (3.81) is still an open issue.

- In Section 3.3.3, we found that the relation between the GIB and prefiltered MSE-optimal quantization does imply the existence of an optimal rate information code derived from an MSE-optimal code. However, a much stronger result would be possible if one were able to find conditions such that equality holds in (3.84).

Chapter 4: Quantized Message Passing LDPC Decoding

- An interesting open question is whether a similar, LUT based decoding approach can be obtained for turbo codes. Other than that, we originally delved into SMLDPC codes searching for an extension of the LUT based decoding approach to iterative demodulation. Since both turbo decoding and evaluating the SN MAP updates can be based on the BCJR algorithm [46, 64], we believe that a viable approach to those problems would be to try to formulate a LUT based version of the BCJR algorithm. Considering that the intermediate LUTs in the CN LUT trees of Section 4.2.3 could be interpreted as "box-plus and quantize" operations, a possible approach to discretizing the BCJR algorithm might involve a "Jacobian logarithm and quantize" operation. In fact, using constant LUTs to evaluate the Jacobian logarithm on a digital signal processor is a fairly common approach in practice. One would need to investigate the potential of using adaptive, information-optimal LUTs as opposed to constant LUTs, or more generally, the differences in performance if LUTs were eliminated altogether in favor of a hybrid max-LUT algorithm that neglects the residual of the Jacobian logarithm. For that, we would need to have a similar algebraic structure on the LUT labels in place that enabled the hybrid min-LUT algorithm. Furthermore, for the case of the SN MAP updates, LUTs for the initialization of the backward messages β could be of interest.

- For degree distribution optimization of irregular codes for LUT decoding, we mainly considered an approach based on error probability traces as originally proposed in [77], cf. (4.49) and (4.50). An approach based on EXIT traces as illustrated in Section 5.7.2 might be interesting as well, since it appears more global and less dependent on good initializations. This is because for the EXIT based approach, we do not require that updated degree distributions are close to the previous ones.

- Quantized decoding has often been associated with an increase in error floors [74, 123, 124]. One of the features of our approach towards LUT decoding is that the LLR values are allowed to grow implicitly during decoding by using iteration dependent LUTs, cf. Section 4.1.1, which helps to avoid error floors [124]. While we did not observe any error floors for bit error rates up to 10^{-13} using the 10GBaseT LDPC code [33], a thorough investigation would involve testing our decoders for an even higher number of transmissions by implementing them onto FPGAs or comparably fast technologies.

Chapter 5: SMLDPC Codes

- The search for SMLDPC ensembles $(\Sigma, \Lambda, \rho, p_{\mathsf{A}|i,j,k})$ with thresholds close to capacity is a fascinating and complicated open problem that we are currently still working on. In contrast to conventional LDPC codes, the parameter space is much larger due to multi dimensional degree structure and the edge coefficient distributions. Unfortunately, the rate is not linear in the degree distributions and the edge coefficient distribution $p_{\mathsf{A}|i,j,k}$ does not enter into the rate at all. While it is clear that the capacity is maximized if $p_{\mathsf{A}|i,j,k}$ is chosen such that the resulting constellation is Gaussian, this does not necessarily result in matched EXIT charts and good iterative decoding performance.

- We did not yet have the opportunity to use the degree distribution optimization techniques developed in this thesis to try and design SMLDPC codes using well known edge coefficient distributions. E.g., we could select $\Lambda_{ij} = \delta_{i-1}\Lambda^c_j$ and design conventional irregular LDPC codes for QAM constellations and then compare our results to the ones existing in the literature, e.g. [98].

- The theory of SMLDPC codes would benefit greatly from an explicit expression for the transformation of densities due to the SN update, cf. (5.145). We obtained an explicit transformation for the simple case of independent edge coefficients and scalar Gaussian approximation updates (5.59) by means of two dimensional Fourier transforms and transformation of random variables. However, we do not believe that for the general update rule (5.46) we can obtain a transformation based on transforming the input densities according to the manipulations of random variables in (5.46). If a general transformation is to be found at all, it might come from using other arguments, e.g., the optimality of the update (5.46).
- The theory of asymptotic stability for BP decoding of LDPC codes (cf. [77]) needs to be extended to SMLDPC codes.
- Efficient graph construction algorithms such as the PEG algorithm [49] are required for the generation of SMLDPC codes.
- For even finer granular control over the graph structure of SMLDPC codes, one could consider using a multi-edge type approach similar to the one for LDPC codes in [76].
- Instead of binary LDPC codes, one could also consider higher order finite fields for the construction of SMLDPC codes, cf. [24].

