DISSERTATIONES
INFORMATICAE
UNIVERSITATIS
TARTUENSIS
72

SANDER MIKELSAAR

Analysis and Optimization of Iteratively Decodable Codes

DISSERTATIONES INFORMATICAE UNIVERSITATIS TARTUENSIS
72

SANDER MIKELSAAR

Analysis and Optimization of Iteratively Decodable Codes

UNIVERSITY OF TARTU
Press
1632

Institute of Computer Science, Faculty of Science and Technology, University of Tartu, Estonia.

Dissertation has been accepted for the commencement of the degree of Doctor of Philosophy (PhD) in Computer Science on September 30, 2025 by the Council of the Institute of Computer Science, University of Tartu.

Supervisors

Assoc. Prof.	Vitaly Skachek University of Tartu, Estonia
Prof.	Boris Kudryashov University of Tartu, Estonia
Assoc. Prof.	Irina Bocharova University of Tartu, Estonia

Opponents

Prof.	Emmanuel Boutillon Université Bretagne Sud, France
Assoc. Prof.	Michael Lentmaier Lund University, Sweden

The public defense will take place on November 21, 2025 at 11:00 in Narva Rd. 18-1021.

The publication of this dissertation was financed by the Institute of Computer Science, University of Tartu.

ISSN 2613-5906 (print) ISSN 2806-2345 (pdf)
ISBN 978-9908-57-035-8 (print) ISBN 978-9908-57-036-5 (pdf)

University of Tartu Press
http://www.tyk.ee

ABSTRACT

Reliable digital communication is fundamental for ensuring an error-free exchange of information between devices. The design of practical communication systems requires optimization of state-of-the-art error-correcting code constructions and methods of physical transmission of data signals. A significant challenge in designing such systems is a restriction on computational complexity for the methods used, especially on mobile devices, for which added complexity directly translates to increased drain on battery life. While coding theory, and codes for communication, are well-established fields of research, the area of digital communications has seen rapid developments in recent history, such as the introduction of 5G telecommunication networks. The focus of this dissertation is to study and develop parts of these technologies for potential future – beyond 5G – communications standards. The work covered can be broadly categorized in two: codes and modulation.

Firstly, in order to achieve robust, powerful error correction required for communication over noisy channels, this dissertation focuses on multiple classes of low-density parity-check codes. This class of codes has already found widespread adoption in modern standards, including 5G, as it achieves excellent error-correction while satisfying complexity requirements. To achieve improved decoding performance of the class of codes compared to existing standard, methods proposed and studied in the dissertation cover all steps involved in the code design process, including generalized and non-binary variants of low-density parity-check codes. Methods of estimating decoding performance as finite length random coding bounds are studied for the two variants.

Secondly, for modulation, joint optimization of powerful error-correcting codes and coded modulation, including shaping, is required in order to approach the theoretical limits of reliable communication over noisy channels. Focusing on non-binary codes, suboptimal approaches of matching code symbols to channel signals will result in a significant performance gap in achievable decoding error rates for the communication systems. By analysis of the relation between the code Hamming and Euclidean distance spectra, an approach for estimation of decoding error probabilities for coded shaped signaling is proposed and used for optimization of multiple schemes of shaped coded modulation.

The resulting bounds and designed codes are visualized and compared to simulation results, including comparisons to existing standards and other well-established methods.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

ACE approximate cycle EMD
ASCM amplitude-sign coded modulation
ASK amplitude shift keying
AWGN additive white Gaussian noise

BCH Bose–Chaudhuri–Hocquenghem
BCJR Bahl-Cocke-Jelinek-Raviv
BER bit error rate
BICM bit-interleaved coded modulation
BP belief propagation
BPCM bit-plane coded modulation
BPSK binary phase-shift keying
BSC binary symmetric channel
BSS binary symmetric source

CCDM constant composition distribution matching
CE constellation extension
CM coded modulation

DE density evolution
DMC discrete memoryless channel

ECC error-correcting code
EMD extrinsic message degree
ESS enumerative sphere shaping
EXIT extrinsic information transfer

FER frame error rate
FHT fast Hadamard transform

GCS geometric constellation shaping
GLDPC generalized LDPC

LDPC low-density parity-check
LLR logarithmic likelihood ratio
LSR linear shift register
LTE long term evolution

M2O many-to-one

MAP maximum a posteriori probability
MB Maxwell-Boltzmann
MGF moment generating function
ML maximum likelihood
MPDM multiset partition distribution matching
MSF minimal span form

NB non-binary
NB QC-LDPC non-binary quasi-cyclic LDPC

PAM pulse amplitude modulation
PAS probabilistic amplitude shaping
PCS probabilistic constellation shaping
PDF probability density function
PEG progressive edge growth
PMF probability mass function

QAM quadrature amplitude modulation
QC quasi-cyclic
QC-LDPC quasi-cyclic LDPC

RS Reed-Solomon

SA simulated annealing
SAC shaping after coding
SBC shaping before coding
SED squared Euclidean distance
SEDS SED spectrum
SICM symbol-interleaved coded modulation
SISO soft-input soft-output
SNR signal-to-noise ratio
SPA sum-product algorithm

TB tail biting
TS tangential-sphere

WAVA wrap-around Viterbi algorithm

NOMENCLATURE

C channel capacity
E_s average signal energy
G generator matrix
$I(X;Y)$ mutual information between X and Y
I_k $k \times k$ identity matrix
J row weight of a regular LDPC code
K column weight of a regular LDPC code
M modulation order
N dimension of a vector space
$Q(x)$ complementary Gaussian distribution function
R code rate
R_T transmission rate
$\mathcal{V}$ set of vertices
$\mathcal{G}$ graph
$\hat{M}$ lifting degree
$\mathbf{c}$ codeword
$\mathbf{e}$ error vector
$\mathbf{m}$ message vector
$\mathbf{p}$ probability distribution
$\mathbf{y}$ channel output vector
$\mathcal{C}$ code
$\mathcal{E}$ set of edges
$\mathcal{M}$ modulation alphabet
$\mathcal{P}$ permutation matrix
$\mathcal{S}$ trellis layer
$\mathcal{T}$ trellis
$\mathring{\mathbf{c}}$ cycle
μ_s state complexity
π interleaver
d_H Hamming distance
d_{min} minimum distance
e edge
g girth
k code dimension
n code length
w weight

LIST OF ORIGINAL PUBLICATIONS

Publications included in the thesis

The names of the authors of publications are listed in alphabetical order.

I Irina E. Bocharova, Boris D. Kudryashov, and Sander Mikelsaar. "Irregular generalized LDPC codes in practical communication scenarios". In: *Proceedings of IEEE Information Theory Workshop (ITW)*. 2022, pp. 517–522. DOI: 10.1109/ITW54588.2022.9965833.

The author proposed and developed a new optimization algorithm for labeling base matrices of generalized LDPC codes by columns of parity-check matrices of constituent codes, which was the main contribution and focus of the paper. Generalized LDPC code design, simulations and derivations of bounds were performed in collaboration with co-authors.

II Irina E. Bocharova, Boris D. Kudryashov, Sander Mikelsaar, and Vitaly Skachek. "Bound on the ML decoding error probability for coded QAM signals with shaping". In: *Proceedings of IEEE International Symposium on Information Theory (ISIT)*. 2023, pp. 270–275. DOI: 10.1109/ISIT54713.2023.10206612.

The author worked in collaboration with the co-authors and supervisors, with contributions in NB LDPC code and shaping book design, simulations and derivations of bounds.

III Irina E. Bocharova, Boris D. Kudryashov, Sander Mikelsaar, and Vitaly Skachek. "Shaping for NB QC-LDPC coded QAM signals". In: *12th International Symposium on Topics in Coding (ISTC)*. 2023, pp. 1–5. DOI: 10.1109/ISTC57237.2023.10273467.

The author proposed the new mapping and shaping optimization criterion and algorithm, which was an integral part of the paper. The algorithm allows for better matching of short and moderate length NB LDPC code symbols to a set of shaped M-PAM signals. The optimized coded modulation spectra and bounds on decoding performance for the same codes used with optimized mappings were derived in collaboration with the co-authors.

IV Irina E. Bocharova, Boris D. Kudryashov, and Sander Mikelsaar. "Analysis of coded shaped QAM signaling at short and moderate lengths". In: *Proceedings of IEEE International Symposium on Information Theory (ISIT)*. 2024, pp. 641–646. DOI: 10.1109/ISIT57864.2024.10619186.

The author proposed and developed the new two-step shaping book optimization algorithm, which was a core contribution of the paper, involving AI techniques, combining a genetic algorithm with improved saddlepoint approximation-based optimization.

1. INTRODUCTION

Digital communication has become an indispensable part of the modern age. Nowadays, it is hard to imagine a world where we would not able to instantly communicate with someone on the other side of the planet, receive any desired piece of information or stream ultra-high definition video whenever we feel like it. While the advancements in the field of digital communication that enable these technologies have been monumental, the requirements for communications systems in the evolving information age will continue to grow. Achieving reliable communication that meets the data throughput requirements of future standards, going beyond technologies like 5G, while satisfying practical constraints like power efficiency, is a multi-faceted challenge that requires joint optimization of powerful error-correcting codes and the signaling used to transmit the encoded data, while also minimizing the complexity of all processes involved.

Low-density parity-check (LDPC) codes have found widespread adoption in modern systems. Due to achieving error correcting performance near the theoretical limit while preserving low complexity of decoding, this class of codes is an appealing choice to satisfy the demanding requirements of the future. Thus, the focus of this dissertation is on the design and optimization of different types of LDPC codes, combined optimization of both shaped and unshaped coded modulation.

Following this chapter, where an introduction to the processes involved in a digital communication system, basics of coding theory and signal processing are given, LDPC codes will be covered in detail in the chapters that follow, starting from binary variants and their decoding, moving on to generalized and non-binary LDPC codes. Finally, the topics of coded modulation and shaping are covered in the last chapters.

1.1. Basic Communication Model

To reliably communicate over a noisy channel, sent information needs to be transformed to allow for correction of errors introduced to communicated messages while in transmission. A basic model of a digital communication system in Fig.1 can be separated into distinct processes.

A data source produces either analog or discrete information sequences, or messages, which are compressed by the source encoder into binary data. The output of the source encoder is considered to be a binary symmetric source (BSS), the produced messages $\mathbf{m}$ are sequences of independently distributed equiprobable binary symbols. As a result of source encoding, redundancy in real-world information is reduced by compression, using the minimum amount of bits possible. Correspondingly, source decoding on the destination side decompresses the messages $\mathbf{m}'$ back to their original form. These two

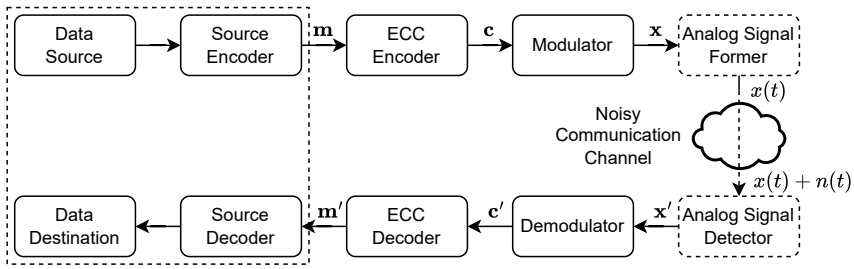

Figure 1. Digital communication system

operations are considered to lie outside the scope of this dissertation, and thus simply considered to be parts of the source and receiver.

As errors in transmission over any channel are practically unavoidable, a sent message $\mathbf{m}$ has to first be transformed to allow for correction of erroneous received sequences $\mathbf{x}'$ by the receiver. Error-correcting coding, or simply *encoding*, produces codewords $\mathbf{c}$ by adding redundancy to messages, typically in a systematic form.

The actual process of information transfer must take place over some medium, called a *communication channel*. These channels can be either physical transmission lines - e.g. coaxial or fiber-optic cables, or broadcast, for example radio waves used for 5G telecommunications. The channels are represented by theoretical *channel models*, used to study and design different communication systems.

To send digital information over a channel, a transformation from discrete signals to an analog carrier signal must be performed. The process of transforming one (or more) properties of the carrier wave to allow information transfer is known as *modulation*. While the actual transformation to and from analog signals can be treated as a separate processes, in this dissertation, they will simply be considered as parts of the modulation and demodulation processes.

On the receiver side of the communication model, the received signals must first be transformed back from their analog waveforms received from the channel, to digital sequences. Known as *demodulation*, the possibly quantized sequences of noisy measured values are output as the received sequences $\mathbf{c}'$.

As the final stage in the model, the *decoder* is used to output an information sequence $\mathbf{m}'$ based on the received sequence $\mathbf{c}'$ and knowledge of the form in which redundancy was added to the message in the encoder. Information transfer is considered successful if the messages $\mathbf{m}$ and $\mathbf{m}'$ of the sender and receiver coincide, meaning that any errors that might have occurred in transmission have successfully been corrected.

In any realistic scenario, errors will still occur. The goal of designing reliable digital communication systems can be simplified to the minimization the expected probability of an unsuccessful transfer through the channel, expressed as

$$P_e(\text{SNR}, R_T) = \Pr(\mathbf{m}' \neq \mathbf{m}),$$

where signal-to-noise ratio (SNR) is used as an indicator or signal strength in relation to the noise in the communication channel and the *data transmission rate* R_T is used to express the "density" of information transfer, determined by the code rate of the error-correcting code and choice of channel modulation, measured as bits per channel use. In other words, the goal is to design a communication system in such a way that, for a given SNR and R_T, the expected probability of successful information transfer – it's reliability – is maximized.

1.2. Error-Correcting Codes and Examples

Starting with an introduction to error-correcting codes (ECCs), necessary definitions are covered, while omitting much of the mathematical background. A more detailed coverage of the background, including proofs of claims made in this section, is provided in [LC83].

Definition 1.2.1 (Binary field)**.** A field GF(2) (or $\mathbb{F}_2$), consisting of two elements $\{0, 1\}$, is called a *binary field* with two operations, addition ($+$, or $\oplus$ (XOR)) and multiplication ($\times$), for two field elements $x, y \in \text{GF}(2)$ defined as:

$+$	0	1
0	0	1
1	1	0

$\times$	0	1
0	0	0
1	0	1

Definition 1.2.2 (Prime field)**.** A field GF(p), where p is a prime, is called a *prime field*, or a *residue field* modulo p, consisting of p elements: $\text{GF}(p) = \{0, 1, \ldots, p-1\}$.

Definition 1.2.3 (Binary extension field)**.** The field

$$\text{GF}(2^m) = \{f_{m-1}x^{m-1} + \ldots + f_1 x + f_0, f_i \in \text{GF}(2)\},$$

with a irreducible polynomial $p(x)$, where additions and multiplications of polynomials are performed by modulo $p(x)$, is called a *binary extension field*.

Definition 1.2.4 (Linear vector space)**.** A set of vectors closed with respect to addition and multiplication is called a *linear vector space*, with addition defined as $\mathbf{x} + \mathbf{y} = (x_1 + y_1, \ldots, x_n + y_n)$ for vectors $\mathbf{x} = (x_1, \ldots, x_n), \mathbf{y} = (y_1, \ldots, y_n)$, where $x_i, y_i \in \text{GF}(2)$, and multiplication of vector $\mathbf{x}$ with a scalar $\alpha \in \text{GF}(2)$ defined as $\alpha\mathbf{x} = (\alpha x_1, \ldots, \alpha x_n)$. A length-$n$ binary vector space is denoted as $\text{GF}(2)^n$.

Definition 1.2.5 (Linear code)**.** A *linear* $[n, k]$ *code* $\mathcal{C}$ is a k-dimensional subspace of a n-dimensional vector space $\mathrm{GF}(2)^n$.

A linear code $\mathcal{C}$ is, in other words, simply a set of 2^k binary codewords of length n satisfying the linearity constraint. Thus, a set of k vectors $\mathbf{g}_i, i = 1, \ldots, k$ can be chosen to form the basis of the code $\mathcal{C}$, such that any of the 2^k codewords of $\mathcal{C}$ are obtainable as a linear combination of the basis vectors:

$$\mathbf{c} = u_1\mathbf{g}_1 + u_2\mathbf{g}_2 + ... + u_k\mathbf{g}_k,$$

where $\mathbf{u} = (u_1, u_2, ..., u_k) \in \mathrm{GF}(2)^k$ is the binary information sequence to be encoded.

Definition 1.2.6 (Generator matrix)**.** A generator matrix of a linear $[n, k]$ code $\mathcal{C}$ is an $k \times n$ matrix G, whose rows are basis vectors of $\mathcal{C}$.

Encoding of a message can then be achieved by multiplication: $\forall \mathbf{u} \in \mathrm{GF}(2)^k : \mathbf{u}G = \mathbf{c} \in \mathcal{C}$. As row operations on G do not alter the basis vectors $\mathbf{g}_i$, a generator matrix G' obtained as a result of performing linear combinations on the rows of G is still a valid encoder for $\mathcal{C}$, simply with an alternative mapping of $\mathbf{u} \to \mathbf{c}$. Column operations, however, do change the resulting codewords, thus resulting in an equivalent code. By performing linear row operations and column swapping, an equivalent code in the systematic form $G_{sys} = (I_k|P)$ can be obtained for any generator matrix G. In this form, the $k \times k$ identity matrix I_k results in systematic codewords in which the first k bits coincide with the uncoded binary information sequence

$$\mathbf{c}_{sys} = (u_1, u_2, ..., u_k, c_{k+1}, ..., c_n) \in \mathcal{C}_{sys}.$$

The $k \times (n - k)$ submatrix P of G_{sys}, encoding $(c_{k+1}, ..., c_n)$ is known as the parity block.

Definition 1.2.7 (Parity-check matrix)**.** A basis for the dual space of an $[n, k]$ code $\mathcal{C}$, placed in a matrix, is called parity-check matrix.

A parity-check matrix H of $\mathcal{C}$ is an $(n - k) \times n$ full-rank matrix, such that $GH^\top = \mathbf{0}$, where $\mathbf{0}$ denotes the $k \times (n - k)$ all-zero matrix. For any codeword $\mathbf{c} \in \mathcal{C}$, $\mathbf{c}H^\top = \mathbf{0}$. In other words, any codeword $\mathbf{c}$ of $\mathcal{C}$ must satisfy all $(n-k)$ parity-checks determined by the parity-check matrix H. A parity-check matrix corresponding to the systematic matrix G_{sys} is obtained as $H_{sys} = \left(-P^\top|I_{n-k}\right)$.

The parity-check matrix of a code can be used to detect errors in encoded messages received from a noisy channel or for decoding of received messages, as an example, using syndrome decoding in the binary symmetric channel (BSC). By representing a received binary message as $\mathbf{r} = \mathbf{c} + \mathbf{e}$, for example, obtained from a BSC channel (Sec. 1.4.1), as the resulting sum of a binary error vector $\mathbf{e}$ and a transmitted codeword $\mathbf{c}$, H can be used to find the syndrome vector $\mathbf{s} = \mathbf{r}H^\top$. As $\mathbf{c}H^\top = \mathbf{0}$ must hold, $\mathbf{r}H^\top = (\mathbf{c} + \mathbf{e})H^\top =$

$\mathbf{0} + \mathbf{e}H^\top = \mathbf{e}H^\top$, and thus, if $\mathbf{s} \neq \mathbf{0}$, at least one error must have occurred during transmission over the BSC.

Definition 1.2.8 (Hamming weight)**.** For a vector $\mathbf{x}$, its *Hamming weight* $w_H(\mathbf{x})$ is defined as the number of non-zero symbols in the vector.