Bibliography

[1] R. Ahlswede and I. Csiszár, "Hypothesis testing with communication constraints", *IEEE Trans. Information Theory*, vol. 32, no. 4, pp. 533–542, Jul. 1986 (cit. on p. 16).

[2] S. Arimoto, "An algorithm for computing the capacity of arbitrary discrete memoryless channels", *IEEE Trans. Information Theory*, vol. 18, no. 1, pp. 14–20, Jan. 1972 (cit. on p. 15).

[3] A. Ashikhmin, G. Kramer, and S. ten Brink, "Extrinsic information transfer functions: Model and erasure channel properties", *IEEE Trans. Information Theory*, vol. 50, no. 11, pp. 2657–2673, Nov. 2004 (cit. on pp. 33, 125).

[4] K. Azuma, "Weighted sums of certain dependent random variables", *Tohoku Mathematical Journal*, vol. 19, no. 3, pp. 357–367, 1967 (cit. on p. 108).

[5] L. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate", *IEEE Trans. Information Theory*, vol. 20, no. 2, pp. 284–287, Mar. 1974 (cit. on p. 100).

[6] A. Balatsoukas-Stimming, M. Meidlinger, R. Ghanaatian, G. Matz, and A. Burg, "A fully-unrolled LDPC decoder based on quantized message passing", in *Proc. SiPS 2015*, Hang Zhou, China, Oct. 2015 (cit. on pp. 6, 65, 78).

[7] T. Berger, *Rate Distortion Theory*. Eaglewood Cliffs, New Jersey: Prentice-Hall, 1971 (cit. on pp. 11, 36, 42, 49 sq.).

[8] C. Berrou and A. Glavieux, "Near optimum error correcting coding and decoding: Turbo-codes", *IEEE Trans. Information Theory*, vol. 44, no. 10, pp. 1261–1271, Oct. 1996 (cit. on p. 1).

[9] R. E. Blahut, "Computation of channel capacity and rate-distortion functions", *IEEE Trans. Information Theory*, vol. 18, no. 4, pp. 460–473, Jul. 1972 (cit. on p. 15).

[10] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge University Press, 2004 (cit. on p. 129).

[11] D. Burshtein, V. Della Pietra, D. Kanevsky, and A. Nadas, "Minimum impurity partitions", *The Annals of Statistics*, pp. 1637–1646, 1992 (cit. on p. 19).

[12] G. Caire, G. Taricco, and E. Biglieri, "Bit-interleaved coded modulation", *IEEE Trans. Information Theory*, vol. 44, no. 3, pp. 927–946, May 1998 (cit. on p. 2).

[13] J. Cardinal, "Compression of side information", in *Proc. IEEE Int. Conf. Multimedia, Expo (ICME)*, vol. 2, Baltimore, MD, USA, Jul. 2003, pp. 569–572 (cit. on p. 17).

[14] J. Cardinal, "Quantization with an information-theoretic distortion measure", Universite Libre de Bruxelles, Tech. Rep. 491, 2002 (cit. on p. 19).

[15] A. Charnes and W. W. Cooper, "Programming with linear fractional functionals", *Naval research logistics (NRL)*, vol. 9, no. 3-4, pp. 181–186, 1962 (cit. on p. 131).

[16] G. Chechik, A. Globerson, N. Tishby, and Y. Weiss, "Information bottleneck for Gaussian variables", *J. Machine Learning Res.*, vol. 6, no. 1, pp. 165–188, 2005 (cit. on pp. 2, 5, 21 sq.).

[17] J. Chen, A. Dholakia, E. Eleftheriou, M. Fossorier, and X.-Y. Hu, "Reduced-complexity decoding of LDPC codes", *IEEE Trans. Communications*, vol. 53, no. 8, pp. 1288–1299, Aug. 2005 (cit. on p. 3).

[18] S.-Y. Chung, J. Forney G.D., T. Richardson, and R. Urbanke, "On the design of low-density parity-check codes within 0.0045 dB of the Shannon limit", *IEEE Communications Letters*, vol. 5, no. 2, pp. 58–60, Feb. 2001 (cit. on pp. 1, 30, 69, 74, 128, 130, 133).

[19] S.-Y. Chung, T. Richardson, and R. Urbanke, "Analysis of sum-product decoding of low-density parity-check codes using a Gaussian approximation", *IEEE Trans. Information Theory*, vol. 47, no. 2, pp. 657–670, Feb. 2001 (cit. on pp. 30, 33, 120).