Definition 1.2.9 (Hamming distance)**.** *Hamming distance* d_H is used to measure the number of positions in which two nonequal binary vectors $\mathbf{x}, \mathbf{y}$ of length n differ from each other:

$$d_H(\mathbf{x}, \mathbf{y}) = w_H(\mathbf{x} \oplus \mathbf{y}) = \sum_{i=1}^{n} w_H(x_i \oplus y_i),$$

where $\oplus$ denotes binary addition (XOR, Def. 1.2.1).

Definition 1.2.10 (Minimum distance)**.** The *minimum distance* of a code $\mathcal{C}$ is the minimum pairwise Hamming distance of codewords, computed as

$$d_{min} = \min_{\mathbf{x},\mathbf{y} \in \mathcal{C}, \mathbf{x} \neq \mathbf{y}} d_H(\mathbf{x}, \mathbf{y}).$$

For a linear code, this can be simplified to

$$d_{min} = \min_{\mathbf{x} \in \mathcal{C}, \mathbf{x} \neq \mathbf{0}} \{w_H(\mathbf{x})\}.$$

Let d_{min} be the minimum distance of a code $\mathcal{C}$ and let $t = w_H(\mathbf{e})$ be the Hamming weight of the error vector $\mathbf{e}$. An (n, k, d_{min}) code $\mathcal{C}$ can guarantee detection of all error patterns in an error vector $\mathbf{e}$ with $t < d_{min}$ or correct all error patterns if $t < \left\lfloor \frac{d_{min}-1}{2} \right\rfloor$. As it determines the error-correcting and detecting capacities of a code, the minimum distance property is one of the most important metrics in coding theory.

Codes can be classified based on various aspects and properties. At the most basic level, codes can either be non-linear (simply a set of codewords) or linear (Def. 1.2.5). As storing the set of all codewords of a general code is impractical, they do not have a wide range of uses. Codes are often divided into linear block codes and convolutional codes. Block codes, such as the Hamming code, work by dividing data into blocks matching the length of the code. Convolutional codes instead work on data streams, which can have an arbitrary length. Alternatively, convolutional codes can be terminated at a set length, which results in a block code.

Another way to classify codes is based on the decoding algorithms they employ, into algebraic and probabilistic codes. In practice, only short block codes can be decoded using optimal decoding due to complexity restrictions. Even for algebraic codes, such as Hamming, Bose–Chaudhuri–Hocquenghem (BCH) or Reed-Solomon (RS) codes, suboptimal decoding with polynomial complexity is used.

As an example, Hamming codes are a family of $[2^r - 1, 2^r - r - 1, 3]$ binary linear block codes. Being perfect codes, they have a minimum distance $d_{min} = 3$ with the highest possible code rate $R = k/n = 1 - \frac{r}{2^r - 1}$, thus attaining the Hamming (sphere-packing) bound. For perfect codes, Hamming spheres (Fig. 2) with radius t perfectly fill the space of all n binary vectors.

A Hamming code is determined by a parity-check matrix consisting of all possible non-zero columns of length r. For $r = 3$, the parity-check matrix of the $[7, 4, 3]$ Hamming code has the form

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

As per the systematic form $H = \left(P^\top | I_{n-k}\right)$, it is simple to obtain the generator matrix $G = (I_k | P)$

$$G = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}. \tag{1.1}$$

Take a message $\mathbf{m} = \begin{pmatrix} 1 & 0 & 1 & 0 \end{pmatrix}$. The corresponding codeword $\mathbf{c}$ is obtained by the encoding

$$\mathbf{c} = \mathbf{m}G = (1\ 0\ 1\ 0) \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix} = (1\ 0\ 1\ 0\ 1\ 0\ 1).$$

As Hamming codes have $d_{min} = 3$, they can guarantee correction of only $t = 1$ error.

Convolutional codes are infinitely long linear codes introduced by Elias in 1955 [Eli55]. Codewords of convolutional codes are produced by combining the input data with the *memory state* stored in the encoder. Upon arrival, new information symbols update the encoder memory. A convolutional encoder consists of k linear shift register (LSR)s, used to encode k input bits into n output bits ($R = k/n$). Each LSR $i = 1, \ldots, k$ has a length v_i, determining the encoder memory $m = \max_i(v_i)$. The *generators* $\mathbf{g}_i$ of a convolutional code are defined as binary arrays $\mathbf{g}_{ij} = (g_{ij0}, g_{ij1}, \ldots, g_{ijv_i})$, $i = 1, 2, \ldots, k$, $j = 1, 2, \ldots, n$. If $g_{ijh} = 1$, the h-th memory element of the i-th LSR is connected to the j-th output, where modulo two adders determine the final output of the j-th bit.

Convolutional codes are of particular interest due to the low state complexity of their trellis representations. This will be covered in more detail in Sec. 1.3.3.

Iteratively decodable codes (Sec. 1.3.4), such as turbo and LDPC codes, have overtaken their algebraic counterparts in practical communication systems ever since the "turbo-revolution" of the 1990s. Due to iterative decoding allowing for the decoding of long codes, near-capacity achieving communication has become a possibility. While turbo codes achieve excellent error correction performance, practical implementations of turbo decoders matching the throughput and power efficiency requirements of standards such as 5G have proven to be difficult, due to the non-parallelizable nature of Bahl-Cocke-Jelinek-Raviv (BCJR) decoding [Gei+23]. Due to this, codes considered for future practical applications can be limited to tail-biting convolutional codes, such as in the long term evolution (LTE) standard [ETS18] for lower block lengths (approx. $n \leq 128$) and lower code rates, polar codes [Ari09] (largely beyond the scope of this dissertation) for medium lengths (approx. $128 < n \leq 512$) and LDPC codes for higher rates and block lengths [Gei+23].

1.3. Decoding Techniques

As the differences in channel models and classes of codes are vast, it is obvious that the different decoding methods associated with them vary significantly as well. Classifying the methods as optimal, suboptimal and iterative, brief introductions will be given in the following sections.

1.3.1. Optimal Decoding

To introduce concepts behind decoding processes, it is helpful to start with the simplest models and methods. Consider the BSC model and binary linear $[n, k]$ code $\mathcal{C}$. A codeword $\mathbf{c}$ is sent over the noisy channel and, during transmission, an *error vector* $\mathbf{e}= (e_1, ..., e_n)$ is added: $\mathbf{y} = \mathbf{c} + \mathbf{e}$.

At the receiver, a *hard decision* must be made, taking a vector space of 2^n binary vectors and dividing them into 2^k decision regions corresponding to the codewords of $\mathcal{C}$. Each region is of size $2^r = 2^{n-k}$ and consists of vectors closest (by Hamming distance) to the codeword in the center of the region.

The simplest optimal decoding is implemented as an exhaustive search over the set of codewords, resulting in complexity proportional to 2^k. The codeword with minimal Hamming distance to the received vector $\mathbf{y}$ is chosen as the solution. This approach is optimal in terms of complexity if the rate of the used code is $R = k/n < 1/2$.

If $R > 1/2$, it is more economical to search for probable error vectors. As the 2^k decision regions are of size 2^r, a search of the 2^r most probable error vectors is optimal. Using the parity-check matrix H of $\mathcal{C}$, each error vector $\hat{\mathbf{e}}$ can be tested as $(\mathbf{y} + \hat{\mathbf{e}})H^\top$. A codeword $\hat{\mathbf{c}}$ is output as the decision if

$$(\mathbf{y} + \hat{\mathbf{e}})H^\top = (\mathbf{c} + \mathbf{e} + \hat{\mathbf{e}})H^\top = \hat{\mathbf{c}}H^\top = \mathbf{0}.$$

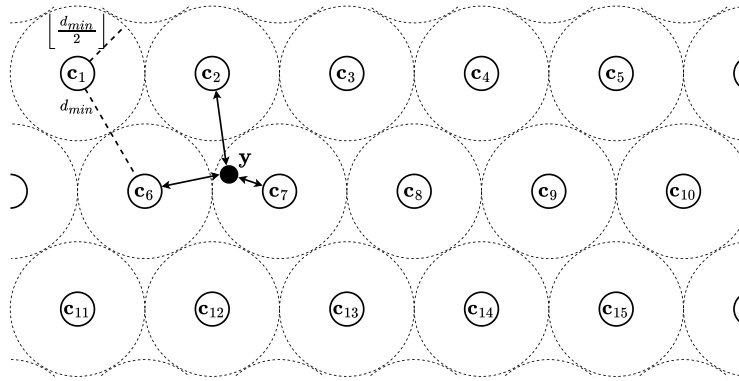

Figure 2. Two-dimensional visualization of Hamming spheres

Clearly, $\hat{\mathbf{c}} = \mathbf{c}$ only if $\hat{\mathbf{e}} = \mathbf{e}$.

An efficient method of decoding, known as *syndrome decoding*, reduces the previous procedure to a lookup table. The syndrome vector $\mathbf{s}$ is computed as

$$\mathbf{s} = \mathbf{y}H^\top = (\mathbf{c} + \mathbf{e})H^\top = \hat{\mathbf{e}}H^\top,$$

and compared to a pre-computed table of 2^r possible syndrome values and their corresponding minimum-weight error vectors $\hat{\mathbf{e}}$. By the addition $\mathbf{y} + \hat{\mathbf{e}}$, the solution $\hat{\mathbf{c}}$ is obtained.

Assume the same systematic $[7, 4, 3]$ Hamming code is used as in the previous example. The codeword $\mathbf{c} = (1010101)$ is sent over the BSC and an error vector $\mathbf{e} = (0010000)$ gets introduced in transmission. At the receiver, the syndrome $\mathbf{s}$ is computed:

$$\mathbf{s} = \mathbf{y}H^\top = (1\ 0\ 0\ 0\ 1\ 0\ 1) \begin{pmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{pmatrix}^\top = (0\ 1\ 1)\,.$$

By referencing the table, the error vector $\hat{\mathbf{e}} = (0010000)$ is obtained:

$\mathbf{s}$	$\mathbf{e}$
(000)	(0000000)
(001)	(0000001)
(010)	(0000010)
(**011**)	(**0010000**)
(100)	(0000100)
$\vdots$	$\vdots$

The transmitted codeword $\mathbf{c}$ is then easily obtained as $\mathbf{c} = \mathbf{y} + \hat{\mathbf{e}}$.

If transmission results in more errors than the code can correct, the decoder fails to produce the correct output. Suppose, for example, that $\mathbf{e} = (1100000)$ instead. Again, the obtained syndrome is $\mathbf{s} = (011)$. As a

result:

$$(\mathbf{y}+\hat{\mathbf{e}})H^{\top} = ((0\ 1\ 1\ 0\ 1\ 0\ 1)+(0\ 0\ 1\ 0\ 0\ 0\ 0))\,H^{\top} = \mathbf{0}.$$

As $\hat{\mathbf{c}} = (0100101)$ is another codeword, being the encoding of $\hat{\mathbf{m}} = (0100)$, it would be erroneously decided that it was the message sent.

Definition 1.3.1 (Decoding error event)**.** A *decoding error event* denotes the case of a received message or codeword not coinciding with the transmitted message or codeword.

Definition 1.3.2 (Error probability)**.** The likelihood of a decoding error event under specific channel conditions is called *error probability.*

Error probability measurements in digital communication systems are expressed separately for received bits and frames (blocks):

Definition 1.3.3 (Bit error rate)**.** The measure of error probability of received bits, expressed as the average ratio of erroneous received bits to the total count of received bits, is called the bit error rate (BER).

Definition 1.3.4 (Frame error rate)**.** Analogously, the average ratio of frames (blocks) received in error to the total amount of received frames is called the frame error rate (FER).

1.3.2. ML, MAP and Symbol-MAP Decoding

While there are varying techniques for optimal decoding of received noisy codewords, they can be divided in two: maximum a posteriori probability (MAP) and maximum likelihood (ML) decoding. These techniques are paramount for decoding of short codes, as well as concatenated code constructions and demodulation.

For a linear $[n, k]$ code $\mathcal{C}$, consider transmitted codewords $\mathbf{c}$, received noisy sequences $\mathbf{y}$ and output $\hat{\mathbf{c}}$ of the decoder, under the assumption that it is a codeword with the highest probability of being the transmitted codeword $\mathbf{c}$, based on the probability rule used by the decoder.

For MAP decoding, $\hat{\mathbf{c}}$ is chosen by maximizing the a posteriori probability $p(\cdot|\cdot)$ of $\mathbf{c}$ based on the channel output $\mathbf{y}$:

$$\hat{\mathbf{c}} = \arg\max_{\mathbf{c}\in\mathcal{C}} p(\mathbf{c}|\mathbf{y}),\ \text{where } p(\mathbf{c}|\mathbf{y}) = \frac{p(\mathbf{y}|\mathbf{c})p(\mathbf{c})}{p(\mathbf{y})}. \tag{1.2}$$

As the goal is to maximize the probability over all $\mathbf{c} \in \mathcal{C}$, the denominator $p(\mathbf{y})$ can be ignored:

$$\hat{\mathbf{c}} = \arg\max_{\mathbf{c}\in\mathcal{C}} p(\mathbf{y}|\mathbf{c})p(\mathbf{c}).$$

The ML decoding rule simply maximizes the probability of $\mathbf{y}$ over all codewords $\mathbf{c} \in \mathcal{C}$:

$$\hat{\mathbf{c}} = \arg\max_{\mathbf{c}\in\mathcal{C}} p(\mathbf{y}|\mathbf{c}).$$

Notably, if codewords $\mathbf{c}$ are either uniformly distributed or their probabilities are unknown, the two decoding rules are equivalent.

While MAP decoding outputs a codeword $\hat{\mathbf{c}}$ with maximal probability, symbol-MAP decoding can be used instead to output soft decisions for probabilities of each code symbol $\hat{c}_i, i \in \{1, \dots n\}$. The symbol-optimal decoder outputs two sequences:

$$\hat{\mathbf{c}} = \{\hat{c}_1, \dots, \hat{c}_n\},\ \hat{c}_i = \arg\max_{c_i} p(c_i|\mathbf{y});$$

$$\mathbf{L} = \{L_1, \dots, L_n\},\ L_i = p(\hat{c}_i|\mathbf{y}).$$

The sequence $\mathbf{L}$ of probabilities describes the reliability of decisions for each of the symbols $\hat{c}_i$. The decoded symbol sequence $\hat{\mathbf{c}}$ does not necessarily correspond to any codeword $\mathbf{c} \in \mathcal{C}$. Referred to as a soft output, such constructions are required for soft-input soft-output (SISO) algorithms, where decoding is done in multiple stages, such as, for example, algorithms used to decode concatenated codes. Symbol-MAP decoding also plays an important role in iterative decoding of turbo and LDPC codes.

Denoting as the set of codewords $C_i(c)$ all codewords $\mathbf{c} \in \mathcal{C}$ that have value $c \in \{0, 1\}$ at position i, the maximized a posteriori probability is expressed as:

$$p(c_i = c|\mathbf{y}) = \frac{p(c_i = c, \mathbf{y})}{p(\mathbf{y})} = \frac{\sum\limits_{\mathbf{c} \in C_i(c)} p(\mathbf{c}, \mathbf{y})}{p(\mathbf{y})}. \tag{1.3}$$

It must be noted that the basic symbol-MAP implementation in Eq. (1.3), assuming an $[n, k]$ code $\mathcal{C}$, requires an exhaustive search over all 2^k codewords for each decoded code symbol $\hat{c}_i$. However, simplified implementations exist, making use of graph representations of codes, as covered in the following section.

In comparison, by definition, any straightforward implementation of ML decoding requires an exhaustive search among 2^k codewords of the decoded code. Thus, for a fixed rate $R = k/n$, resulting in an exponential complexity increase as the length n grows.

1.3.3. Viterbi and BCJR Algorithms

Due to restrictive complexities of the previously covered decoding schemes, simplifications are required to allow for the decoding of long codes, in turn allowing for near-capacity achieving decoding. The code trellis based approach to simplifying decoding is the most often used method for achieving this. In this section, the general algorithm is described and considered, with further simplifications specific to the decoding of LDPC codes are covered in Sec. 2.4.1.

Definition 1.3.5 (Trellis)**.** A trellis $\mathcal{T} = (\mathcal{S}, A, \mathcal{E})$ is a graph used to represent a code $\mathcal{C}$, where each codeword $\mathbf{c} \in \mathcal{C}$ is represented by a path in $\mathcal{T}$. The set $\mathcal{S}$ of nodes (states) and alphabet A are used to define ordered triples $(s, s', a) \in \mathcal{E}$, called edges, where $s, s' \in \mathcal{S}, a \in A$.

A trellis diagram is a specific construction, where:

1. Each vertex (node) belongs to a layer $\mathcal{S}_i$. For any two vertices $u, v \in \mathcal{S}_i$, there is no edge e between them - making the layers nonintersecting.
2. Movement in a trellis occurs along paths, traveling between connected vertices in layers of increasing order (from $\mathcal{S}_i$ to $\mathcal{S}_{i+1}$).
3. Each vertex other than v_0 has at least one incoming edge (from a lower layer), and each vertex other than v_f has at least one outgoing edge (to a higher layer).
4. The first ($\mathcal{S}_0$) and last ($\mathcal{S}_f$) layer both contain only one vertex (v_0 and v_f, correspondingly).
5. Each edge $e \in \mathcal{E}$ is labeled with symbols of codewords $\mathbf{c}$ or a metric, such as Hamming or squared Euclidean distance of the symbol from a corresponding symbol in a received sequence.

Definition 1.3.6 (Conventional trellis)**.** A *conventional trellis* $\mathcal{T} = (\mathcal{S}, A, \mathcal{E})$ for an $[n, k]$ code is a directed graph with labeled edges $e \in \mathcal{E}$, where the state layers can be decomposed as $\mathcal{S} = \mathcal{S}_0 \cup \mathcal{S}_1 \cup \ldots \mathcal{S}_f$, and, for each edge $e = (s, s', a) \in \mathcal{E}$, if $s \in \mathcal{S}_i$, then $s' \in \mathcal{S}_{i+1}$ for $i = 0, 1, \ldots, n$.

Definition 1.3.7 (Sectionalized trellis)**.** A trellis $\mathcal{T} = (\mathcal{S}, A, \mathcal{E})$, where all edges $(s, s', a) \in \mathcal{E}$ from $\mathcal{S}_t$ to $\mathcal{S}_{t+1}$ have a length of $l_t \geq 1$, $t = 0, 1, 2, \ldots, f$, and $\sum_{t=1}^{f} l_t = n$ for a code $\mathcal{C}$ of length n, is called a *sectionalized trellis.*

Definition 1.3.8 (Bitwise trellis)**.** The special case of the trellis with $l_t = 1$, $t = 1, \ldots, n$ is called a *bitwise* (or full unsectionalized [LV96]) trellis.

Definition 1.3.9 (State complexity)**.** The maximal *state complexity* μ_s of a sectionalized trellis $\mathcal{T}$ is

$$\mu_s = \max_{0,1,\ldots,f} \{\log_2 |\mathcal{S}_i|\}.$$

As a well-known example of trellis-based decoding, the Viterbi algorithm [Vit67] was proposed as a decoding algorithm for convolutional codes. It is a dynamic programming technique that is used to find the most probable path, known as the Viterbi path, through a trellis that minimizes "distance" - the metric associated with each edge of the trellis, computed based on a received sequence. Utilizing the algorithm, the Viterbi decoder is an ML decoding algorithm.

The complexity of trellis decoding is determined by the state complexity μ_s of the used trellis, shown in [Wol78] to be $\mu_s \leq \min\{k, r\}$. Thus, to minimize decoding complexity, the state complexity of the trellis must be minimized.

Definition 1.3.10 (Minimal trellis)**.** A trellis $\mathcal{T}$ for a linear code $\mathcal{C}$ is called minimal if, for every other trellis $\mathcal{T}'$ of $\mathcal{C}$, no layer $\mathcal{S}_i$ of $\mathcal{T}$ has a size greater than the corresponding layer $\mathcal{S}'_i$ of $\mathcal{T}'$ ($|\mathcal{S}_i| \leq |\mathcal{S}'_i|$ for all i).

All linear codes have a minimal trellis, which can be obtained from a generator matrix G of $\mathcal{C}$ that is in minimal span form (MSF), meaning that, for each row of G, the positions of first and last nonzero symbols are distinct. Consider a generator matrix G in MSF:

$$G = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{pmatrix}. \tag{1.4}$$

The minimal trellis constructed from the matrix G is shown in Fig. 3.

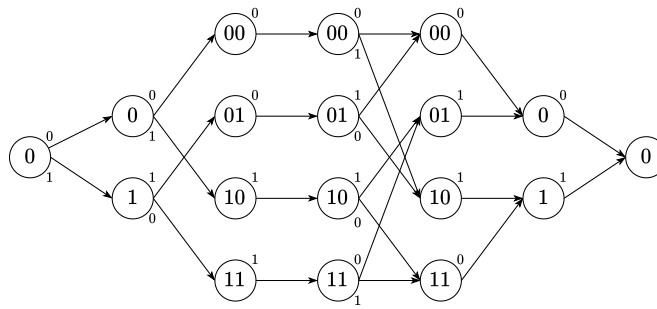

Figure 3. Minimal trellis constructed from G in Eq. (1.4)

Due to the very regular nature of their trellises, convolutional codes are of particular interest in the context of low complexity trellis-based decoding. In [MW86], three methods of of constructing block codes from convolutional codes are described. *Direct truncation* (DT), where codewords of the block code are obtained as the n-bit outputs of the convolutional encoder based on all possible k-bit inputs, which results in insufficient protection of the last information bits. *Zero tail* (ZT), where all k-bit inputs are followed by a k_0-bit tail of zeros, resulting in rate loss of the resulting block code. Tail biting (TB) ([Sv79]), obtained by initializing the convolutional encoder with the last k_0 information bits, while ignoring the output. The TB construction does not result in rate loss and offers full protection of all bits at the cost of some added complexity in decoding. In block form, the n_0-bit "tail" corresponding to the k_0-bit zero tail is cut from the end and moved to the beginning of the generator matrix. The three methods are visualized in Fig. 4.

Differing from ZT terminated convolutional codes, TB codes have codewords corresponding to paths in its trellis diagram that start and end at the same state. For example, consider the generator matrix G_{TB} of a $[10, 5]$ TB

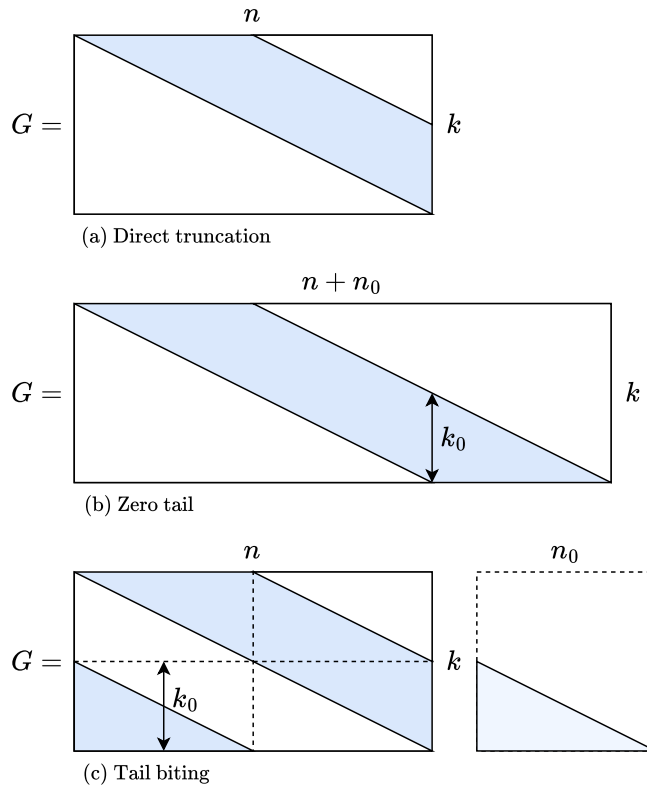

Figure 4. Three methods of block code construction from convolutional codes

code obtained from the $R = 1/2$ convolutional code $G(D) = (1, 1 + D + D^2)$:

$$G_{TB} = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 1 & & & & \\ & & 1 & 1 & 0 & 1 & 0 & 1 & & \\ & & & & 1 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & & & & & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & & & & & 1 & 1 \end{pmatrix}, \tag{1.5}$$

with only the non-trivial elements of G_{TB} filled to highlight the code structure. The missing entries are all zero elements. The corresponding TB trellis diagram is given in Fig. 5.

For the TB trellis in Fig. 5, being a sectionalized trellis of a $[10, 5]$ TB code with each edge having length two ($l_t = 2$), each of the 10-bit codewords is represented by a length five path through the trellis diagram. With exactly two edges leaving each state, with the exception of the last layer, the edge labels are given next to each state (node), corresponding to the relative vertical directionality of the exiting edges.

Denote the number of states at each level of the trellis as 2^ν. A straightforward implementation of ML decoding using the Viterbi algorithm, starting

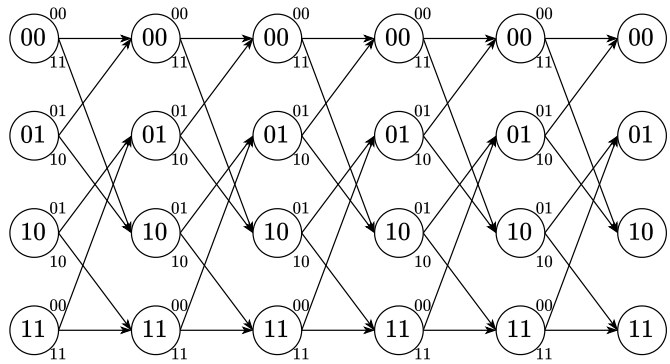

Figure 5. TB trellis diagram of the $[10, 5]$ code in Eq. (1.5)

at each inital trellis state, searching for paths to a matching final state, requires a total of 2^{ν} Viterbi trials, resulting in a total complexity of $O(2^{2\nu})$.

The BCJR algorithm [Bah+74] is based on the trellis representation of an ECC, enabling iterative symbol-MAP decoding used for turbo and LDPC codes.

The a posteriori probability of a symbol $c \in \{0, 1\}$ at position t is computed as

$$p(c_t = c|\mathbf{y}) = \frac{p(c_t = c, \mathbf{y})}{p(\mathbf{y})}. \tag{1.6}$$

Denote the set of all codewords $\mathbf{c} \in \mathcal{C}$ that have value c at position t as $C_t(c)$,

$$p(c_t = c, \mathbf{y}) = \sum_{\mathbf{c} \in C_t(c)} p(c, \mathbf{y}) = \sum_{(m',m) \in S_t(c)} p(s_{t-1} = m', s_t = m, \mathbf{y}), \tag{1.7}$$

where (m', m) is a trellis state pair at a time moment t (corresponding to position t), and $S_t(c)$ is the set of state pairs corresponding to the symbol c.

Then, the logarithmic likelihood ratio (LLR) for the symbol c_t is

$$\lambda(c_t) = \ln \frac{p(c_t = 1|\mathbf{y})}{p(c_t = 0|\mathbf{y})} = \ln \frac{\sum\limits_{(m',m) \in S_t(1)} p(s_{t-1} = m', s_t = m, \mathbf{y})}{\sum\limits_{(m',m) \in S_t(0)} p(s_{t-1} = m', s_t = m, \mathbf{y})}. \tag{1.8}$$

The representation of likelihoods in logarithmic domain allows for a significant reduction in decoding complexity. A simplification of the BCJR algorithm for decoding of the single parity-check codes required for LDPC codes is covered as part of Sec. 2.4.1.

Represent the distribution $p(\cdot)$ from Eq. (1.8) as

$$\sigma_t(m', m) = p(s_{t-1} = m', s_t = m, \mathbf{y}) = \alpha_{t-1}(m')\gamma_t(m', m)\beta_t(m), \tag{1.9}$$

where, a partial vector $\mathbf{y}$ from index i to j is denoted as $\mathbf{y}_i^j = (y_i, \ldots, y_j)$,

$$\begin{aligned} \alpha_t(m) &= p(s_t = m, \mathbf{y}_1^t), \\ \gamma_t(m', m) &= p(s_t = m, y_t | s_{t-1} = m'), \\ \beta_t(m) &= p(\mathbf{y}_{t+1}^n | s_t = m). \end{aligned} \tag{1.10}$$

The computations of $\alpha_t(m)$ and $\beta_t(m)$ can be performed recursively on the trellis and, for a bitwise trellis, the *edge metric* γ is computed as

$$\gamma_t(m', m) = \sum_{c_t} \Pr(s_t = m, c_t, y_t | s_{t-1} = m') = \sum_{c_t} p(c_t | m, m') p(y_t | c_t).$$

The decoding algorithm can be expressed as three steps:

- With initial $\alpha_0(0) = 1, \alpha_1(m) = 0, m \neq 0$, for $t = 1, \ldots, n$ (forward pass):
$$\alpha_t(m) = \sum_{m'} \alpha_{t-1}(m') \gamma_t(m', m). \tag{1.11}$$
- With initial $\beta_n(0) = 1, \beta_n(m) = 0, m \neq 0$, for $t = n-1, \ldots, 0$ (backward pass):
$$\beta_t = \sum_{m'} \beta_{t+1}(m') \gamma_{t+1}(m', m). \tag{1.12}$$
- For $t = 1, \ldots, n$ compute LLRs using $\sigma_t(m', m)$ from Eq. (1.9):
$$\lambda_t = \frac{\sum\limits_{(m', m) \in S_t(1)} \sigma_t(m', m)}{\sum\limits_{(m', m) \in S_t(0)} \sigma_t(m', m)}. \tag{1.13}$$

The absolute value of $|\lambda(c_t)|$ in Eq. (1.8) is referred to as the *reliability* of c_t, whereas the *hard decision* $\hat{c}_t$ is made as

$$\hat{c}_t = \begin{cases} 0 & \lambda(c_t) \leq 0 \\ 1 & \lambda(c_t) \geq 0 \end{cases}. \tag{1.14}$$

The complexity of BCJR decoding is proportional to the number of states in the trellis representation of the code. For this reason, techniques such as using ZT and TB trellises, reduced memory states and others are used in practice to balance trade-offs between complexity and decoding error rates.

1.3.4. Iterative Decoding

As mentioned, the BCJR algorithm is an integral part of iterative decoding for turbo codes [BGT93]. Being the first practical codes to approach the Shannon limit, their introduction can be considered a revolutionary event in

the field, leading to a wider adaptation of iterative decoding techniques and the rediscovery of LDPC codes in [MN96].

A model of the turbo decoder is given in Fig. 6 to illustrate the general design of iterative decoding schemes.

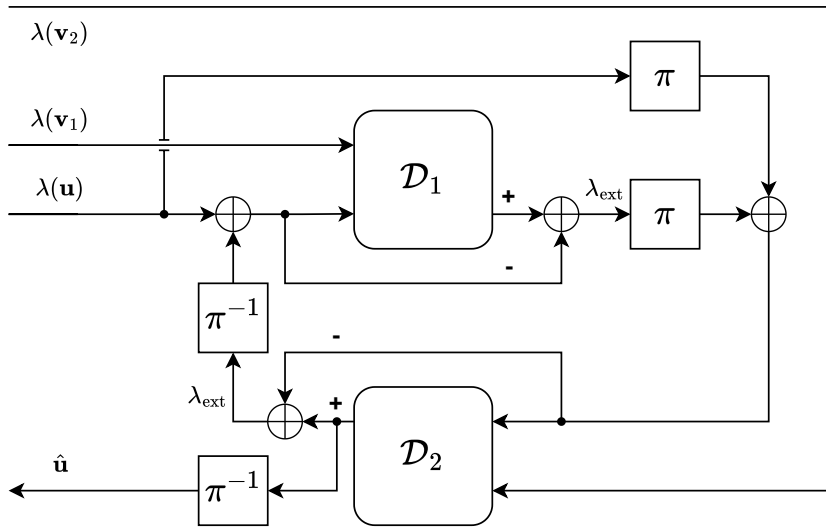

Figure 6. Turbo decoder

Assume a systematically encoded sequence $\mathbf{y}$ was received as channel output. Transforming the received sequence into LLRs $\lambda(\mathbf{y})$, triplets of $(\lambda(u_t), \lambda(v_{1t}), \lambda(v_{2t}))$ of an information bit and two parity bits, for a time instance t, are obtained by appropriately reordering the sequence. At the core of the decoding scheme are two BCJR decoder blocks $\mathcal{D}_1, \mathcal{D}_2$. Performing SISO symbol-MAP decoding on the information bit u_t and a parity bit v_{jt} (for $\mathcal{D}_j$), using information gathered from all positions other than t (except in the first iteration, where only channel outputs are considered), the outputs λ_{ext} of the decoding blocks $\mathcal{D}_j$ are referred to as *extrinsic information*.

Extrinsic information is obtained from the output of a decoder after subtraction of "intrinsic" information and is used in the following decoding steps (both from $\mathcal{D}_1$ to $\mathcal{D}_2$, and in following iterations via the *feedback loop*) to progressively improve reliabilities of decoding decisions. An analogous concept was also proposed as part of Gallager's original LDPC decoders in [Gal62].

Using a feedback loop, the component decoders $\mathcal{D}_1$, $\mathcal{D}_2$ iteratively improve reliability of decoding decisions, until a maximum number of iterations is reached or a reliability threshold is reached, at which point, an output $\hat{\mathbf{u}}$ is produced. An increase in the maximum number of decoding iterations has a diminishing effect on improving the BER achieved. In [BGT93], a comparison of 1-18 iterations is provided. In practice, it is common to limit

the number of iterations to 5-10, as decoding performance gains from further increased number of iterations are limited.

While the iterative decoding techniques used for LDPC codes share aspects with turbo decoding, due to the very different structures of the code classes, techniques used for them also differ significantly from turbo codes. The decoding of LDPC codes is described in detail in Sec. 2.4.1.

1.4. Channel Models

A channel is determined by conditional probability distributions of the output sequences, given the input sequences. Communication channels can be divided in two groups: *continuous-time* and *discrete-time* channels. In what follows, only discrete-time channels are considered. The discrete-time channels can have either continuous (analog) or discrete input and output alphabets.

Definition 1.4.1 (Stationary channel)**.** If, for any n, j and any sequences $\mathbf{x}_{j+1}^{j+n} \in X^n, \mathbf{y}_{j+1}^{j+n} \in Y^n$, conditional probabilities $P\left(\mathbf{y}_{j+1}^{j+n} | \mathbf{x}_{j+1}^{j+n}\right)$ do not depend on j, the channel is called *stationary.*

Definition 1.4.2 (Memoryless channel)**.** If, for any n, j, and any sequences $\mathbf{x}_{j+1}^{j+n} \in X^n, \mathbf{y}_{j+1}^{j+n} \in Y^n$,

$$P(\mathbf{y}_{j+1}^{j+n} | \mathbf{x}_{j+1}^{j+n}) = \prod_{i=j+1}^{j+n} P(y_i | x_i),$$

the channel is called *memoryless.*

A real channel takes a length-n input over an alphabet X, converting it into a sequence over an alphabet Y. A *discrete stationary channel* is a discrete-time channel that also outputs a discrete alphabet. More simply, in a memoryless channel setting, it is assumed that noise introduced in transmission through the channel acts independently across time, having no "memory" of what came before. A discrete memoryless channel (DMC) is determined by a transition probability matrix

$$P = \begin{pmatrix} p_{0,0} & p_{0,1} & \cdots & p_{0,L-1} \\ p_{1,0} & p_{1,1} & \cdots & p_{1,L-1} \\ \vdots & \vdots & & \vdots \\ p_{K-1,0} & p_{K-1,1} & \cdots & p_{K-1,L-1} \end{pmatrix}, \tag{1.15}$$

where $[K-1]$ is the input alphabet and $[L-1]$ is the output alphabet.

1.4.1. Binary Symmetric Channel

Due to its simplicity, the binary symmetric channel (BSC) is a channel model that is often used for theoretical performance analysis of ECCs. The

BSC has binary input and output alphabet $X = Y = \{0, 1\}$ and symmetric transition probabilities $p_{0,1} = p_{1,0} = p$, $p_{0,0} = p_{1,1} = p - 1 = q$. In matrix form,

$$P = \begin{pmatrix} p-1 & p \\ p & p-1 \end{pmatrix}. \tag{1.16}$$

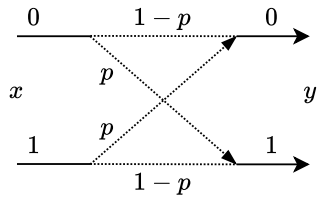

Figure 7. BSC channel model

The only parameter of the BSC is the *crossover probability* p. It determines the probability of a bit flip occurring during its transmission over the channel. Having both binary input and output, the channel output $\mathbf{y}$ is expressed as:

$$\mathbf{y} = \mathbf{c} \oplus \mathbf{e}; \quad \mathbf{e} = (e_1, ..., e_n); \quad p(e_i = 1) = p,$$

where $\mathbf{c}$ is the transmitted codeword and $\mathbf{e}$ denotes the error vector.

A fundamental notion in information theory, called *channel capacity* C, is used to describe the limit on the rate of transmission (Def. 1.6.1) R_T at which information could reliably be transmitted over a noisy channel. Introduced by Shannon [Sha48], capacity of the BSC is expressed as

$$C = 1 - h(p),$$

using Shannon entropy of a binary distribution:

$$h(p) = -p \log_2(p) - (1 - p) \log_2(1 - p).$$

1.4.2. Additive White Gaussian Noise Channel

The additive white Gaussian noise (AWGN) channel can be described either in the form of a continuous- or a discrete-time channel. A discrete-time channel can always be obtained by sampling of the continuous-time channel according to the Nyquist-Shannon sampling theorem. Hereafter, only the discrete AWGN channel will be considered. While not a perfect representation of real-world conditions, the AWGN channel is a much more representative model when compared to the BSC. The AWGN channel outputs a sum

$\mathbf{y} = \mathbf{x} + \mathbf{n}$ of the channel input signal $\mathbf{x}$ and noise $\mathbf{n}$, which is a sequence of independent, identically distributed zero-mean Gaussian random variables with variance $\sigma^2 = N_0/2$, where $\frac{N_0}{2}$ is the two-sided spectral density of the white Gaussian noise.