[20] T. Courtade and T. Weissman, "Multiterminal source coding under logarithmic loss", *IEEE Trans. Information Theory*, vol. 60, no. 1, pp. 740–761, Jan. 2014 (cit. on pp. 16 sq.).

[21] T. M. Cover and J. A. Thomas, *Elements of information theory*, 2nd ed. Hoboken, New Jersey: Wiley-Interscience, 2006 (cit. on pp. 12, 15, 39, 42, 49).

[22] I. Csiszár and J. Körner, *Information theory: coding theorems for discrete memoryless systems*, 2nd ed. Cambridge University Press, 2011 (cit. on p. 17).

[23] M. Danieli, S. Forchhammer, J. Andersen, L. Christensen, and S. Christensen, "Maximum mutual information vector quantization of log-likelihood ratios for memory efficient HARQ implementations", in *Proc. Data Compression Conference (DCC)*, Snowbird, UT, USA, Mar. 2010, pp. 30–39 (cit. on p. 12).

[24] M. C. Davey and D. MacKay, "Low-density parity check codes over GF(q)", *IEEE Communications Letters*, vol. 2, no. 6, pp. 165–167, 1998 (cit. on pp. 23, 141).

[25] D. Declercq, B. Vasic, S. Planjery, and E. Li, "Finite alphabet iterative decoders – Part II: Towards guaranteed error correction of LDPC codes via iterative decoder diversity", *IEEE Trans. Communications*, vol. 61, no. 10, pp. 4046–4057, Oct. 2013 (cit. on p. 3).

[26] Q. Du, V. Faber, and M. Gunzburger, "Centroidal voronoi tessellations: Applications and algorithms", *SIAM review*, vol. 41, no. 4, pp. 637–676, 1999 (cit. on p. 19).

[27] L. Duan, B. Rimoldi, and R. Urbanke, "Approaching the AWGN channel capacity without active shaping", in *Proc. IEEE International Symposium on Information Theory (ISIT)*, Ulm, Germany, Jun. 1997, p. 374 (cit. on p. 92).

[28] A. El Gamal and Y.-H. Kim, *Network information theory*. Cambridge University Press, 2011 (cit. on pp. 49, 51).

[29] "ETSI EN302 307: Second generation framing structure, channel coding and modulation systems for broadcasting, interactive services, news gathering and other broadband satellite applications", *DVB-S2 Std. V1.4.1*, Nov. 2014 (cit. on pp. 3, 83 sq.).

[30] A. G. Fàbregas, A. Martinez, and G. Caire, *Bit-interleaved coded modulation*. Now Publishers Inc, 2008 (cit. on p. 2).

[31] U. Fincke and M. Pohst, "Improved methods for calculating vectors of short length in a lattice, including a complexity analysis", *Mathematics of Computation*, vol. 44, no. 170, pp. 463–471, 1985 (cit. on p. 100).

[32] R. Gallager, "Low-density parity-check codes", *IEEE Trans. Information Theory*, vol. 8, no. 1, pp. 21–28, Jan. 1962 (cit. on pp. iii, 23, 85).

[33] R. Ghanaatian, A. Balatsoukas-Stimming, C. Muller, M. Meidlinger, G. Matz, A. Teman, and A. Burg, "A 588 Gbps LDPC decoder based on finite-alphabet message passing", *IEEE Trans. VLSI Systems*, vol. 26, no. 2, pp. 329–340, Feb. 2018 (cit. on pp. 6, 68, 78, 140).

[34] R. Gilad-Bachrach, A. Navot, and N. Tishby, "An information theoretic tradeoff between complexity and accuracy", in *Learning Theory and Kernel Machines*, Springer, 2003, pp. 595–609 (cit. on pp. 16 sq., 48 sq.).

[35] A. Globerson and N. Tishby, "On the optimality of the Gaussian information bottleneck curve", The Hebrew University of Jerusalem, Tech. Rep., Feb. 2004 (cit. on p. 22).

[36] S. Gordon, H. Greenspan, and J. Goldberger, "Applying the information bottleneck principle to unsupervised clustering of discrete and continuous image representations", in *Proc. 9th IEEE International Conference on Computer Vision*, Nice, France, Oct. 2003, pp. 370–377 (cit. on p. 2).

[37] F. Gray, *Pulse code communication*, U.S. Patent 2 632 058, Mar. 1947 (cit. on p. 91).

[38] R. M. Gray, *Entropy and information theory*. Springer, New York, 1990 (cit. on p. 49).