Definition 1.4.3 (Signal-to-noise ratio)**.** The *Signal-to-noise ratio* (SNR) is used to describe the average energy E_s of the transmitted signals compared to the total noise power, computed as

$$\mathrm{SNR} = \frac{E_s}{\sigma^2}. \tag{1.17}$$

The capacity of the discrete-time AWGN channel is given in [Sha48] as:

$$C = \frac{1}{2}\log_2\left(1 + \mathrm{SNR}\right). \tag{1.18}$$

The output of the discrete-time AWGN is expressed using sequences $\mathbf{y} = \mathbf{x} + \mathbf{n}$, with the signal sequence $\mathbf{x}$ consisting of signal points from a modulation alphabet: $x_i \in \mathcal{M}$. In the simplest case of binary phase-shift keying (BPSK) modulation, $\mathcal{M} = \{-\sqrt{E_s}, +\sqrt{E_s}\}$, with binary input $b_i \in \{0, 1\}$ mapped to $\mathcal{M}$ as

$$x_i = (2b_i - 1)\sqrt{E_s}.$$

Demodulation, producing a hard decision on $\hat{b}_i$, can thus be implemented simply by measuring the sign of the received signal $y_i = x_i + n_i$:

$$\begin{cases} y_i \geq 0 \Rightarrow \hat{b}_i = 1 \\ y_i < 0 \Rightarrow \hat{b}_i = 0 \end{cases}. \tag{1.19}$$

In the case of BPSK modulated AWGN channel, as in Fig. 8, the likelihood of erroneous transmission is analogous to the BSC case, but the crossover probability is replaced by the *error region* of the conditional probability density function (PDF).

As the noise n_i is an independent Gaussian random variable with variance σ^2 and mean $\mu = 0$, the probability of a received symbol y_i conditional on transmitted $x_i \in \mathcal{M}$ is expressed as:

$$P(y_i|x_i) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(y_i - x_i)^2}{2\sigma^2}}.$$

By quantization, using Eq. (1.19), the BPSK-modulated AWGN is fully reduced to BSC, using the *complementary Gaussian distribution function* $Q(x)$, with crossover probability

$$p = Q(\sqrt{2\mathrm{SNR}}) = Q\left(\frac{\sqrt{E_s}}{\sigma}\right) = \frac{1}{\sqrt{2\pi}} \int_{\sqrt{E_s}/\sigma}^{\infty} e^{-y^2/2}\, dy = \frac{1}{2}\mathrm{erfc}\left(\frac{\sqrt{E_s}}{\sqrt{2}\sigma}\right).$$

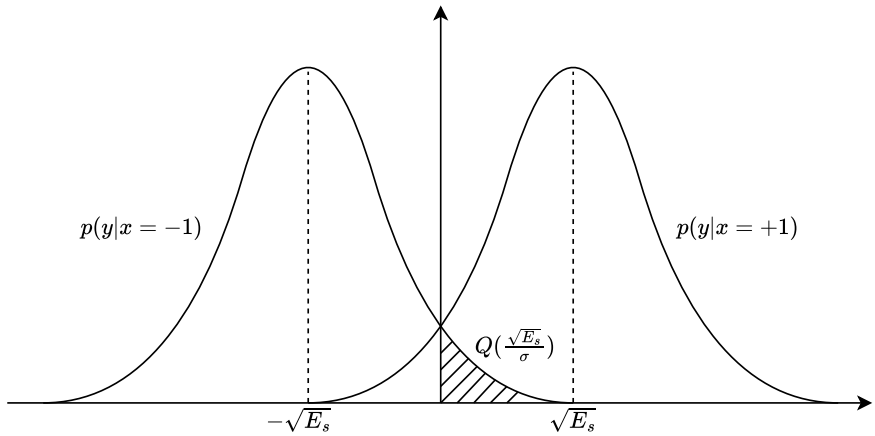

Figure 8. BPSK-modulated AWGN with highlighted crossover region

1.5. Modulation

The process of modulation transforms sequences consisting of elements of a finite alphabet - the code symbols, to sequences of parameters, by which characteristics of an electromagnetic harmonic waveform are augmented in order to send information over a channel. The inverse operation, demodulation, estimates these parameters based on the received waveform and outputs estimates of reliability (or quantized hard decisions) for the code symbols.

Different methods of carrier wave manipulation or modulation include changing either the frequency, amplitude or phase, or a combination of these properties, of the carrier signal to contain the transmitted information. Some of the best known of these methods include continuous-time variants, such as *amplitude* and *frequency modulations* (AM, FM) used in radio broadcasting. Due to inapplicability in the context of digital communications, these methods are not covered in this dissertation, focusing instead on the following types.

Quadrature amplitude modulation (QAM) signals are represented by two orthogonal pulse amplitude modulation (PAM)-signal components (Fig. 10). PAM (Fig. 9) is also known as amplitude shift keying (ASK) modulation. The achievable transmission rate per signal for QAM signaling in the AWGN channel is precisely double that of PAM signaling with the same SNR ratio per signal component. As such, for the analysis of channel modulation that follows, no distinction is made between research results for QAM and PAM, meaning that results obtained for M-PAM signaling can be easily reformulated for M^2-QAM signaling.

1.5.1. Signal Constellations

In the context of digital communication systems, the process of modulation transforms a sequence of binary data to a sequence of symbols from a discrete set of signals. The size of the signal set $\mathcal{M}$ is referred to as the *modulation order* M and denoted as, for example, M-PAM. The signal set $\mathcal{M}_{PAM}(M)$ of M-PAM signals consists of uniformly spaced signed odd-valued points:

$$\mathcal{M}_{PAM}(M) = \{-M+1, \ldots, -3, -1, 1, 3, \ldots, M-1\}.$$

A modulation scheme of order $M = 2^p$ has to map binary sequences of length p to the M signal points. A commonly used technique of assigning sequences to points is called *Gray coding*, introduced in [Gra53].

Definition 1.5.1 (Gray code)**.** A *Gray code* is a binary numbering system, in which any two adjacent values differ by exactly one bit.

Gray mapping was proven in [Agr+04] to be the optimal constellation labeling for PAM, QAM and phase-shift keying (PSK) constellations in terms of achievable bit error probabilities. The mapping is common in practical applications, for example, as part of bit-interleaved coded modulation (BICM) [GMC08], covered in Sec. 1.5.2.

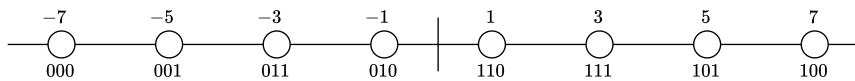

Figure 9. 8-PAM modulation with Gray mapping

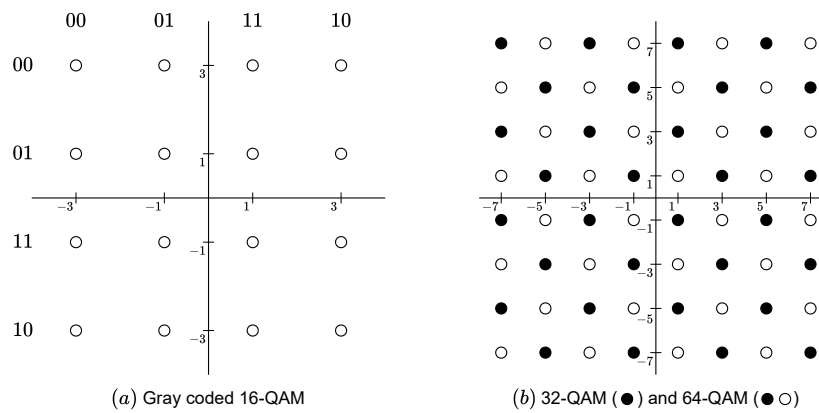

Figure 10. Signal constellation diagrams for QAM signalling

Due to the Gaussian distribution of error magnitudes, the symbols neighboring a transmitted symbol are the most likely ones to be received in error. Due to this, Gray coding (Fig. 9, 10(a)) is used to map binary sequences differing by at most one bit to adjacent signal points in order to minimize the amount of resulting bit errors after demodulation.

1.5.2. Coded Modulation

Modulation in digital communication systems is used to transform the encoded discrete-time information sequences into continuous-time waveforms that the real world requires. In 1982, trellis-coded modulation was introduced by Ungerboeck in [Ung82], which was designed to increase the Euclidean distance between channel signal sequences by encoding the modulated signals. Since then, it has been widely accepted that modulation and coding needs to be combined in a single entity, in order to approach channel capacity.

Bit-interleaved coded modulation (BICM) [CTB98], is a well-known coded modulation model, used in wireless communication, such as LTE [ETS18], 5G [ETS21] and Wi-Fi [EE21]. It can be generalized as a concatenation of an encoder of a binary code $\mathcal{C}$ and a N-dimensional modulator over a signal set $\mathcal{M}$. It uses a bit interleaver π and a one-to-one map $\mu : \{0,1\}^m \to \mathcal{M}^n$. The idea behind interleaving is to remove local dependencies and thus spread "bad" (unrecoverable) error events over different modulation symbols, reducing the harmful impact of their occurrence [GMC08].

To map the m-bit interleaved output of π onto signal sequences, often, a Gray mapping (*labeling*) is used (Fig. 9). In practice, the mapping can be implemented as a simple lookup table (Table 1).

i	$bin(i)$	$\mathbf{x} \in \mathcal{M}^2$	i	$bin(i)$	$\mathbf{x} \in \mathcal{M}^2$
0	0000	$(3,-3)$	8	1000	$(-3,-3)$
1	0001	$(3,-1)$	9	1001	$(-3,-1)$
2	0010	$(3,\ \ 3)$	10	1010	$(-3,-3)$
3	0011	$(3,\ \ 1)$	11	1011	$(-3,\ \ 1)$
4	0100	$(1,-3)$	12	1100	$(-1,-3)$
5	0101	$(1,-1)$	13	1101	$(-1,-1)$
6	0110	$(1,\ \ 3)$	14	1110	$(-1,\ \ 3)$
7	0111	$(1,\ \ 1)$	15	1111	$(-1,\ \ 1)$

Table 1. Example of a Gray coded mapping for 16-QAM signals, with $\mathcal{M} = \{-3,-1,1,3\}$

1.5.3. Shaping

To approach the Shannon limit over the AWGN channel, shaping of modulation signals is required, in addition to good error-correcting codes and coded modulation. To maximize the mutual information $I(X;Y)$ between AWGN channel input X and output Y, the input distribution $\{p(x)\}$ of modulated signals $x \in \mathcal{M}$ must approach the Gaussian distribution [FW89].

In practice, digital communication systems operate over discrete signal sets. If implemented in a straightforward fashion, they would produce uniformly distributed channel inputs X, which can not approach channel capacity. In theory, an ultimate gain of 1.53dB, in terms of SNR, is achievable

by a Gaussian distribution of $\{p(x)\}$ with zero mean and variance E_s [For+84] compared to a uniform distribution.

There are many different approaches to shaping, designed for specific types of codes, modulation schemes and channel models. In Sec. 4.3, the focus is on probabilistic amplitude shaping (PAS).

Two diverging approaches to shaping were introduced: geometric constellation shaping (GCS) [Sv93], using Gaussian-like signal sets with low-dimensional equiprobable signaling, and probabilistic constellation shaping (PCS) (e.g. [FW89], [For92], [KP93]), where a signal constellations over a signal set $\mathcal{M}$ in an N-dimensional vector space are designed and chosen based on nonuniform signal distributions. Both strategies impose their own sets of challenges when it comes to practical implementations. With PCS, the use of soft information in demodulation can be challenging, wrongly detected symbols can lead to error propagation and rate-variable schemes require solutions for desynchronization and buffering. In [BSS15], PAS was introduced as a concatenated distribution matcher and systematic binary encoder, designed to meet requirements of contemporary communications standards, such as DVB-S2. In [Gül+20b], enumerative sphere shaping (ESS) proposed as a solution to rate loss resulting from short block lengths with constant composition distribution matching (CCDM) [SB16] with PAS.

i	$\mathbf{A}^N(i)$	i	$\mathbf{A}^N(i)$
0	$(1,1,1,1)$	10	$(1,5,1,1)$
1	$(1,1,1,3)$	11	$(3,1,1,1)$
2	$(1,1,1,5)$	12	$(3,1,1,3)$
3	$(1,1,3,1)$	13	$(3,1,3,1)$
4	$(1,1,3,3)$	14	$(3,1,3,3)$
5	$(1,1,5,1)$	15	$(3,3,1,1)$
6	$(1,3,1,1)$	16	$(3,3,1,3)$
7	$(1,3,1,3)$	17	$(3,3,3,1)$
8	$(1,3,3,1)$	18	$(5,1,1,1)$
9	$(1,3,3,3)$		

Table 2. ESS shaping set, $N = 4$, $E_{max} = 28$ 8-PAM modulation, amplitudes $\mathcal{A} = \{1,3,5,7\}$, [Gül+20b]

Notably, the used 8-PAM system includes amplitude $a = 7$, which is not included in the example shaping set. The set of permissible signal sequences $\mathbf{A}^N = \{a_1, \ldots, a_N\}$ is determined by

$$\mathcal{S}_{E_{max}}^{N,m} = \left\{ (a_1, a_2, \ldots, a_N) \Big| \sum_{n=1}^{N} a_n^2 \leq E_{max} \right\},$$

resulting in signal sequences contained within an N-dimensional sphere with radius $r_\circ^{sp} = \sqrt{E_{max}}$.

The intuition behind a spherical shaping of signal sets comes from the optimality of a Gaussian distribution of signal points. As the signal points are not continuous, the discrete domain counterpart, Maxwell-Boltzmann (MB) distribution, is used to provide a target probability mass function (PMF), which determines the target amplitude composition $\{\#(a)\}, a \in \mathcal{A}$, of the shaping set. The PMF used to determine the shaping set from [Gül+20b], given in Table 2, is $P_A(a) = \{11/19, 7/19, 1/19, 0\}$.

1.6. Thresholds and Bounds

Analogous to the Shannon limit or channel capacity in Sec. 1.4, thresholds allow for determination of theoretical limitations of codes or other parts of communication systems. For example, density evolution (DE) (Sec. 2.5.1), is a well-known approach to find an optimal degree distribution of LDPC codes, based on an estimate of decoding thresholds. Bounds are similarly used to provide theoretical limitations to studied systems. A wide range of bounds exist in the field of coding theory, for example, the Singleton [Sin64], Hamming [Ham50] or Gilbert-Varshamov [Gil52], [Var57] bounds on distance properties of codes. Decoding bounds can be used to evaluate and compare performance of different codes, ensembles or constructions.

1.6.1. SNR Thresholds

To optimize the choice of modulation order, signal sets and other parameters for shaping to be used with the AWGN channel (Sec. 5), SNR thresholds are used. Assume an input signal set $\mathcal{M} = \{x\}$ with a finite alphabet $|\mathcal{M}| = M$ with a corresponding symmetric probability distribution $\mathbf{p}= (p_1, p_2, ..., p_M)$ of the signal points.

The output of the AWGN channel is a random variable Y with the conditional probability density function:

$$f_{Y|\mathcal{M}}(y|x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(y-x)^2}{2\sigma^2}}, \; x \in \mathcal{M},$$

where $\sigma = \sqrt{E_s/\mathrm{SNR}}$, $E_s = \sum_{x\in\mathcal{M}} x^2 p_x$ is the signal energy, p_x denotes the probability of x.

Definition 1.6.1 (Transmission rate)**.** The *transmission rate* R_T, in the context of using modulated signaling of order M to transmit messages encoded by a rate R_{ECC} code, is given as

$$R_T = R_{ECC} \log M \quad \text{bits per dimension.}$$

For shaped M-PAM signaling with a given transmission rate R_T, the SNR limit SNR_{SH} is determined as a solution to the equation with respect to SNR_{SH}

$$R_\mathrm{T} = \max_{\boldsymbol{p}} I(\mathcal{M}, \boldsymbol{p}, \mathrm{SNR}_\mathrm{SH}),$$

where

$$I(\mathcal{M}, \boldsymbol{p}, \mathrm{SNR}) = \sum_{x \in \mathcal{M}} \int_{-\infty}^{\infty} p_x f_{Y|X}(y|x) \log \frac{f_{Y|X}(y|x)}{f_Y(y)} \mathrm{d}y$$

$$= -\frac{\log e}{2} - \frac{2}{\sigma\sqrt{2\pi}} \int_0^{\infty} S(y) \log(S(y)) \mathrm{d}y,$$

$$S(y) = \sum_{x \in X_M} p_x \exp\left\{-\frac{(y-x)^2}{2\sigma^2}\right\}.$$

Here, $I(\cdot)$ is the mutual information between the input and output of the AWGN channel at a given SNR, using a modulation signal alphabet $\mathcal{M}$ with a probability distribution of signals defined by $\mathbf{p}$.

Unshaped M-PAM signaling thresholds SNR_U, with a given R_T, can analogously be obtained by solving the equation

$$R_\mathrm{T} = I(X_M, \boldsymbol{p}_\mathrm{uni}, \mathrm{SNR}_\mathrm{U}),$$

where the probability of the signal points X_M is uniform: $\boldsymbol{p}_\mathrm{uni} = (p_1, p_2, ..., p_M)$, $p_i = 1/M$, $i = 1, ..., M$.

The ultimate AWGN channel limits for a given R_T follows from the capacity given as Eq. (1.18), and can be obtained as

$$\mathrm{SNR}_\mathrm{AWGN}(R_\mathrm{T}) = 2^{2R_\mathrm{T}} - 1.$$

Table 3. Theoretical limits of achievable SNRs (in dB) for unshaped and shaped M-PAM signaling using probabilities p_i and the channel limits.

$R_\mathrm{T} = R \times p$	$\mathrm{SNR}_\mathrm{AWGN}$	M	Thresholds		p_i, i=1,2, ...,$M/2$
			SNR_U	SNR_SH	
1.5	8.45	4	9.30	8.87	0.35 0.15
		6	9.06	8.50	0.32 0.15 0.035
2.25	13.35	8	14.36	13.56	0.224 0.161 0.082 0.033
		10	14.29	13.41	0.207 0.155 0.088 0.037 0.013

Computed thresholds for two transmission rates R_T are given in Table 3 together with ultimate SNR limits SNR_{AWGN} for the channel and a given R_T. Notably, extending the M-PAM signal constellation from $M = 4$ to $M = 6$, combined with an optimal distribution $\boldsymbol{p}$, results in a gap of only 0.05 dB between SNR_{AWGN} and SNR_{SH} for $R_T = 1.5$. A similar gap of only 0.06 dB is obtained by extension to $M = 10$ at $R_T = 2.25$.

1.6.2. Bounds

In order to assess the quality of coded communication system without the need for experimental results by simulation, it is beneficial to have an alternate method of estimating its performance. Poltyrev's tangential-sphere (TS) bound from [Pol94] is considered one of the tightest bounds on decoding performance for codes with known weight enumerators. The TS bound gives an upper bound on the ML decoding error probability for the AWGN channel, based on the squared Euclidean distance (SED) spectrum (or SEDS) of the corresponding signal set

$$P_e \leq \sum_{w \leq w_0} A_w \Theta_w(x) + 1 - \chi^2_{n-1}\left(\frac{w_0}{\sigma^2}\right), \tag{1.20}$$

where

$$\Theta_w(x) = \int_{\sqrt{w/2}}^{\sqrt{w_0}} f\left(\frac{y}{\sigma}\right) \phi_{n-1}\left(\frac{w_0 - y^2}{\sigma^2}\right) \mathrm{d}y,$$

the Gaussian probability density function

$$f(x) = \frac{1}{\sqrt{2\pi}} \exp\left\{-\frac{x^2}{2}\right\},$$

and probability density and distribution functions of the χ-squared distribution with n degrees of freedom are

$$\phi_n(x) = \frac{x^{n/2-1} e^{-x/2}}{2^{n/2}\Gamma(n/2)},$$

$$\chi^2_n(x) = \frac{\gamma(n/2, x/2)}{\Gamma(n/2)}.$$

A_w is the w-th coefficient of the SED spectrum, and parameter w_0 is a solution of the equation

$$\sum_{w \leq w_0} A_w \int_0^{\arccos\sqrt{\frac{w}{4w_0}}} \sin^{n-3}\phi \mathrm{d}\phi = \sqrt{\pi}\frac{\Gamma\left(\frac{\mathrm{n}-2}{2}\right)}{\Gamma\left(\frac{\mathrm{n}-1}{2}\right)},$$

where $\Gamma(\cdot)$ is the gamma function.

The TS bound will later be applied to ensembles of random codes, based on expected values of their weight enumerators.

1.6.3. Code Ensembles and Generating Functions

Most commonly used bounds on ML decoding error probabilities over the AWGN channel, such as the Shannon bound, are not applicable to channel

models with higher order modulation due to the assumption that all codewords have the same energy. Most upper bounds require knowledge of the code weight enumerators (spectrum).

A trellis-based method for computing the distance spectrum of convolutional codes was introduced in [RC89]. In [Boc+04], a more efficient bidirectional tree search algorithm for computing the weight enumerators was presented. Quasi-cyclic (QC) codes can be obtained by a proper termination of convolutional codes. However, trellises of convolutional LDPC codes or their terminated versions have large memory, that makes computation of their enumerators infeasible for QC-LDPC codes of practical lengths. For this reason, similarly to [Gal62], random coding arguments are used in the thesis for analysis of ensembles of LDPC codes.

Code ensembles used for the theoretical analysis of LDPC codes were already introduced by Gallager in [Gal62], [Gal63], deriving the average over the code ensemble spectrum for a random ensemble of (J, K)-regular LDPC codes, for both the binary and NB cases. For the Gallager ensemble of binary (J, K)-regular LDPC codes, consider a parity-check matrix

$$H_b^{\mathrm{T}} = \left(H_{b,1}^{\mathrm{T}} \mid H_{b,2}^{\mathrm{T}} \mid \ldots \mid H_{b,J}^{\mathrm{T}}\right)^{\mathrm{T}}. \tag{1.21}$$

Each strip $H_{b,i}$ is of width $L_w = r/J$, with the first strip

$$H_{b,1} = \left(I_{L_w} \ \ldots \ I_{L_w}\right),$$

consisting of K identity matrices I_{L_w} of order L_w. Each strip $H_{b,i}, i > 1$, is a random permutation of $H_{b,1}$.

Definition 1.6.2 (Generating function)**.** Consider a sequence of numbers $(a_n)_{n\geq 0}$. The (ordinary) *generating function* associated with the sequence is the series

$$G(s) = \sum_{i=0}^{\infty} a_i s^i,$$

where s is a formal variable.

Definition 1.6.3 (Moment generating function)**.** For any random variable X possessing a moment generating function (MGF), the function is defined as

$$M(s) = \sum_{x \in X} p(x) s^x,$$

where $p(x)$ is the probability of x, $\sum p(x) = 1$.

Generating functions are a common mathematical tool for solving enumeration problems – determining the number of objects of a given size that satisfy a certain condition. The weight generating function of length-n binary sequences $\mathbf{x}$ with weight w satisfying the equality

$$\mathbf{x} H_i^{\mathrm{T}} = \mathbf{0} \tag{1.22}$$

is same for all strips $i = 1, \dots J$, computed as

$$G(s) = \sum_{w=0}^{n} N_{n,w} s^w = (g(s))^{L_w}, \tag{1.23}$$

where $N_{n,w}$ is the number of sequences $\mathbf{x}$ satisfying Eq. (1.22) and

$$g(s) = \sum_{i \text{ even}} \binom{K}{i} s^i = \frac{(1+s)^K + (1-s)^K}{2}, \tag{1.24}$$

is the weight generating function corresponding to binary sequences that satisfy the nonzero part of a single parity-check equation.

For each strip, the probability of Eq. (1.22) being satisfied by a random weight-w binary sequence $\mathbf{x}$ of length n is

$$p(w) = \frac{N_{n,w}}{\binom{n}{w}}. \tag{1.25}$$

Then, the probability of $\mathbf{x}$ satisfying Eq. (1.22) for all strips $i = 1, \dots, J$ is $(p(w))^J$, allowing for computation of average spectrum coefficients

$$E\{A_{n,w}\} = \binom{n}{w} (p(w))^J = \binom{n}{w}^{1-J} N_{n,w}^J, \tag{1.26}$$

where $E\{\cdot\}$ denotes expected value over the code ensemble.

Finally, the weight distribution of a random ensemble is represented by the weight generating function

$$G_n(s) = \sum_{w=0}^{n} A_{n,w} s^w. \tag{1.27}$$

As per [Boc+17], the generating function can be computed recursively as a series expansion. In general form, for generating functions $f(s) = \sum_{l \geq 0} f_l s^l$ and $F_L(s) = (f(s))^L = \sum_{l \geq 0} F_{l,L} s^l$, a recursive solution is given as

$$F_{l,L} = \begin{cases} f_l, & L = 1 \\ \sum_{i=0}^{l} f_i F_{l-i,L-1}, & L > 1 \end{cases}. \tag{1.28}$$

1.7. Outline and Contributions

The dissertation is organized as follows: Chapter 2 focuses on LDPC codes. In particular, binary LDPC codes and their quasi-cyclic variants are discussed in depth, including their design and construction. Sections 2.5 and 2.5.1 include an overview of algorithms and methods designed and used

for code construction. The chapter includes an overview of decoding algorithms commonly used for LDPC codes, as well as other background and preliminaries.

In Chapter 3, the study of LDPC codes is extended to generalized and non-binary variants, with a focus on code construction and theoretical performance comparison of the two code classes. Sections of the chapter were covered in

- "Irregular Generalized LDPC codes in Practical Communication Scenarios," 2022 IEEE Information Theory Workshop (ITW).

In Chapter 4 the topics of coded modulation and shaping are covered, predominantly from a perspective of theoretical analysis of the systems. This includes limits, bounds and comparisons to simulated performance analysis of combined coded modulation and shaping schemes, based on

- "Bound on the ML Decoding Error Probability for Coded QAM Signals with Shaping," 2023 IEEE International Symposium on Information Theory (ISIT).

Chapter 5 covers methods and algorithms for optimization of shaped coded modulation, based on principles covered in the preceding chapter. Parts of the chapter were covered in

- "Shaping for NB QC-LDPC Coded QAM Signals," 2023 12th International Symposium on Topics in Coding (ISTC),
- "Analysis of Coded Shaped QAM Signaling at Short and Moderate Lengths," 2024 IEEE International Symposium on Information Theory (ISIT).

2. LOW-DENSITY PARITY-CHECK CODES

Low-density parity-check (LDPC) codes are an important class of linear block codes, first introduced by R. G. Gallager in the 1960s [Gal62], [Gal63]. With a wide range of practical applications for both wired and wireless communication systems, LDPC codes have, for example, become a part of the standards in digital video broadcasting (DVB-S2) [EE09] and Wi-Fi 6 (802.11.ax) [EE21].

Defined by sparse parity-check matrices H, LDPC codes were originally defined by Gallager as (n, J, K) regular block codes of length n with J and K non-zero entries in rows and columns correspondingly. LDPC code parity-check matrices can be conveniently represented as bipartite graphs, known as Tanner graphs [Tan81]. Representing code columns and rows as variable and check nodes allows for graph-based code design, evaluation and optimization.

Making use of iterative decoding algorithms, such as belief propagation (BP), LDPC codes achieve highly competitive decoding error rates, outperforming other block codes under restrictions on decoding complexity. With BP-decoded LDPC codes of large lengths able to approach the Shannon limit, the attractiveness of this class of codes for modern communication standards is obvious.

In order to cover the topic of LDPC code design optimization, binary LDPC codes and basics of graph-based representations in Sections 2.1, 2.2 are covered to start with. Section 2.5.1 discusses the problem of searching for good binary LDPC codes, with quasi-cyclic LDPC (QC-LDPC) codes covered in Section 2.3, utilizing them as base matrices.

2.1. Description of Binary LDPC Codes

While initially, in [Gal62], Gallager introduced LDPC codes as (J, K)-regular sparse parity-check codes, with the main idea of achieving an exponential decrease in decoding error rates for a fixed J and rate R as n grows, constructions for LDPC codes were expanded in [Gal63]. These constructions also included non-binary (NB) LDPC codes, covered in Sec. 3.

Irregular graph-based codes were studied in [Lub+97], and irregular LDPC codes in [Lub+01] and [RSU01]. Irregularity with carefully chosen degree distributions, for example, by density evolution (Sec. 2.5.1), combined with highly optimized code design, allows for practical, near-capacity-achieving codes.

In [Tan81], Tanner proposed a recursive construction of long codes from subcodes based on bipartite graphs. These graphs have since then become known as Tanner graphs (Sec. 2.2). The graph representation of LDPC codes plays an important role in enabling low-complexity decoding of LDPC

codes. Tanner's work is particularly notable, as LDPC codes had been otherwise largely forgotten since their introduction in 1962. In the same vein, it took more than a decade and the "rediscovery" [MN96] of LDPC codes by MacKay and others (e.g. [Wib96]) before Tanner graphs became commonly used for design and analysis of LDPC codes.

Definition 2.1.1 (Regular LDPC codes)**.** A binary linear $[n, k]$-code determined by a parity-check matrix H is called (J, K)-regular LDPC code if each column of H contains J ones and each row contains K ones, with $J \ll k$ and $K \ll n$.

Compared to n and k, J and K are assumed to be small, remaining constant, even as n and k grow. The code rate of regular LDPC codes can be expressed as $R \geq 1 - J/K$ and the total number of nonzero elements in their parity-check matrices as $nJ = (n - k)K$.

Gallager's ensemble of random regular LDPC codes (Sec. 1.6.3) are commonly used to describe LDPC codes for theoretical analysis. Error performance of optimal decoding can be estimated based on the average weight distribution, represented via its weight generating function Eq. (1.27), where the average spectrum coefficients are given by Eq. (1.26).

Due to the sparsity of the parity-check matrices of LDPC codes, rows of H can be decoded as independent parity-check equations. As such, the decoding process of LDPC codes is parallelizable, which is a major upside of LDPC codes, compared to turbo codes.

In practice, LDPC decoding is performed iteratively, updating reliabilities (propagating belief) of decisions for each symbol recurrently. A more detailed description of BP decoding is given in Section 2.4.1.

2.2. Tanner Graphs

Definition 2.2.1 (Undirected graph)**.** An undirected graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ is a structure defined by vertices $\mathcal{V} = \{v_i\}$ and edges $\mathcal{E} = \{e_i\}$ connecting two vertices: $e_i = \{v_i, v_j\} = \{v_j, v_i\}$.

Definition 2.2.2 (Vertex degree)**.** The number of edges connected to a vertex determines the degree of the vertex.

Definition 2.2.3 (Bipartite graph)**.** A bipartite graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ is a graph defined by two disjoint sets of vertices $\mathcal{V} = \mathcal{V}_0 \cup \mathcal{V}_1, \mathcal{V}_0 \cap \mathcal{V}_1 = \varnothing$, such that no edge $e \in \mathcal{E}$ connects two vertices in one set: $\forall e = \{v_i, v_j\} \in \mathcal{E}, v_i \in \mathcal{V}_0, v_j \in \mathcal{V}_1$.

Definition 2.2.4 (Path)**.** A path (walk) p in a graph $\mathcal{G}$ is defined as an alternating sequence of vertices $v \in \mathcal{V}$ and edges $e \in \mathcal{E}$, such that any edge in the sequence has endpoints corresponding to the vertices it connects in the sequence: $p = \{v_1 \xrightarrow{e_1} v_2 \dots v_\ell \xrightarrow{e_\ell} v_{\ell+1}\} : e_i = \{v_i, v_{i+1}\}$. The length ℓ of a path is equal to the number of edges it contains.

Definition 2.2.5 (Cycle)**.** A path $\mathring{\mathbf{c}}$ that starts and ends at the same vertex ($v_1 = v_{\ell+1}$) is called a cycle.

Definition 2.2.6 (Simple cycle)**.** If a cycle only consists of distinct vertices (other than the start/end vertex v_1) and edges, it is called a simple cycle.

Definition 2.2.7 (Girth)**.** The girth g of a graph $\mathcal{G}$ is the length of the shortest simple cycle in $\mathcal{G}$.

Tanner graphs [Tan81] are bipartite graphs used to represent any rate $R = n/k$ linear block code parity-check matrix H in graph form. By denoting the set $\mathcal{V}_v$ as the n variable (symbol) nodes, representing columns of the parity-check matrix H and $\mathcal{V}_c$ as the $n-k$ check (constraint) nodes, representing rows of H, an edge $e = \{v_i, v_j\}$ thus translates to a non-zero entry in the corresponding parity-check matrix in the j-th row and i-th column. A (J, K)-regular block code is therefore represented by a Tanner graph with degree J variable and degree K check nodes.

As an example, consider the parity-check matrix H of the $[7, 4, 3]$ Hamming code and the corresponding Tanner graph in Fig. 11.

$$H = \begin{array}{c} \\ c_1 \\ c_2 \\ c_3 \end{array} \begin{array}{c} \begin{array}{ccccccc} v_1 & v_2 & v_3 & v_4 & v_5 & v_6 & v_7 \end{array} \\ \begin{pmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{pmatrix} \end{array}$$

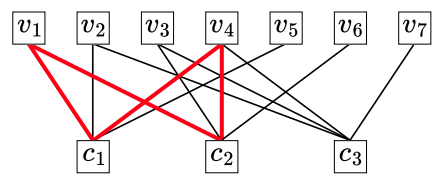

Figure 11. Tanner graph representation of the $[7, 4, 3]$ Hamming code

A common goal in LDPC code design is to avoid small Tanner graph cycles. In most instances, this refers to cycles of size 4, such as $\mathring{\mathbf{c}} = \{v_1 - c_1 - v_4 - c_2 - v_1\}$ highlighted in Fig. 11. As such, a common goal for most LDPC code designs is to achieve a Tanner graph girth of 6 or higher, or to maximize girth for a set of parameters.

There are multiple reasons why increased girth is associated with an improved decoding performance. It was shown by Tanner that girth determines a bound on the minimum distance of the code [Tan81]. However, the precise impact of minimum distance on the decoding performance is arguable [MMM04]. The minimum distance of a code as the only measure of code quality does not characterize achievable ML decoding performance.

Assuming that the Gilbert-Varshamov bound is tight, bounded distance decoding up to $d_{\min}/2$ does not allow to achieve channel capacity even for codes achieving this bound.

For iterative decoding, such as BP, the influence of the minimum distance of the code on decoding performance is even less explicit. However, cycles in the Tanner graph will result in dependence in the parity-check equations [RU01]. Thus, the presence of cycles, especially small ones, can, on average, only degrade the decoding error rates achieved by iterative decoding. It follows intuitively that a larger girth will guarantee a larger number of decoding iterations, where the independence assumption holds, thus improving achievable error rates.

Furthermore, trapping (and stopping [Di+02], [Tia+03]) sets, also called "near codewords", are structures in the graph representation of a code, which have a negative impact on error rates and the error floor region, in particular [Ric03].

Definition 2.2.8 (Stopping set)**.** A set S of variable nodes is called a stopping set if all its neighbors (check nodes) are connected to S at least twice.

Definition 2.2.9 (Trapping set)**.** An (a, b) trapping set T is a set of a variable nodes and b neighboring odd-degree check nodes in the subgraph induced by T.

Identification of trapping sets can be done by analysis of error events during simulation. A failure set $T(\mathbf{y})$ for a given input $\mathbf{y}$ is the set of bits not corrected by the ECC decoder. If $T(\mathbf{y}) \neq \varnothing$, it is thus identified as a trapping set.

It was shown in [Ric03] that only small trapping sets contribute to an error floor. An (a, b) trapping set in an $[n, k]$ code is called small if $a \leq \sqrt{n}$ and $b \leq 4a$ [LM05].

As demonstrated experimentally in [Ric03], while optimization of girth does offer a significant improvement compared to codes defined by random graphs, "neighbourhood-optimized" codes, minimizing the amount of trapping sets, exhibit a significantly reduced variance, and average improvement, of the achieved error floor regions, compared to optimization of girth alone.

For a set of a variable nodes, the extrinsic message degree (EMD) is the number of check nodes singly connected to the set. An upper bound for EMD can be obtained as the approximate cycle EMD (ACE), computed for a length $2a$ cycle $\mathring{\mathbf{c}}$ as $\sum_{i=1}^{a}(d_i - 2)$, where d_i is the degree of the i-th variable node in $\mathring{\mathbf{c}}$ [Tia+04]. ACE can be used in code construction to provide a statistical increase to the size of the smallest stopping set, thus improving achievable error rates using iterative decoding. As such, graph-based optimization of LDPC code design, allowing targeted exclusion of negatively impacting code structures, is an attractive choice for obtaining

codes with good performance.

While the list of properties of the designed code that needs to be optimized is large, it is also often conflicting. If one was to simply maximize girth, the obvious choice would be to only consider $(2, K)$-regular codes. However, this would result in a code with an ACE of zero for cycles, contributing to a suboptimal error floor. The proper prioritization of all these properties, as well as others not covered in this section, such as degree (weight) distributions and others, remains a highly non-trivial and open topic of research. Methods used to provide a solution to this problem, as used as part of the work within this dissertation, are covered in following sections.

2.3. Quasi-Cyclic LDPC codes

As practical applications require LDPC codes of large block lengths, structured codes that can be more easily stored and operated with become a necessity. Following from quasi-cyclic (QC) codes, introduced in [TW67], QC-LDPC codes result in long lengths with compact representations and low decoding complexity.

QC-LDPC codes are used in modern communication standards, such as IEEE Std 802.11n, as defined in [EE09]. Introduced for high-performance error correction, replacing convolutional codes, the standardized QC-LDPC codes include code rates $R \in \{1/2, 2/3, 3/4, 5/6\}$ and code lengths $n \in \{648, 1296, 1944\}$. For example, the proposed $[648, 324]$ code is defined by a base matrix H_b of size 24×12, with a column weight distribution $deg(2) = 11$, $deg(3) = 10$, $deg(12) = 3$ and a lifting factor of $\hat{M} = 27$. The descriptions of these constructions are covered in the following section.

2.3.1. Base and Degree Matrices

A common method of constructing QC-LDPC codes uses an all-ones base matrix. Consider a $(3, 6)$-regular matrix:

$$H_b = \begin{array}{c} \\ c_1 \\ c_2 \\ c_3 \end{array} \begin{array}{c} \begin{array}{cccccc} v_1 & v_2 & v_3 & v_4 & v_5 & v_6 \end{array} \\ \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} \end{array}. \tag{2.1}$$

Definition 2.3.1 (Circulant matrix)**.** A square matrix, in which each row is a cyclic shift of the preceding row, is called a circulant matrix.

Using circulant permutation matrices $\mathcal{P}$ of order $\hat{M}$, cyclic shifts are denoted as:

$$\mathcal{P}^0 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad \mathcal{P}^1 = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}, \quad \mathcal{P}^2 = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}. \tag{2.2}$$

The resulting lifted $[3\hat{M}, 6\hat{M}]$ QC-LDPC code is obtained as:

$$H(\mathcal{P}) = \begin{array}{c} \\ c_1 \\ c_2 \\ c_3 \end{array} \begin{array}{c} \begin{array}{cccc} v_1 & v_2 & ... & v_6 \end{array} \\ \begin{pmatrix} \mathcal{P}^{\mu_{11}} & \mathcal{P}^{\mu_{12}} & \dots & \mathcal{P}^{\mu_{16}} \\ \mathcal{P}^{\mu_{21}} & \mathcal{P}^{\mu_{22}} & \dots & \mathcal{P}^{\mu_{26}} \\ \mathcal{P}^{\mu_{31}} & \mathcal{P}^{\mu_{32}} & \dots & \mathcal{P}^{\mu_{36}} \end{pmatrix} \end{array}, \tag{2.3}$$

with $\mu_{ij} \in \{0, 1, \dots, \hat{M} - 1\}$.