[39] ——, *Toeplitz and circulant matrices: A review*. Now Publishers Inc., 2006 (cit. on p. 47).

[40] R. M. Gray and D. L. Neuhoff, "Quantization", *IEEE Trans. Information Theory*, vol. 44, no. 6, pp. 2325–2383, Oct. 1998 (cit. on p. 11).

[41] U. Grenander and G. Szegö, *Toeplitz Forms and Their Applications*. Berkeley and Los Angeles: University of California Press, 1958 (cit. on p. 47).

[42] J. Hagenauer, E. Offer, and L. Papke, "Iterative decoding of binary block and convolutional codes", *IEEE Trans. Information Theory*, vol. 42, no. 2, pp. 429–445, Mar. 1996 (cit. on p. 66).

[43] T. Han, "Hypothesis testing with multiterminal data compression", *IEEE Trans. Information Theory*, vol. 33, no. 6, pp. 759–772, Nov. 1987 (cit. on p. 16).

[44] B. M. Hochwald and S. ten Brink, "Achieving near-capacity on a multiple-antenna channel", *IEEE Trans. Communications*, vol. 51, no. 3, pp. 389–399, Mar. 2003 (cit. on pp. 90, 100).

[45] W. Hoeffding, "Probability inequalities for sums of bounded random variables", *Journal of the American Statistical Association*, vol. 58, no. 301, pp. 13–10, 1963 (cit. on p. 108).

[46] P. A. Höher and T. Wo, "Superposition modulation: Myths and facts", *IEEE Communications Magazine*, vol. 49, no. 12, pp. 110–116, Dec. 2011 (cit. on pp. 4, 6, 85, 92 sqq., 100, 139).

[47] R. Horst and H. Tuy, *Global optimization: Deterministic approaches*. Springer, 1990 (cit. on p. 18).

[48] J. Hou, P. Siegel, L. Milstein, and H. Pfister, "Capacity-approaching bandwidth-efficient coded modulation schemes based on low-density parity-check codes", *IEEE Trans. Information Theory*, vol. 49, no. 9, pp. 2141–2155, Sep. 2003 (cit. on pp. 92, 126).

[49] X.-Y. Hu, E. Eleftheriou, and D. Arnold, "Regular and irregular progressive edge-growth tanner graphs", *IEEE Trans. Information Theory*, vol. 51, no. 1, pp. 386–398, Jan. 2005 (cit. on pp. 83, 141).

[50] "IEEE standard for air interface for broadband wireless access systems", *IEEE Std. 802.16-2012 (Revision of IEEE Std 802.16-2009)*, Aug. 2012 (cit. on p. 3).

[51] "IEEE standard for information technology – local and metropolitan area networks – specific requirements – part 11: Wireless LAN medium access control and physical layer specifications", *IEEE Std. 802.11n-2009*, Oct. 2009 (cit. on p. 3).

[52] "IEEE standard for information technology – local and metropolitan area networks – specific requirements – part 11: Wireless LAN medium access control and physical layer specifications – amendment 4: Enhancements for very high throughput for operation in bands below 6 GHz", *IEEE Std. 802.11ac-2013*, Dec. 2013 (cit. on p. 3).

[53] "IEEE standard for information technology – telecommunications and information exchange between systems – local and metropolitan area networks-specic requirements part 3: Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specications", *IEEE Std. 802.3an*, Sep. 2006 (cit. on pp. 3, 66, 78).

[54] T. Koch and A. Lapidoth, "At low SNR, asymmetric quantizers are better", *IEEE Trans. Information Theory*, vol. 59, no. 9, pp. 5421–5445, Sep. 2013 (cit. on p. 139).

[55] A. Kraskov, H. Stögbauer, and P. Grassberger, "Estimating mutual information", *Physical Review E*, vol. 69, no. 6, p. 066 138, 2004 (cit. on p. 53).

[56] F. R. Kschischang, B. J. Frey, and H. A. Loeliger, "Factor graphs and the sum-product algorithm", *IEEE Trans. Information Theory*, vol. 47, no. 2, pp. 498–519, Feb. 2001 (cit. on p. 1).

[57] B. M. Kurkoski, "On the relationship between the KL means algorithm and the information bottleneck method", in *Proc. 11th International ITG Conference on Systems, Communications and Coding*, Hamburg, Germany, Feb. 2017, pp. 1–6 (cit. on p. 19).

[58] B. M. Kurkoski and H. Yagi, "Quantization of binary-input discrete memoryless channels", *IEEE Trans. Information Theory*, vol. 60, no. 8, pp. 4544–4552, Aug. 2014 (cit. on pp. 6, 20 sq., 64).

[59] B. M. Kurkoski, K. Yamaguchi, and K. Kobayashi, "Noise thresholds for discrete LDPC decoding mappings", in *Proc. IEEE GLOBECOM*, New Orleans, LA, USA, Nov. 2008, pp. 1–5 (cit. on pp. 3, 56, 59, 64, 68).

[60] J. Lewandowsky and G. Bauch, "Trellis based node operations for LDPC decoders from the information bottleneck method", in *Proc. Int. Conf. Signal Process. Comm. Systems*, Cairns, Australia, Dec. 2015, pp. 1–10 (cit. on pp. 3, 12, 19, 21, 56, 68, 137).

[61] J. Lewandowsky, M. Stark, and G. Bauch, "Optimum message mapping LDPC decoders derived from the sum-product algorithm", in *Proc. IEEE ICC 2016*, Kuala Lumpur, Malaysia, May 2016, pp. 1–6 (cit. on p. 3).

[62] M. Luby, M. Mitzenmacher, M. Shokrollahi, and D. Spielman, "Improved low-density parity-check codes using irregular graphs", *IEEE Trans. Information Theory*, vol. 47, no. 2, pp. 585–598, Feb. 2001 (cit. on pp. 23, 27, 103).

[63] M. G. Luby, M. Mitzenmacher, M. A. Shokrollahi, D. A. Spielman, and V. Stemann, "Practical loss-resilient codes", in *Proc. 29th annual ACM symposium on Theory of computing*, El Paso, TX, USA, 1997, pp. 150–159 (cit. on pp. 1, 3, 23 sq.).

[64] X. Ma and L. Ping, "Coded modulation using superimposed binary codes", *IEEE Trans. Information Theory*, vol. 50, no. 12, pp. 3331–3343, Dec. 2004 (cit. on pp. 6, 85, 92, 100, 102, 139).

[65] D. J. C. MacKay, "Good error-correcting codes based on very sparse matrices", *IEEE Trans. Information Theory*, vol. 45, no. 2, pp. 399–431, Mar. 1999 (cit. on pp. 85, 135).

[66] D. J. MacKay and R. M. Neal, "Good codes based on very sparse matrices", in *Cryptography and Coding 5th IMA Conf.* C. Boyd, Ed. Berlin, Germany: Springer Berlin Heidelberg, 1995, pp. 100–111 (cit. on p. 23).

[67] M. Meidlinger, A. Balatsoukas-Stimming, A. Burg, and G. Matz, "Quantized message passing for LDPC codes", in *Proc. 49th Asilomar Conf. Signals, Systems and Computers*, Pacific Grove, CA, USA, Nov. 2015 (cit. on pp. 6, 12, 65).

[68] M. Meidlinger and G. Matz, "On irregular LDPC codes with quantized message passing decoding", in *Proc. IEEE Signal Processing Advances in Wireless Communications (SPAWC)*, Sapporo, Japan, Jul. 2017, pp. 731–735 (cit. on pp. 6, 12).

[69] ——, "Quantized symmetric message passing for irregular LDPC codes", *submitted to IEEE Trans. Communications*, 2018 (cit. on p. 6).

[70] M. Meidlinger, A. Winkelbauer, and G. Matz, "On the relation between the Gaussian information bottleneck and MSE-optimal rate-distortion quantization", in *Proc. IEEE Workshop on Statistical Signal Processing (SSP)*, Gold Coast, Australia, Jun. 2014, pp. 89–92 (cit. on p. 5).

[71] R. Motwani and P. Raghavan, *Randomized algorithms*. Cambridge University Press, 1995 (cit. on p. 108).

[72] G. Pichler and G. Koliander, "Information bottleneck on general alphabets", submitted to IEEE International Symposium on Information Theory (ISIT), available online at: `https://arxiv.org/abs/1801.01050`, 2018 (cit. on p. 49).

[73] G. Pichler, P. Piantanida, and G. Matz, "Distributed information-theoretic clustering", submitted to IEEE Trans. Information Theory, available online at: `https://arxiv.org/abs/1602.04605`, 2018 (cit. on p. 17).

[74] S. Planjery, D. Declercq, L. Danjean, and B. Vasic, "Finite alphabet iterative decoders – Part I: Decoding beyond belief propagation on the binary symmetric channel", *IEEE Trans. Communications*, vol. 61, no. 10, pp. 4033–4045, Oct. 2013 (cit. on pp. 56, 140).