Definition 2.3.2 (Degree matrix)**.** A matrix of lifting degrees W, where $W_{ij} = (\mu_{ij})$ is called an degree (or exponent) matrix.

It is clear that the small base matrices H_b have many small cycles in their graph representations, which would have a significant negative effect on decoding performance if not removed from their lifted counterparts. To produce QC-LDPC codes with good performance, degrees $\mu_{m,n}$ must be chosen such that small cycles in the base Tanner graph determined by H_b are eliminated.

A cycle $\mathring{\mathbf{c}} = \{v_1, ..., v_{2k}\}$ in the base Tanner graph, with corresponding degrees $\{\mu_1, ..., \mu_{2k}\}$, will remain ([Fos04]) in the resulting lifted QC-LDPC code only if

$$\sum_{i=1}^{k} (\mu_{2i-1} - \mu_{2i}) = 0 \mod \hat{M}. \tag{2.4}$$

Various different approaches of creating high-girth QC-LDPC codes have been developed. Algebraic methods of code construction provide compact representations of parity-check matrices while being relatively inflexible in their design, and are typically worse in terms of decoding error rates compared to computer search-based methods. Methods of QC-LDPC code design are covered in Sec. 2.5.

2.3.2. Encoding of QC-LDPC Codes

While LDPC codes can be encoded by multiplication with the generator matrix G, such an approach is impractical due to its complexity. Since G is not sparse, multiplication of $\mathbf{m}G = \mathbf{c}$ has complexity proportional to the number of non-zero elements in G, which scales quadratically with code length n.

Taking into account that communication standards use the QC-LDPC codes with specific structure of their parity-check matrices, as given below, the encoding scheme of this class of LDPC codes will be covered. To facilitate the encoding scheme, the rate $R = b/c$ parity-check matrices of QC-LDPC codes are required in (polynomial) form

$$H(D) = (H_{bd}(D), \mathbf{h}_0(D), H_{inf}(D)), \tag{2.5}$$

where $H_{bd}(D)$ is a bidiagonal matrix of size $(c - b) \times (c - b - 1)$

$$H_{bd}(D) = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ \vdots \\ (c-b-1) \\ (c-b) \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & \cdots & (c-b-1) \end{array} \\ \begin{pmatrix} D^0 & 0 & 0 & \cdots & 0 \\ D^0 & D^0 & 0 & \cdots & 0 \\ 0 & D^0 & D^0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & D^0 \\ 0 & 0 & 0 & \cdots & D^0 \end{pmatrix} \end{array}, \tag{2.6}$$

the column $\mathbf{h}_0$ consists of 3 non-zero elements,

$$\mathbf{h}_0^{\mathrm{T}} = (D^\alpha \dots D^0 \dots D^\alpha)^{\mathrm{T}}, \tag{2.7}$$

with $\alpha > 0$, and $H_{inf}(D)$ denotes the information part of the codeword and can be any appropriately sized monomial submatrix.

By substituting D with an $\hat{M} \times \hat{M}$ circulant matrix $\mathcal{P}$, the binary parity-check matrix of the $[c\hat{M}, b\hat{M}]$ QC-LDPC code is obtained. Consider an example with $b = 4$, $c = 8$:

$$H(\mathcal{P}) = \begin{pmatrix} \mathcal{P}^0 & \mathbf{0} & \mathbf{0} & \mathcal{P}^\alpha & \mathcal{P}^{\mu_{15}} & \mathcal{P}^{\mu_{16}} & \mathcal{P}^{\mu_{17}} & \mathcal{P}^{\mu_{18}} \\ \mathcal{P}^0 & \mathcal{P}^0 & \mathbf{0} & \mathcal{P}^0 & \mathcal{P}^{\mu_{25}} & \mathcal{P}^{\mu_{26}} & \mathcal{P}^{\mu_{27}} & \mathcal{P}^{\mu_{28}} \\ \mathbf{0} & \mathcal{P}^0 & \mathcal{P}^0 & \mathbf{0} & \mathcal{P}^{\mu_{35}} & \mathcal{P}^{\mu_{36}} & \mathcal{P}^{\mu_{37}} & \mathcal{P}^{\mu_{38}} \\ \mathbf{0} & \mathbf{0} & \mathcal{P}^0 & \mathcal{P}^\alpha & \mathcal{P}^{\mu_{45}} & \mathcal{P}^{\mu_{46}} & \mathcal{P}^{\mu_{47}} & \mathcal{P}^{\mu_{48}} \end{pmatrix}, \tag{2.8}$$

with $\mu_{ij} \in \{0, 1, \dots, \hat{M} - 1\}$.

A resulting codeword is then given as $\mathbf{c} = (\mathbf{c}_{check}\ \mathbf{c}_{inf})$, with the length $(c - b)\hat{M}$ $\mathbf{c}_{check}$ denoting the check part to be computed, and $\mathbf{c}_{inf} = (\mathbf{c}_{c-b+1} \dots \mathbf{c}_c)$ being the message sequence, given in b blocks of length $\hat{M}$.

For the two parts, it must hold that

$$\mathbf{c}_{check}\,(H_{bd}(\mathcal{P})\ \mathbf{h}_0(\mathcal{P}))^{\mathrm{T}} = \mathbf{c}_{inf} H_{inf}(\mathcal{P})^{\mathrm{T}}. \tag{2.9}$$

Denote the partial syndrome $\mathbf{s} = (\mathbf{s}_1\ \mathbf{s}_2 \dots \mathbf{s}_{c-b}) = \mathbf{c}_{inf} H_{inf}(\mathcal{P})^{\mathrm{T}}$, with all $\mathbf{s}_i$ denoting the $(c - b)$ blocks of length $\hat{M}$ corresponding to the information part of the codeword. Multiplying both sides of Eq. (2.9) by an all-one vector $\mathbf{e}^{\mathrm{T}} = (1\ 1 \dots 1)^{\mathrm{T}}$ of length $(c - b)\hat{M}$,

$$\mathbf{c}_{c-b}\left(\mathcal{P}^\alpha + \mathcal{P}^0 + \mathcal{P}^\alpha\right) = \mathbf{s}\mathbf{e}^{\mathrm{T}} \tag{2.10}$$

is obtained, as $H_{bd}(\mathcal{P})^{\mathrm{T}}\mathbf{e}^{\mathrm{T}} = \mathbf{0}$. Thus, the last block $\mathbf{c}_{c-b}$ of $\mathbf{c}_{check}$ (corresponding to the column $\mathbf{h}_0$) is obtained as

$$\mathbf{c}_{c-b} = \sum_{i=1}^{c-b} \mathbf{s}_i. \tag{2.11}$$

From this, all other blocks of $\mathbf{c}_{check}$ are obtained recursively from an updated syndrome

$$\hat{\mathbf{s}} = \mathbf{s} + \mathbf{c}_{c-b}\mathbf{h}_0(\mathcal{P}) = (\hat{\mathbf{s}}_1\ \hat{\mathbf{s}}_2 \dots \hat{\mathbf{s}}_{c-b}), \tag{2.12}$$

with

$$\begin{aligned} \mathbf{c}_1 &= \hat{\mathbf{s}}_1 \\ \mathbf{c}_i &= \hat{\mathbf{s}}_i + \mathbf{c}_{i-1}, \quad i \in \{2, 3, \dots, (c-b-1)\} \end{aligned}. \tag{2.13}$$

As mentioned in Sec. 1.3.4, the introduction of iterative decoding algorithms lead to practical codes able to approach the Shannon limit, triggering what could be considered a revolutionary step for communication systems. For LDPC codes, the iterative message-passing algorithm used for decoding is called the belief propagation (BP) algorithm. BP itself is an general algorithm that finds use in a range of applications, notably operating on graphical representations and models, such as Markov fields and Bayesian networks, widely used in machine learning and artificial intelligence.

As per design, the sparsity of the LDPC code parity-check matrix H results in its rows $\mathbf{h}_i$ being almost non-intersecting. This allows for them to be decoded independently as single parity-check codes, often referred to as constituent or row codes. In the decoding process, each code symbol belongs to multiple parity-check equations. Due to the independent decoding of row codes, each symbol "receives" multiple independent estimates on its value. This allows for calculation of reliabilities for the estimations of hard decisions, which can be used recurrently - as input of the next iteration of decoding the row codes of H.

2.4. Decoding of LDPC Codes

A parity-check matrix H of a $[n, k]$ binary LDPC code is a $m \times n$, $m = n - k$, linear system of m sparse parity-checks. Codewords $\mathbf{c}$ corresponding to the matrix H are a set of binary vectors of length n, which satisfy the m parity-check equations. The core building block of binary LDPC codes are the simplest of elementary codes - the single parity-check code. A binary *exclusive or* ($\oplus$) parity-check equation of n symbols $c_1, c_2, ..., c_n$ is satisfied only if $c_1 \oplus c_2 \oplus ... \oplus c_n = 0$.

Iterative decoding of LDPC codes was already proposed by Gallager in [Gal62]. Despite the linear complexity of his proposed decoding scheme, hardware of the period was insufficient to make LDPC codes practically usable. Despite that, the core principles of his proposed probabilistic decoding scheme, using a procedure to iteratively improve on decoding decisions, can be seen as having been "resurrected" with the turbo revolution of the 1990s.

2.4.1. Belief Propagation Decoding

An LDPC code can be seen as a concatenation determined by parity-check matrices $\mathbf{h}_j, j = 1, \ldots, n-k$, where $\mathbf{h}_j$ denotes the j-th row of H. By symbol-MAP decoding codes with parity-check matrix $\mathbf{h}_j$, the soft outputs of row decoders can be seen as being analogous to extrinsic information in Turbo decoding (Sec. 1.3.4).

Consider the parity-check matrix of a $[n, n-1]$ binary single parity-check code

$$H = [1, 1, \ldots, 1].$$

Let $\mathbf{y}$ be a received sequence over a discrete communication channel, encoded by the single parity-check code, and $P(x_i = c|\mathbf{y})$ for $c \in \{0, 1\}$ be the a posteriori probability of symbol x_i having value c in position i. Then, the probability of a decision in favor of symbol value $c \in \{0, 1\}$ in position i is computed by MAP decoder as

$$P(x_i = c|\mathbf{y}, \check{S}) = \begin{cases} \left(1 + \prod_{j=1, j\neq i}^{n} (1 - 2p_j)\right)/2, & c = 0 \\ \left(1 - \prod_{j=1, j\neq i}^{n} (1 - 2p_j)\right)/2, & c = 1 \end{cases}, \tag{2.14}$$

where $\check{S}$ is a binary random variable, denoting the event that the check is satisfied and p_j is the probability of one in position j. Computation of Eq. (2.14) is greatly simplified compared to BCJR decoding in Eq. (1.8) applied to a code trellis.

The main construction of BP decoding of LDPC codes then follows from Eq. (2.14) and the assumption of independence of parity-checks. Assume, for simplicity, a (J, K)-regular binary LDPC code with a parity-check matrix H is used to encode a message transmitted over the channel. Each row $\mathbf{h}_j$ of H can be considered a parity-check matrix of the $[K, (K-1)]$ linear single parity-check code. Let $\check{S}_j$ denote the event that the j-th check is satisfied. As all codewords have to satisfy all checks of H, denote the combined event of all checks $\check{S}_j$ containing the symbol x_i as $\check{S} = \bigcap_j \check{S}_j$. Then, for a received sequence $\mathbf{y}$, probabilities p_i of the values of the symbols x_i are expressed as

$$\frac{P(x_i = 0|\mathbf{y}, \check{S})}{P(x_i = 1|\mathbf{y}, \check{S})} = \frac{(1 - p_i)}{p_i} \prod_{j=1}^{J} \frac{1 + \prod_{h=1, h\neq i}^{K} (1 - 2p_{jh})}{1 - \prod_{h=1, h\neq i}^{K} (1 - 2p_{jh})}, \tag{2.15}$$

where $p_{ji} = P(x_i = c|\mathbf{y}, \check{S}_j)$ is the probability of value c in position i of the j-th check.

2.4.2. Sum-Product Algorithm

In practice, the assumption of independence of parity-check equations in the decoding process of LDPC codes is unrealistic. As mentioned, BP is an iterative message-passing algorithm. This lack of independence results in suboptimality of decoding performance due to correlation of the passed messages in the iterations. Tanner graph representations of LDPC codes allow for a more intuitive interpretation of the decoding process. A decoding iteration consists of two steps, called the horizontal and vertical step. During the horizontal step, messages are computed and sent from check nodes c_j to connected variable nodes v_i and vice versa during the vertical step.

More formally, after initialization $Q_{ij}^{(0)}(0) = 1 - p_j$ and $Q_{ij}^{(0)}(1) = p_j$ based on the received vector $\mathbf{y}$, the horizontal step of an iteration t computes

$$\begin{aligned} P_{ij}^{(t)}(0) &= \left(1 + \prod_{h \neq j} (1 - 2Q_{hi}^{(t-1)}(1))\right) /2 \\ P_{ij}^{(t)}(1) &= 1 - P_{ij}^{(t)}(0) \end{aligned} . \tag{2.16}$$

And, in the vertical step,

$$\begin{aligned} Q_{ji}^{(t)}(0) &= K_{ij}(1 - p_j) \prod_{k \neq i} P_{kj}^{(t)}(0) \\ Q_{ji}^{(t)}(1) &= K_{ij} p_j \prod_{k \neq i} P_{kj}^{(t)}(1) \end{aligned} , \tag{2.17}$$

is computed using normalization coefficients K_{ij}, chosen so that $Q_{ji}^{(t)}(0) + Q_{ji}^{(t)}(1) = 1$.

After an iteration, a hard decision can be made based on a posteriori probabilities

$$\begin{aligned} Q_{j}^{(t)}(0) &= K_{j}(1 - p_j) \prod_{k} P_{kj}^{(t)}(0) \\ Q_{j}^{(t)}(1) &= K_{j} p_j \prod_{k} P_{kj}^{(t)}(1) \end{aligned} , \tag{2.18}$$

as $\hat{x}_j = 1$ if $Q_j^{(t)}(1) > Q_j^{(t)}(0)$, and $\hat{x}_j = 0$ otherwise. This allows for syndrome $\mathbf{s}$ to be computed and, if found to be all-zero, decoding to be terminated. Otherwise, decoding continues up to a maximum of t_{max} iterations.

The precision and computational complexity required for this implementation of the sum-product algorithm (SPA) limits its use to only short LDPC codes. The complexity and precision issues can be solved by performing the computations in logarithmic domain.

Consider again the $1 \times k$ matrix H of a single parity-check code defined by a (J, K)-regular LDPC code and a sequence $\mathbf{y}$, received from a DMC. Denote p_i (Eq. (1.3)) as the a posteriori probability of the symbol in position i having value 1 based on $\mathbf{y}$. The LLRs are denoted as

$$\mathrm{LLR}(x_i) = l_i = \ln \frac{1 - p_i}{p_i} = \alpha_i \beta_i, \tag{2.19}$$

where α denotes the sign and β denotes the absolute value of the likelihood. Analogously, let $p_{ij} = \Pr(x_i = 1 | \mathbf{y}, \check{S}_j)$ be the probability with the addition of j-th check event $\check{S}_j$ being satisfied, and $\alpha_{ij}\beta_{ij} = \ln((1 - p_{ij})/p_{ij})$. A decoding iteration computes updated values $\alpha'_i \beta'_i$

$$\alpha'_i \beta'_i = \alpha_i \beta_i + \sum_{j=1}^{J} \left(\prod_{k=1, k \neq i}^{K} \alpha_{jk} \right) \gamma \left(\sum_{k=1, k \neq i}^{K} \gamma(\beta_{jk}) \right), \tag{2.20}$$

where

$$\gamma(\beta) = -\ln(\tanh(\beta/2)), \text{ for } \beta > 0. \tag{2.21}$$

The algorithmic form of the SPA in logarithmic domain for decoding binary LDPC codes is given in Alg. 1. To initialize, symbol LLRs l_j are computed based on the received sequence $\mathbf{y}$. By iterative updates to symbol LLRs, decoding proceeds either until a maximum t_{max} iterations is reached or an all-zero syndrome is achieved.

2.4.3. Practical BP Decoders

Due to their popularity, LDPC decoders have been an active area of research. Improvements to Gallager's original sum-product BP mostly focus on the MAP decoding of single parity-checks, with the iterative updates of Eq. (2.20) having the largest impact on overall computational complexity.

Denote variable and check nodes as $v \in \mathcal{V}_v$ and $c \in \mathcal{V}_c$ correspondingly, and input LLRs as L_v, initializing an array of messages L_{cv}. Using the γ-function, as in Eq. (2.20), the decoding steps are reformulated as
Horizontal step:

$$L_{cv} = \left(\prod_{v' \neq v} \mathrm{sign}(L_{cv'}) \right) \gamma \left(\sum_{v' \neq v} \gamma(L_{cv'}) \right),$$

Vertical step:

$$L_{cv} = L_v + \sum_{c' \neq c} L_{c'v},$$

allowing for improved memory requirements, reducing the amount of tentative (intermediate) LLRs stored to be equal to the number of nonzero

Algorithm 1: Log-domain SPA for decoding binary LDPC codes

$L_{ij}^{(0)} = l_j$; $t = 1$; // Initialize

while $t \leq t_{max}$ **do**

  for $i = 1$ **to** r **do** // Horizontal step

    $\alpha_{hi}^{(t-1)} = \text{sign}\left(L_{hi}^{t-1}\right)$;

    $Z_{ij}^{t} = \left(\prod_{h \neq j} \alpha_{hi}^{(t-1)}\right) \gamma\left(\sum_{h=1, h \neq j}^{K} \gamma\left(|L_{hi}^{(t-1)}|\right)\right)$;

  for $j = 1$ **to** n **do** // Vertical step

    $L_{ji}^{(t)} = l_j + \sum_{k=1, k \neq i}^{J} Z_{kj}^{(t)}$;

    $L_j = l_j + \sum_{k=1}^{J} Z_{kj}$;

    if $L_j < 0$ **then** $\hat{x}_j = 1$; // Hard descisions

    else $\hat{x}_j = 0$;

  $\mathbf{s} = \hat{\mathbf{x}} H^{\mathrm{T}}$;

  if $\mathbf{s} = \mathbf{0}$ **then return** $\hat{\mathbf{x}}$; // Syndrome check

  else $t = t + 1$;

return $\hat{\mathbf{x}}$;

elements in the parity-check matrix. Furthermore, as $\gamma(|x|)$ is monotonically decreasing and $\gamma(\gamma(|x|)) = |x|$, it follows that

$$\gamma\left(\sum_{v' \neq v} \gamma(|L_{cv'}|)\right) \leq \gamma\left(\max_{v' \neq v} \gamma(|L_{cv'}|)\right) = \min_{v' \neq v} |L_{cv'}|. \tag{2.22}$$

Min-Sum decoding, introduced in [FMI99], follows from Eq. (2.22), replacing the horizontal step with

$$L_{cv} = \left(\prod_{v'} \text{sign}(L_{cv'})\right) \min_{v' \neq v} |L_{cv'}|. \tag{2.23}$$

Based on [CF02a] and [CF02b], added scaling (normalization) and offset parameters, ς and o, respectively, improve the tightness of approximation of Eq. (2.23) as

$$L_{cv} = \left(\prod_{v'} \text{sign}(L_{cv'})\right) \left(\varsigma \min_{v' \neq v} |L_{cv'}| - o\right). \tag{2.24}$$

The efficiency of BP decoding can be improved by using a *layered* decoding schedule, proposed in [Hoc04], which reduces a decoding iteration to a single loop. Messages to symbol nodes are processed and updated immediately

after computation of parity checks, reducing memory requirements and resulting in faster convergence of decoding iterations.