[75] T. Richardson and R. Urbanke, "Multi-edge type LDPC codes", Available: `http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.106.7310&rep=rep1&type=pdf`, 2002 (cit. on p. 24).

[76] ——, "Multi-edge type LDPC codes", in *Proc. Workshop Honoring Prof. Bob McEliece 60th Birthday, California Inst. Technol.*, Pasadena, CA, USA, 2002, pp. 24–25 (cit. on pp. 24, 141).

[77] T. Richardson, M. Shokrollahi, and R. Urbanke, "Design of capacity-approaching irregular low-density parity-check codes", *IEEE Trans. Information Theory*, vol. 47, no. 2, pp. 619–637, Feb. 2001 (cit. on pp. 1, 3, 6, 23, 30, 33, 69, 71 sqq., 79, 85, 118 sq., 128, 130, 133, 140 sq.).

[78] T. Richardson and R. Urbanke, "The capacity of low-density parity-check codes under message-passing decoding", *IEEE Trans. Information Theory*, vol. 47, no. 2, pp. 599–618, Feb. 2001 (cit. on pp. 7, 27 sqq., 55, 102 sqq.).

[79] T. Richardson and R. Urbanke, *Modern coding theory.* Cambridge University Press, 2008 (cit. on pp. iii, 24, 30, 128).

[80] F. J. C. Romero and B. M. Kurkoski, "Decoding LDPC codes with mutual information-maximizing lookup tables", in *Proc. IEEE International Symposium on Information Theory (ISIT)*, Hong Kong, China, Jun. 2015, pp. 426–430 (cit. on pp. 3, 56, 68).

[81] ——, "LDPC decoding mappings that maximize mutual information", *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 9, pp. 2391–2401, Sep. 2016 (cit. on pp. 3, 12, 56).

[82] D. Sakrison, "Source encoding in the presence of random disturbance", *IEEE Trans. Information Theory*, vol. 14, no. 1, pp. 165–167, Jan. 1968 (cit. on pp. 5, 43, 45 sq., 52).

[83] C. Schlegel, M. Burnashev, and D. Truhachev, "Generalized superposition modulation and iterative demodulation: A capacity investigation", *Journal of Electrical and Computer Engineering*, vol. 2010, p. 1, 2010 (cit. on p. 92).

[84] E. Schneidman, N. Slonim, N. Tishby, R. deRuyter van Steveninck, and W. Bialek, "Analyzing neural codes using the information bottleneck method", in *Advances in Neural Information Processing Systems (NIPS) 13*, MIT Press, 2002 (cit. on p. 2).

[85] C.-P. Schnorr and M. Euchner, "Lattice basis reduction: Improved practical algorithms and solving subset sum problems", *Mathematical programming*, vol. 66, no. 1-3, pp. 181–199, 1994 (cit. on p. 100).

[86] C. E. Shannon, "Coding theorems for a discrete source with a fidelity criterion", *IRE Nat. Conv. Rec.*, vol. 4, pp. 142–163, Mar. 1959 (cit. on pp. 2, 11, 13).

[87] C. E. Shannon, "A mathematical theory of communication", *The Bell System Technical Journal*, vol. 27, pp. 379–423, 623–656, 1948 (cit. on pp. 1, 4, 51, 85).

[88] E. Sharon, A. Ashikhmin, and S. Litsyn, "EXIT functions for the Gaussian channel", in *Proc. 40th Allerton Conf. Communication, Control, and Computing*, vol. 41, Monticello, IL, USA, Oct. 2003, pp. 972–981 (cit. on p. 33).

[89] A. Shokrollahi and R. Storn, "Design of efficient erasure codes with differential evolution", in *Proc. IEEE International Symposium on Information Theory (ISIT)*, Sorrento, Italy, Jun. 2000, p. 5 (cit. on p. 133).

[90] J. Singh, O. Dabeer, and U. Madhow, "On the limits of communication with low-precision analog-to-digital conversion at the receiver", *IEEE Trans. Communications*, vol. 57, no. 12, pp. 3629–3639, Dec. 2009 (cit. on pp. 51, 139).

[91] N. Slonim and N. Tishby, "Agglomerative information bottleneck", in *Advances in Neural Information Processing Systems (NIPS) 12*, S. A. Solla, T. K. Leen, and K. Müller, Eds., MIT Press, 2000, pp. 617–623 (cit. on p. 19).

[92] ——, “Document clustering using word clusters via the information bottleneck method”, in *Proc. 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Athens, Greece, 2000, pp. 208–215 (cit. on p. 2).

[93] B. Smith, F. R. Kschischang, and W. Yu, “Low-density parity-check codes for discretized min-sum decoding”, in *Proc. 23rd Biennial Symposium on Communications*, Kingston, ON, Canada, May 2006, pp. 14–17 (cit. on pp. 71, 82).

[94] R. Storn and K. Price, “Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces”, *Journal of global optimization*, vol. 11, no. 4, pp. 341–359, 1997 (cit. on p. 133).

[95] C. Studer, A. Burg, and H. Bolcskei, “Soft-output sphere decoding: Algorithms and VLSI implementation”, *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 2, pp. 290–300, 2008 (cit. on p. 100).

[96] E. Telatar, “Capacity of multi-antenna Gaussian channels”, *European transactions on telecommunications*, vol. 10, no. 6, pp. 585–595, 1999 (cit. on p. 95).

[97] S. ten Brink, “Designing iterative decoding schemes with the extrinsic information transfer chart”, *AEU Int. J. Electron. Commun*, vol. 54, no. 6, pp. 389–398, 2000 (cit. on pp. 31 sq.).

[98] S. ten Brink, G. Kramer, and A. Ashikhmin, “Design of low-density parity-check codes for modulation and detection”, *IEEE Trans. Communications*, vol. 52, no. 4, pp. 670–678, Apr. 2004 (cit. on pp. 31, 33, 92, 126, 141).

[99] N. Tishby, F. C. Pereira, and W. Bialek, “The information bottleneck method”, in *Proc. 37th Allerton Conf. Communication, Control, and Computing*, Monticello, IL, USA, Sep. 1999, pp. 368–377 (cit. on pp. 2, 12, 14 sqq.).

[100] J. Tong, L. Ping, and X. Ma, “Superposition coded modulation with peak-power limitation”, *IEEE Trans. Information Theory*, vol. 55, no. 6, pp. 2562–2576, Jun. 2009 (cit. on p. 92).

[101] D. Tse and P. Viswanath, *Fundamentals of wireless communication*. Cambridge University Press, 2005 (cit. on p. 90).

[102] M. Tüchler and J. Hagenauer, “EXIT charts of irregular codes”, in *Proc. 2002 Conf. Information Sciences and Systems*, Princeton, NJ, Mar. 2002 (cit. on pp. 7, 33, 119).

[103] M. Tüchler, A. C. Singer, and R. Koetter, “Minimum mean squared error equalization using a priori information”, *IEEE Trans. Signal Processing*, vol. 50, no. 3, pp. 673–683, 2002 (cit. on pp. 7, 101).

[104] G. Ungerboeck, “Channel coding with multilevel/phase signals”, *IEEE Trans. Information Theory*, vol. 28, no. 1, pp. 55–67, Jan. 1982 (cit. on p. 92).

[105] S. Verdú, “On channel capacity per unit cost”, *IEEE Trans. Information Theory*, vol. 36, no. 5, pp. 1019–1030, Sep. 1990 (cit. on p. 17).

[106] J. Wang, T. Courtade, H. Shankar, and R. D. Wesel, “Soft information for LDPC decoding in flash: Mutual-information optimized quantization”, in *Proc. IEEE GLOBECOM*, Houston, TX, USA, Dec. 2011, pp. 1–6 (cit. on p. 3).

[107] M. B. Westover and J. A. O’Sullivan, “Achievable rates for pattern recognition”, *IEEE Trans. Information Theory*, vol. 54, no. 1, pp. 299–320, Jan. 2008 (cit. on p. 16).

[108] N. Wiener, *Extrapolation, Interpolation, and Smoothing of Stationary Time Series*. New York, NY, USA: Wiley, 1949 (cit. on pp. 5, 35 sq.).

[109] A. Winkelbauer and G. Matz, “On quantization of log-likelihood ratios for maximum mutual information”, in *Proc. IEEE Signal Processing Advances in Wireless Communications (SPAWC)*, Stockholm, Sweden, Jun. 2015, pp. 316–320 (cit. on pp. 12, 21, 62).

[110] A. Winkelbauer, G. Matz, and A. Burg, “Channel-optimized vector quantization with mutual information as fidelity criterion”, in *Proc. 47th Asilomar Conf. Signals, Systems and Computers*, Pacific Grove, CA, USA, Nov. 2013, pp. 851–855 (cit. on pp. 2, 12).

[111] A. Winkelbauer, “Blind performance estimation and quantizer design with applications to relay networks”, PhD thesis, Technische Universität Wien, 2014 (cit. on p. 48).