The Min-Sum algorithm is specifically designed to reduce the hardware complexity issues of SPA decoding by reducing the number of multiplications required in a decoding iteration. A modified Min-Sum algorithm designed for the Field Programmable Gate Array (FPGA) is presented in [AKR19], with comparisons of various Min-Sum implementations from a hardware complexity perspective.

2.5. Design of LDPC Codes

The original LDPC codes proposed by Gallager were (J, K)-regular, constructed from J submatrices, the first of which containing ones in descending order, *i.e.,* i-th row containing ones in columns $(i-1)K+1$ to iK and the remaining submatrices obtained simply from column permutations of the first. Since the resurgence of LDPC codes, design of near-capacity achieving LDPC codes has greatly evolved, and continues to be an active area of research.

Among the most common methods of LDPC code construction, algebraic or combinatorial approaches are included. Popular algebraic approaches include constructions based on finite geometries ([KLF01], [Tan+05]). These constructions can have compact representations, simplified decoders and good girth of the corresponding Tanner graph. Gallager proposed ([Gal63] Appendix C) the use of permutation (not necessarily circulant) matrices in LDPC code construction. More recent designs, such as [VM04] use quasi-cyclic (QC) structures, similar to the self-orthogonal QC codes introduced in [TW67]. These constructions allow for greatly reduced complexity of implementation, compact representation and good Tanner graph girth.

Due to their popularity, the design of LDPC codes became an active area of study. While design methods vary greatly, they all aim to improve code performance by optimizing a number of common factors.

2.5.1. Density Evolution and EXIT Chart Based Approaches

It was shown in [Lub+97] that an upper bound exists on the fraction of errors, which BP decoding algorithms are able to correct based on degree distribution of a bipartite graph. As such, finding an optimal distributions for irregular parity-check matrices, defined by column and row weights, is paramount. One such technique, known as density evolution (DE), proposed in [RU01], aims to come to a threshold "convergence", approaching the achievable capacity for a set of channel parameters.

Assume an all-zero codeword is transmitted. By estimating the probability distribution of messages from a variable node to a check node, obtained by BP, graph-based codes can be designed to improve convergency of this

distribution to a degenerated distribution, assigning probability one to the all-zero codeword (*i.e.,* probability of error tends to zero). Denote the fractions of column and row weights, or equivalently, the degrees of variable and check nodes of the Tanner graph as

$$\lambda(x) = \sum_{i=2}^{d_v} \lambda_i x^{i-1} \text{ and } \rho(x) = \sum_{j=2}^{d_c} \rho_j x^{j-1}, \tag{2.25}$$

where d_v and d_c represent the maximum variable and check node degrees, correspondingly. The distributions $\lambda(x)$ and $\rho(x)$ determine a random ensemble of irregular LDPC codes. Discretized DE, proposed in [Chu+01], is an improved algorithm for DE, modeling discretized sum-product (SP) decoding.

Defining, for any PMF p,

$$\lambda(p) = \sum_{i=2}^{d_v} \lambda_i \underbrace{p * p \ldots * p}_{i-1}, \tag{2.26}$$

where $*$ is discrete convolution, and

$$\rho(p) = \sum_{j=2}^{d_c} \rho_j \underbrace{\mathcal{R}(p, \mathcal{R}(p, \mathcal{R}(\ldots) \ldots))}_{j-1 \text{ times}}, \tag{2.27}$$

where $\mathcal{R}$ is used as a nested, efficiently computable two-input operator used to compute the message update rule for check nodes, with

$$\mathcal{R}(a, b) = 2 \tanh^{-1} \left(\tanh \frac{a}{2} \tanh \frac{b}{2} \right). \tag{2.28}$$

Denoting as v the LLR of a message from a variable node with degree d_v to a check node, with

$$v = \sum_{i=0}^{d_v - 1} u_i, \tag{2.29}$$

where u_i, $i = 1, \ldots, d_v - 1$ are the incoming logarithmic likelihood ratio (LLR) from all neighboring check nodes (other than the check node corresponding to the message v) of the variable node, and u_0 denotes the LLR corresponding to the variable node output bit.

The construction in [Chu+01] reduces the the DE algorithm to

$$p_u^{(\mathcal{I}+1)} = \rho \left(p_{u_0} * \lambda(p_u^{(\mathcal{I})}) \right), \tag{2.30}$$

where $*$ is discrete convolution, $\mathcal{I}$ is the iteration number, and the initial PMF $p_u^{(0)}$ corresponds to the all-zero codeword. Notably, as all u_i are

assumed to be i.i.d, in Eq. (2.30), $p_u = p_{u_i}$, $i = 1, \ldots, d_v - 1$. The approach is covered in detail in [Chu00].

The algorithm is run iteratively for a fixed channel noise power until one of two outcomes occurs: either the density of u tends to infinity, meaning that the probability of error tends to zero, or it converges to a finite error probability. By progressively increasing the noise level, a threshold value is found based on the maximum noise level at which the error probability will tend to zero as the number of iterations tends to infinity.

Using linear programming, starting with a fixed $\rho(x)$ and an initial $\lambda(x)$, the degree distribution $\lambda(x)$ of variable nodes is incrementally improved based on the computed threshold. The process is repeated until a convergence of the distribution, using different initial distributions $\lambda(x)$. In [Chu00], the approach was used to construct codes of length 10^7 achieving a 0.04dB gap to the Shannon limit.

However, it should be noted that DE assumes an absence of short cycles and thus independence of messages in message-passing decoding. This assumption is unrealistic, considering the LDPC codes used in practical applications, such as the WiFi [EE21] and DVB-S2 [EE09] standards have girth $g \geq 6$.

As such, DE is not directly applicable to fixed (short or medium) length or higher rate codes, for which short cycles are inevitable, especially for parity-check matrix densities produced by the DE approach itself. As such, it is important to remember that DE is a method of asymptotic performance analysis, and that, while code density and weight distributions have great importance, they need to be considered jointly with other code metrics, such as the graph-based structures covered in Sec. 2.2.

Following the introduction of DE, there has been a significant amount of further related development. Notably, in [CRU01], the computationally expensive process of computing the PDFs of passed messages, required for calculation of DE thresholds can be replaced by approximation by Gaussian PDFs for the binary AWGN channel.

An approach based on extrinsic information transfer (EXIT) charts, introduced in [Bri01] for turbo codes, studies the information flow through iterative decoders. Used to predict the convergence of decoding iterations based on mutual information, and thus the decoding performance, it was used in [Bri00] to design serial-concatenated codes with good performance. Due to the similarities in decoding, EXIT charts can similarly be applied to the design of LDPC codes by treating variable and check node decoding as concatenated decoding.

To optimize decoding performance, the design of all types of LDPC codes used throughout the dissertation is done by optimization of the following parameters and metrics, as covered in Sec. 2.1, 2.2:

1. Column weight distribution

2. Girth, window girth and girth profile
3. ACE
4. Cycle multiplicity

2.5.2. Progressive Edge Growth

The progressive edge growth (PEG) algorithm was introduced in [HEA05] as a simple method of code construction with the goal of maximizing the girth of the Tanner graph. Since then, the algorithm has been improved and expanded on in numerous works, for example, in [XB04], [VS08], [SH15], [Dio+16], or [LK04], specifically for QC-LDPC codes, to the extent that it should be considered a family of "PEG-like" code construction algorithms.

The codes used throughout this dissertation are constructed either by the PEG algorithm, or by an algorithm introduced in [Boc+21], based on simulated annealing (SA). While the algorithm can be used to produce codes with excellent decoding performance, it is limited to the construction of codes with column weights 2 and 3 only. The general SA technique is a widely applicable probabilistic optimization algorithm that can be used to produce (near) optimal solutions in large search spaces, which also makes it well-suited for code search.

The PEG algorithm operates on the Tanner graph representation of the code, adding edges one-by-one to the graph, attempting to maximize the resulting girth for a given degree distribution. Iterating over variable (symbol) nodes, the algorithm adds edges to it until the required degree is achieved. Denoting the sets of Tanner graph nodes as $\mathcal{V}_c$ and $\mathcal{V}_v$, for check and variable nodes, correspondingly, for a node $v_i \in \mathcal{V}_v$, the first edge is chosen as (v_i, c_j), where $c_j \in \mathcal{V}_c$ is a check node with the lowest degree. Other edges are selected based on constructing a tree stemming from v_i. The tree is expanded until a layer l is reached, such that in layer $l+1$ every check node would be reached. Denoting the set of check nodes reached within l layers as $\mathcal{N}_{v_i}^l$, a random check node $c_j \in \mathcal{V}_c \setminus \mathcal{N}_{v_i}^l$ with the lowest degree is chosen, and the edge (v_i, c_j) is added to the graph.

In practice, as the algorithm has significant randomness and no guarantee of girth or any other property or parameter, large numbers of codes must be generated and analyzed, possibly by simulation. Thus, obtaining codes with near optimal performance still requires high overall complexity, despite the relative simplicity of the algorithm itself.

The average decoding performance of codes constructed by the algorithm can be significantly improved by making modifications to the original algorithm. While the changes increase the complexity of producing a single code, in practice, it simply shifts computationally expensive procedures to an earlier stage of the general process. Producing fewer codes that require evaluation by simulation is preferential, especially when the goal is to assess

and improve achieved error floors.

The PEG-like algorithm is applicable for construction of various types of LDPC codes, such as regular and irregular binary LDPC codes or base matrices of QC-LDPC or NB QC-LDPC codes.

To best illustrate the process of code construction using the proposed PEG-like algorithm, assume an example construction of a rate $R = 3/4$ code of length $n = 48$, working on a Tanner graph $\mathcal{G}$ with sets of variable and check nodes $\mathcal{V}_v, \mathcal{V}_c$, $|\mathcal{V}_v| = 48$ and $|\mathcal{V}_c| = 12$. Assume a variable node degree distribution $D_v = \{d_{v_i}\}, v_i \in \mathcal{V}_v$, ordered such that $\forall i, j \in \{1, \ldots, 48\}, i < j : d_{v_i} \leq d_{v_j}$. If the constructed matrix will serve as a base matrix of a (possibly NB) QC-LDPC code, $d_{v_{12}}$ would be assigned weight-3 instead, corresponding to the $\mathbf{h}_0$ column (see Sec. 2.3.2), to facilitate low-complexity encoding. The degree distribution can be divided into sections $D_{sec} = \{D_2, D_3, D_h\}$, for variable nodes of degree $2, 3$ and "high-degree" nodes, correspondingly.

Structure: To facilitate linear complexity encoding (Sec. 2.3.2), the constructed code must be partially structured. Thus, only the 36×12 H_{inf} part needs to be "filled". As adding the first edge to any v_i turns the variable node into a leaf, no cycles can be created during this step. Thus, as the first step, iterating over $v_i, i \in \{13, \ldots, 48\}$, an edge (v_i, c_j) is added to $\mathcal{G}$, keeping the distribution of check node degrees as uniform as possible. In the second iteration over variable nodes, due to the ordering of the degree distribution D_v, the weight-2 section $(v_i : d_{v_i} \in D_2)$ is completed first, while maximizing girth in the unfinished graph. As cycles created at this stage will consist of degree-2 nodes, with cycle ACE equal to zero, it is paramount that their size is maximized. Next, the weight-3 section D_3 is completed, while analyzing the cycles created in the process. The high-degree section D_h is filled last.

Cycles: While computationally expensive, any cycle of length ℓ, or smaller, created by adding an edge to a variable node v_i can be found by constructing a tree with $\ell/2$ layers spreading from v_i. When encountering a duplicate node when constructing the tree, the corresponding cycle can easily be found in a "meet in the middle" fashion, based on the path from the root to the duplicate nodes. When used to construct base matrices for QC-LDPC codes, the knowledge of cycles in the Tanner graph of the code is a requirement in the lifting process. Thus, the computationally complex process of finding cycles may as well be done in the code construction phase, where they can be used to make decisions about the edge selection process of the PEG approach. For a large cycle length ℓ, and large matrices, management of found cycles and efficient detection of duplicate cycles becomes non-trivial. A simple, but effective solution is to, firstly, keep separate data structures for each cycle size. Next, to facilitate detection of duplicates, cycles can be assigned an integer identifier based on the sum of the indices of edges participating in the cycle. While not necessarily unique,

only cycles with matching identifiers must be examined for duplicates.

Girth: As girth remains one of the primary indicators of code quality and created cycles in the edge adding process are easily identified, a girth requirement can be added to the input parameters of the PEG-like algorithm. If it is impossible to add an edge to a variable node without breaking the girth requirement, the algorithm can simply terminate and restart, either from the beginning, or from an intermediate point, for example, rebuilding the D_3-part, while preserving the lower-degree section. In addition, especially considering the high-degree section D_h, it can be desirable to set varying girth requirements for different sections of the constructed code.

Selective rebuilding: After construction of a code satisfying all initial parameters, identifying nodes with suboptimal characteristics, removing them and rebuilding the Tanner graph by only changing the selected nodes over a large number of iterations can yield in greater improvements to the specific characteristic than simply generating numerous codes and selecting the best one. Choosing characteristics, such as the average or mean ACE of participated cycles for each variable node, can result in notable decoding performance improvements. With this approach, removal of only a few problematic nodes is unlikely to result in significant improvement. Instead, at least $10 - 20\%$ of variable nodes should be chosen for removal and rebuilding. Alternatively, an entire section D_2, D_3 or D_h could be removed and reconstructed, attempting to improve some aspect of a code.

Improvements to the edge selection process in the algorithm can also significantly reduce the total complexity of the algorithm when using a girth requirement. After an edge (v_i, c_j) is added to the graph, the constructed tree spreading from v_i can be used to assess distances between other nodes. For example, if a check node c_j is found to have a distance of 5 to a check node v_i, adding an edge between the two creates a cycle of length 6. If the requirement is to have a girth larger than 6, c_j can be excluded from the set of possible candidates for edges added to v_i.

Various PEG-like constructions implement similar modifications and other improvements. In [VS08] the algorithm is used to construct codes with improved ACE, thus improving decoding performance compared to the original PEG algorithm. In [LK04] and [Lin+08], the algorithm is used to construct the base matrices of QC-LDPC codes and their liftings. In [Dio+15], a construction avoiding trapping sets is proposed.

A generalization of the PEG-like graph building algorithm is given as Alg. 2. In its inputs, R acts as a stand-in for a variety of conditions or parameters that the resulting code is required to satisfy in step 11 of Alg. 2 ($\mathit{eval}(\mathcal{G}', R)$), such as girth, cycle multiplicity, ACE or a combination of metrics. Similarly, the function $\mathit{best}(C_{v_i})$, choosing the best of the valid found edge candidates, can utilize a variety of methods to determine the best option. Alternatively, if complexity is prioritized, evaluation in step 11 can

Algorithm 2: PEG-like algorithm with requirements

```
1  Input: D_sec, R, I_max;
2  G(V = V_v ∪ V_c, E);                          // Initialize Tanner graph
3  for D_d ∈ D_sec do                            // Iterate over graph sections
4      I = 0;                                    // Rebuild counter
5      G̃ = G;                                    // Backup graph
6      while ∃v_i ∈ D_d : deg(v_i) < d  &  I < I_max do
7          for v_i ∈ D_d do                      // All v_nodes in section
8              C_{v_i} = {};                     // Valid edge candidates
9              for c_j ∈ candidates(v_i) do
10                 G' = G(V, E ∪ {v_i, c_j});
11                 if eval(G', R) then C_{v_i} = C_{v_i} ∪ {c_j, G'};
                   // Evaluate graph with added edge candidate
                      and store it if requirements are met
12             if C_{v_i} ≠ ∅ then
13                 c_j = best(C_{v_i});
                   // Choose best edge based on some criteria
14                 G = G(V, E ∪ {v_i, c_j});     // Add edge to graph
15                 update_candidates(G);
                   // Remove edge candidates based on added edge
16             else G = G̃; I = I + 1; go to 6;   // Restore from
                   backup, try rebuilding section from beginning
17 return G;
```

be performed until the first valid candidate is found and skipping to step 14. Selection of edge candidates always prioritizes check nodes with minimum degrees, but not as a strict requirement. The initial graph $\mathcal{G}$ may contain preexisting edges to provide structure or to facilitate selective rebuilding.

The described PEG-like algorithm has been used to construct codes used in this dissertation, most notably as base matrices of generalized, binary and non-binary QC-LDPC codes. In addition, due to the flexibility and ease of modification of the algorithm, it has proven valuable in various projects, allowing for extensive experimentation in LDPC code design.

2.5.3. Base Matrices of QC-LDPC Codes and their Labeling

Methods of QC-LDPC code construction can be grouped in two: (pseudo) random approaches and structured LDPC codes. Structured codes are commonly constructed based on finite fields, geometries or combinatorial structures (*e.g.,* [KLF01], [Tan+05]). While these methods can yield codes with good girth, they are generally much less flexible in their design and, other than a more compact representation, do not offer much practical benefit. As such, these methods are not studied in detail in this dissertation. Methods

of combining the structured and random-like approaches, *e.g.,* [TB21], aim to overcome these issues while offering lowered search complexity.

As a high-level description, the main steps in the design of QC-LDPC codes can be defined as:

- Choose column and row weights
- Construct the base matrix
- Construct the degree matrix

The choice of weight distribution was covered in Sec. 2.5.1, together with other code characteristics that are important for designing codes with good decoding performance. Good binary LDPC codes of appropriate size, constructed using these criteria and methods (Sec. 2.5), generally make for good base matrices of QC-LDPC codes.