[112] A. Winkelbauer, S. Farthofer, and G. Matz, “The rate-information trade-off for Gaussian vector channels”, in *Proc. IEEE International Symposium on Information Theory (ISIT)*, Honolulu, HI, USA, Jun. 2014 (cit. on p. 49).

[113] T. Wo and P. A. Höher, “Superposition mapping with application in bit-interleaved coded modulation”, in *Proc. International ITG Conference on Source and Channel Coding*, Siegen, Germany, Jan. 2010, pp. 1–6 (cit. on p. 4).

[114] T. Wo, “Superposition mapping and related coding techniques”, PhD thesis, Christian-Albrechts-Universität Kiel, Kiel, 2011 (cit. on pp. 4, 7, 85, 92 sq., 123, 125).

[115] T. Wo and P. A. Höher, “A universal coding approach for superposition mapping”, in *Proc. 6th International Symposium on Turbo Codes and Iterative Information Processing (ISTC)*, Brest, France, Sep. 2010, pp. 314–318 (cit. on pp. 4, 7, 85, 92 sq., 123, 135).

[116] T. Wo, M. Noemm, D. Hao, and P. A. Höher, “Iterative processing for superposition mapping”, *Journal of Electrical and Computer Engineering*, vol. 2010, p. 6, 2010 (cit. on pp. 92, 95, 102).

[117] H. Wymeersch, *Iterative receiver design*. Cambridge University Press, 2007 (cit. on p. 1).

[118] A. D. Wyner, “On source coding with side information at the decoder”, *IEEE Trans. Information Theory*, vol. 21, no. 3, pp. 294–300, May 1975 (cit. on p. 17).

[119] G. Zeitler, “Low-precision analog-to-digital conversion and mutual information in channels with memory”, in *Proc. 48th Allerton Conf. Communication, Control, and Computing*, Monticello, IL, USA, Sep. 2010, pp. 745–752 (cit. on p. 2).

[120] G. Zeitler, A. Singer, and G. Kramer, “Low-precision A/D conversion for maximum information rate in channels with memory”, *IEEE Trans. Communications*, vol. 60, no. 9, pp. 2511–2521, Sep. 2012 (cit. on p. 12).

[121] L. M. Zhang and F. R. Kschischang, “Multi-edge-type low-density parity-check codes for bandwidth-efficient modulation”, *IEEE Trans. Communications*, vol. 61, no. 1, pp. 43–52, Jan. 2013 (cit. on pp. 92, 126, 129 sq.).

[122] T. Zhang, Z. Wang, and K. Parhi, “On finite precision implementation of low density parity check codes decoder”, in *Proc. IEEE International Symposium on Circuits and Systems (ISCAS)*, Sydney, Australia, May 2001, pp. 202–205 (cit. on p. 3).

[123] X. Zhang and P. Siegel, “Quantized min-sum decoders with low error floor for LDPC codes”, in *Proc. IEEE International Symposium on Information Theory (ISIT)*, Cambridge, MA, USA, Jul. 2012, pp. 2871–2875 (cit. on pp. 3, 140).

[124] ——, “Quantized iterative message passing decoders with low error floor for LDPC codes”, *IEEE Trans. on Communications*, vol. 62, no. 1, pp. 1–14, Jan. 2014 (cit. on p. 140).

[125] Z. Zhang, L. Dolecek, B. Nikolic, V. Anantharam, and M. Wainwright, “Design of LDPC decoders for improved low error rate performance”, *IEEE Trans. Communications*, vol. 57, no. 11, pp. 3258–3268, Nov. 2009 (cit. on p. 3).

[126] J. Zhao, F. Zarkeshvari, and A. H. Banihashemi, “On implementation of min-sum algorithm and its modifications for decoding low-density parity-check (LDPC) codes”, *IEEE Trans. Communications*, vol. 53, no. 4, pp. 549–554, Apr. 2005 (cit. on p. 3).

[127] K. Zhao, W. Zhao, H. Sun, X. Zhang, N. Zheng, and T. Zhang, “LDPC-in-SSD: Making advanced error correction codes work effectively in solid state drives”, in *Proc. 11th USENIX FAST*, 2013, pp. 243–256 (cit. on p. 3).

[128] S. Zhao, X. Ma, and B. Bai, “Decoding algorithms of LDPC coded superposition modulation”, *IEEE Communications Letters*, vol. 18, no. 3, pp. 487–490, Mar. 2014 (cit. on p. 92).