Assuming an existing base matrix H_b and knowledge of a collection $\mathring{\mathbf{C}}_\ell$ of all cycles in H_b up to length ℓ, the lifting of QC-LDPC codes by carefully chosen edge labeling aims to maximize the Tanner graph girth of the constructed code. The size of the search space can be significantly reduced by assigning a lifting degree $\mu_i = 0$ for the first non-zero entry in each row and column. For example, assume that a base matrix has the form

$$H_b = \begin{array}{c} \\ c_1 \\ c_2 \\ c_3 \end{array} \begin{array}{c} \begin{array}{cccccc} v_1 & v_2 & v_3 & v_4 & v_5 & v_6 \end{array} \\ \begin{pmatrix} 1 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} \end{array}, \tag{2.31}$$

with the corresponding Tanner graph:

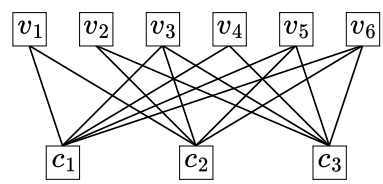

Figure 12. Base Tanner graph of H_b in Eq. (2.31)

The search space will be limited to assigning a total of eight values in the corresponding degree matrix

$$W = \begin{array}{c} \\ c_1 \\ c_2 \\ c_3 \end{array} \begin{array}{c} \begin{array}{cccccc} v_1 & v_2 & v_3 & v_4 & v_5 & v_6 \end{array} \\ \begin{pmatrix} \mu_1 & -1 & \mu_2 & \mu_3 & \mu_4 & \mu_5 \\ \mu_6 & \mu_7 & \mu_8 & -1 & \mu_9 & \mu_{10} \\ -1 & \mu_{11} & \mu_{12} & \mu_{13} & \mu_{14} & \mu_{15} \end{pmatrix} \end{array}, \tag{2.32}$$

where -1 corresponds to a zero entry in the base matrix, $\mu_i = 0$ for $i \in \{1, 2, \ldots, 7, 11\}$ and the remaining entries are non-zero. Examples of cycles can be visualized using the induced subgraph from v_1:

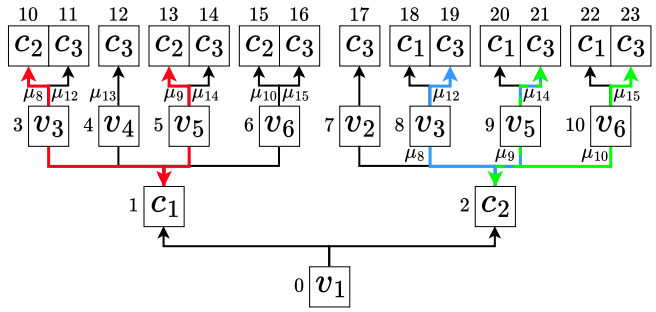

Figure 13. Edge-labeled subgraph of Fig. 12 spreading from a variable node v_1

In Fig. 13, only edges with non-assigned (non-zero) degrees are labeled. The direction of edges determines the sign of the degree for computation of Eq. (2.4). For example, the cycle

$$\mathring{\mathbf{c}}_1 = c_2 \xleftarrow{\mu_8} v_3 \xrightarrow{0} c_1 \xleftarrow{0} v_5 \xrightarrow{\mu_9} c_2,$$

highlighted in red, will only remain in the lifted code if $-\mu_8 + 0 - 0 + \mu_9 = 0 \mod \hat{M}$, where $\hat{M}$ is the lifting factor. In order to eliminate cycles when lifting the base matrix, a complete list of cycles, up to a specific length, in H_b has to be constructed. The j-th cycle $\mathring{\mathbf{c}}_j$, is represented as a length-N_{eq} equation (vector) form $\mathbf{a}_j$, where N_{eq} is the total number of non-zero entries in the base matrix H_b. An entry a_i of $\mathbf{a}_j$ is set to equal the number of times the i-th edge is traversed in $\mathring{\mathbf{c}}_j$, with the sign of a_i determined by the direction of movement through the edge, as before. For example, the equation $\mathbf{a}_1$, corresponding to $\mathring{\mathbf{c}}_1$ is given as

$$\begin{array}{c} \\ \mathbf{a}_1 = \end{array} \begin{array}{c} \begin{array}{ccccccccccccccc} \mu_1 & \mu_2 & \mu_3 & \mu_4 & \mu_5 & \mu_6 & \mu_7 & \mu_8 & \mu_9 & \mu_{10} & \mu_{11} & \mu_{12} & \mu_{13} & \mu_{14} & \mu_{15} \end{array} \\ \begin{pmatrix} 0 & 1 & 0 & -1 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}. \end{array} \tag{2.33}$$

Traversing every edge of $\mathring{\mathbf{c}}_1$ twice would, for example, result in a length-8 cycle, which, as an equation, would be represented by the vector with $a_4 = a_8 = -2$ and $a_2 = a_9 = 2$.

Representing lifting degrees in vector form, with ordering matching Eq. (2.33), as $\mathbf{w} = (\mu_j), j = 1, \ldots, N_{\text{eq}}$, checking if the cycle $\mathring{\mathbf{c}}_1$ remains after lifting, equivalent to Eq. (2.4), is given as $\mathbf{a}_1 \mathbf{w}^{\mathrm{T}} = 0 \mod \hat{M}$.

By constructing matrices A_ℓ, in which rows represent all equations $\mathbf{a}_i$ of cycles $\mathring{\mathbf{c}}_i$ of length ℓ in the base matrix H_b, the girth of the lifted QC-LDPC code can be checked as $A_\ell \mathbf{w}^{\mathrm{T}}$. Girth g is achieved if, for all $\ell < g$, all components of the resulting products are non-zero. For example, a section

of the A_4 matrix with rows corresponding to the highlighted cycles (blue $\mathring{\mathbf{c}}_2$, green $\mathring{\mathbf{c}}_3$) in Fig. 13:

$$A_4 = \begin{array}{c} \\ \mathbf{a}_1 \\ \mathbf{a}_2 \\ \mathbf{a}_3 \\ \vdots \end{array} \begin{array}{c} \begin{array}{ccccccccccccccc} \mu_1 & \mu_2 & \mu_3 & \mu_4 & \mu_5 & \mu_6 & \mu_7 & \mu_8 & \mu_9 & \mu_{10} & \mu_{11} & \mu_{12} & \mu_{13} & \mu_{14} & \mu_{15} \end{array} \\ \left(\begin{array}{ccccccccccccccc} 0 & 1 & 0 & -1 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 1 & -1 \\ & & \vdots & & & & & \vdots & & & & & \vdots & & \end{array} \right) \end{array}. \tag{2.34}$$

As can be seen from Fig. 13 and Eq. (2.34), many cycles partially overlap. Thus, in the straightforward multiplication by a degree vector $\mathbf{w}$, many duplicate operations would be performed. Operating over the constructed subgraphs (Fig. 13) instead allows for further simplification. Associating triples (t, e, v) with each node in the subgraph, where t denotes the serial number assigned to each node ($t \in \{0, \ldots, 23\}$ in Fig. 13), e is a signed degree index i of an edge connected to the node, where the sign indicates the direction of the edge ($+$ if the edge is "leaving" the node, $-$ otherwise) and v is the name of the node (v_i or c_j). Summation of assigned degrees from the root to each intermediate node allows for the checking of cycle elimination with reduced redundant computation compared to fully checking the system equations associated with all cycles.

By applying the described methods, an efficient algorithm can be obtained for lifting QC-LDPC codes. Combining a randomized greedy search algorithm with fast evaluation of remaining cycles, as in [BKJ16], these methods are used in construction of all types of QC-LDPC codes used in this dissertation.

3. NON-BINARY AND GENERALIZED LDPC CODES

Generalized LDPC (GLDPC) codes and non-binary (NB) LDPC codes over GF(2^m), $m > 1$ are two powerful generalizations of Gallager's binary (J, K)-regular LDPC codes. In the following sections, these two classes of codes are extensively covered, including code design and optimization, encoding and decoding. Lastly, GLDPC and NB LDPC codes are analyzed and compared using decoding bounds and through simulation.

3.1. Non-Binary LDPC Codes

NB LDPC codes over extensions of the Galois field started receiving attention after their rediscovery in [DM98]. While the class of codes was already introduced by Gallager in his original paper [Gal62], in which codes were defined over arbitrary fields, the term NB LDPC codes is now typically associated with *binary images* (Def. 3.1.3) of NB LDPC codes over extensions of the binary field GF(2^m). This slight abuse of notation will also hold true throughout this dissertation.

A big advantage of NB LDPC codes over their binary counterparts has been shown (e.g. [DCG04], [Pfl+09], [MTB15]) from their pairing with QAM signaling. A significant part of research in this area has been on the design and optimization of NB LDPC codes using columns of exactly two non-zero elements in the parity-check matrix. In this case, alphabets of GF(2^m), $m \geq 6$, are used. It is known by simulation that the BP decoding performance of short and medium length (typically with binary lengths of ≤ 2000) NB LDPC codes improves as m increases. However, the increase in decoding performance comes at a cost of increased decoding complexity, as m grows (see Sec. 3.1.3).

Constructions other than "ultra-sparse" NB LDPC codes were studied in [DM98], [HE04], [PFD08], and references therein. In combination with the study of irregular NB LDPC codes in [HEA05], [HZW08], [CDD12], it follows that using codes over small alphabets ($m < 6$) is preferable. Constructions of irregular NB LDPC codes over small alphabets were studied, *e.g.,* in [GSD10], [Ran+14], [Boc+21]. Due to the relatively low impact to the increase in decoding complexity, NB LDPC codes over small alphabets are an attractive choice for practical applications, especially when paired and jointly optimized with QAM signaling (Sec. 5).

3.1.1. NB QC-LDPC Code Description

To define NB QC-LDPC codes over GF(2^m), with $m > 1$, let $p(x) = p_0 + p_1 x + \cdots + x^m$ with $p_i \in \{0, 1\}$, $i = 0, \ldots m - 1$, be a primitive polynomial over GF(2^m) with a root α.

Definition 3.1.1 (Non-binary quasi-cyclic LDPC (NB QC-LDPC) code)**.** An $(\hat{M}c, \hat{M}b)$ NB QC-LDPC code over GF(q), $q = 2^m$, is defined by a $(c-b) \times c$ polynomial parity-check matrix

$$H(D) = \{\alpha_{ij} h_{ij}(D)\},$$

where the entry in position (i, j) is defined by $\alpha_{ij} \in \mathrm{GF}(q)$, $q = 2^m$, $i \in \{1, \ldots, (c-b)\}$, $j \in \{1, \ldots, c\}$, h_{ij} is either a monomial $D^{w_{ij}}$ or zero, with $w_{ij} \in \{0, 1, \ldots, \hat{M} - 1\}$, where $\hat{M}$ denotes the lifting factor (see Sec. 2.3). To obtain the corresponding q-ary NB QC-LDPC block code parity-check matrix, $D^{w_{ij}}$ is replaced by the w_{ij}-th power of a circulant permutation matrix $\mathcal{P}$ (Def. 2.3.1, *e.g.,* Eq. (2.2)) of order $\hat{M}$.

Definition 3.1.2 (Companion matrix)**.** For a monic polynomial $p(x) = p_0 + p_1 x + \cdots + x^m$, the companion matrix is an $m \times m$ matrix

$$C = \begin{pmatrix} 0 & 0 & \cdots & 0 & -p_0 \\ 1 & 0 & \cdots & 0 & -p_1 \\ 0 & 1 & \cdots & 0 & -p_2 \\ \vdots & & \ddots & & \vdots \\ 0 & 0 & \cdots & 1 & -p_{m-1} \end{pmatrix}.$$

Importantly, the set of matrices $\{C_i = C^i\} \cup \{[0]_{m \times m}\} \mod 2$, with $i \in \{0, 1, \ldots, q-1\}$, $q = 2^m - 1$, with corresponding addition and multiplication operations, forms a field isomorphic to GF(2^m) and columns of C_i are binary representations of powers of the root: $\alpha^i, \alpha^{i+1}, \ldots, \alpha^{i+m-1}$. For any pair C_i, C_j of matrices with $j > i$, the sets of their columns do not overlap in case of $\min\{j - i, i - j + 2^m - 1\} > m$.

Definition 3.1.3 (Binary image of a NB QC-LDPC code)**.** The binary form (image) of the q-ary NB QC-LDPC code is obtained by replacing the non-zero field elements of the parity-check matrix with the binary $(m \times m)$ companion matrices of the corresponding field elements, and zero elements with all-zero matrices of the same size.

Let $\boldsymbol{\alpha}_i = (\alpha_{i1}, \alpha_{i2}, ..., \alpha_{in_i})$ be a vector consisting of nonzero elements of the i-th row of the NB LDPC code nonbinary parity-check matrix, and let n_i be the number of nonzero elements of that row. After replacing these nonzero elements by their binary $m \times m$ companion matrices, an $m \times mn_i$ parity-check matrix of a binary linear code, called the i-th *constituent* code of the NB LDPC code is obtained. If the underlying LDPC codes for both constructions have J nonzero elements in each column and K nonzero elements in each row, the corresponding NB LDPC codes are called (J, K)-*regular*, otherwise the codes are called *irregular*.

Analogous to the binary case (Sec. 2.3.2), to ensure low complexity of encoding, NB QC-LDPC are restricted to the form

$$H(D) = \begin{pmatrix} H_{\mathrm{bd}}(D) & \mathbf{h}_0(D) & H_{\mathrm{inf}}(D) \end{pmatrix}, \tag{3.1}$$

with the bidiagonal part $H_{\mathrm{bd}}(D)$ of the same form as in Eq. (2.6), with the addition of field elements $\beta_i \in \mathrm{GF}(2^m)$ for non-zero entries

$$H_{\mathrm{bd}}(D) = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ 1 & \beta_2 & 0 & \cdots & 0 \\ 0 & \beta_2 & \beta_3 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \beta_{c-b-1} \\ 0 & 0 & 0 & \cdots & \beta_{c-b-1} \end{pmatrix}, \tag{3.2}$$

and the required form of the special column $\mathbf{h}_0(D) = (\gamma_0, 0, \ldots, 0, \gamma_1)^{\mathrm{T}}$, $\gamma_0 \neq \gamma_1$, with weight 2, differing from the binary case. As in Sec. 2.3.2, the information part H_{inf} can be any submatrix of the appropriate size.

For example, with $b = 4$, $c = 8$,

$$H(D) = \begin{pmatrix} D^0 & 0 & 0 & \gamma_0 D^0 & \alpha_{15} D^{\mu_{15}} & 0 & 0 & \alpha_{18} D^{\mu_{18}} \\ D^0 & \beta_2 D^0 & 0 & 0 & 0 & \alpha_{26} D^{\mu_{26}} & \alpha_{27} D^{\mu_{27}} & 0 \\ 0 & \beta_2 D^0 & \beta_3 D^0 & 0 & \alpha_{35} D^{\mu_{35}} & 0 & \alpha_{37} D^{\mu_{37}} & \alpha_{38} D^{\mu_{38}} \\ 0 & 0 & \beta_3 D^0 & \gamma_1 D^0 & 0 & \alpha_{46} D^{\mu_{46}} & \alpha_{47} D^{\mu_{47}} & \alpha_{48} D^{\mu_{48}} \end{pmatrix}, \tag{3.3}$$

where $\alpha_{ij}, \beta_j, \gamma_0, \gamma_1 \in \mathrm{GF}(2^m)$, $\alpha_{ij}, \beta_j, \gamma_0, \gamma_1 \neq 0$, and $\mu_{ij} \in \{0, 1, \ldots, \hat{M}-1\}$ for lifting factor $\hat{M}$.

3.1.2. Encoding of NB QC-LDPC Codes

The encoding process of NB QC-LDPC codes does not differ significantly from the binary variant, covered in Sec. 2.3.2, with the obvious exception of operations being performed over the code alphabet $\mathrm{GF}(2^m)$. Operating over a non-binary field also allows for a reduced form of the $\mathbf{h}_0$ column in Eq. (3.1), unlike the binary case.

As in Sec. 2.3.2, a codeword $\mathbf{u} = (\mathbf{c}\ \mathbf{m})$ consists of the check part $\mathbf{c}$ and information part $\mathbf{m}$. As $H\mathbf{u}^{\mathrm{T}} = \mathbf{0}^{\mathrm{T}}$, it follows that

$$[H_{\mathrm{bd}}\mathbf{h}_0]\mathbf{c}^{\mathrm{T}} = H_{\mathrm{inf}}\mathbf{m}^{\mathrm{T}}. \tag{3.4}$$

By multiplication with the all-one vector $\mathbf{e} = (1, \ldots, 1)$ of length $\hat{M}(c-b)$,

$$\mathbf{e}[H_{\mathrm{bd}}\mathbf{h}_0]\mathbf{c}^{\mathrm{T}} = \mathbf{c}_{c-b}\,(\gamma_0 + \gamma_1) = \mathbf{e}H_{\mathrm{inf}}\mathbf{m}^{\mathrm{T}}, \tag{3.5}$$

is obtained. Differing from the binary case, in Eq. (2.10), the non-binary case allows for a reduced form (weight two) column $\mathbf{h}_0$ with the condition $\gamma_0 \neq \gamma_1$. The partial syndrome $\mathbf{s} = (\mathbf{s}_1, \ldots, \mathbf{s}_{c-b})$ is obtained as

$$\mathbf{s}^{\mathrm{T}} = H_{\mathrm{inf}}\mathbf{m}^{\mathrm{T}}. \tag{3.6}$$

The length-$\hat{M}$ block of parity-check symbols, corresponding to the special $\mathbf{h}_0$ column is then obtained as

$$\mathbf{c}_{c-b} = (\gamma_0 + \gamma_1)^{-1} \sum_{i=1}^{c-b} \mathbf{s}_i, \tag{3.7}$$

operating over GF(2^m). Partial syndrome is then updated for the first and last symbol

$$\mathbf{s}_1 = \mathbf{s}_1 + \mathbf{c}_{c-b}\gamma_0. \tag{3.8}$$

The remaining parity-check symbol blocks are obtained as

$$\begin{aligned} \mathbf{c}_1 &= \mathbf{s}_1 \\ \mathbf{c}_i &= \beta_i^{-1} \sum_{j=1}^{i} \mathbf{s}_j \text{ for } i \in \{2, \ldots, (c-b-1)\} \end{aligned}. \tag{3.9}$$

3.1.3. Decoding of NB QC-LDPC Codes

Compared to the binary case, BP decoding of NB LDPC codes follows the same ideas with the exception of operating over a non-binary field. Denoting the sequence of received channel symbols as y_i, $i = 1, \ldots, n$, using BPSK signaling for simplicity, sets of symbols and checks are defined as

- $\mathcal{N}(j)$, $j \in \{1, \ldots, r\}$ symbols y_i participating in the j-th check,
- $\mathcal{M}(i)$, $i \in \{1, \ldots, n\}$ set of checks dependent on symbol y_i.

For a code symbol $\beta \in$ GF(2^m), probability of $y_i = \beta$, given extrinsic information from all checks other than the j-th, is denoted as Q_{ji}^{β}. The probability of the j-th check being satisfied is denoted as R_{ji}^{β}, assuming $y_i = \beta$ and that other symbols of the j-th check have probabilities Q_{jk}^{β}, where $k \in \mathcal{N}(j)$, $k \neq i$ and $\beta \in$ GF(2^m). Then, the generalized sum-product algorithm is given in 4 steps:

Firstly, initialization uses channel likelihoods L_i^{β} of $y_i = \beta$ as $Q_{ji}^{\beta} = L_i^{\beta}$, where $i \in \{1, 2, \ldots, n\}$, $j \in \{1, 2, \ldots, r\}$.

Next, in the two message passing steps, first R_{ji}^{β} is updated for all checks $j \in \{1, 2, \ldots, r\}$ as the horizontal step, summing over all symbol vectors $\mathbf{y}$ that satisfy the j-th check and have the i-th symbol $y_i = \beta$:

$$R_{ji}^{\beta} = \sum_{\mathbf{y}:\ \mathbf{y}\mathbf{h}_j^{\mathrm{T}}=0, y_i=\beta} \prod_{k \in \mathcal{N}(j), k \neq i} Q_{jk}^{y_k}, \tag{3.10}$$

where $\mathbf{h}_j^{\mathrm{T}}$ denotes the j-th column of the code parity-check matrix H. Then, the vertical step updates Q_{ji}^{β} for all symbols, $i \in \{1, 2, \ldots, n\}$ using a normalization factor $A_{ji} \in \mathbb{R}$ to ensure that $\sum_{\beta=0}^{2^m-1} Q_{ji}^{\beta} = 1$:

$$Q_{ji}^{\beta} = A_{ji} L_i^{\beta} \prod_{l \in \mathcal{M}(i), l \neq j} R_{il}^{\beta}. \tag{3.11}$$

Finally, as the fourth step, the intermediate solution is obtained as:

$$\hat{y}_i = \arg\max_{\beta}\{L_i^{\beta} \prod_{l \in \mathcal{M}(i)} R_{il}^{\beta}\}. \tag{3.12}$$

The algorithm continues iteratively until either a maximum number of iterations t_{max} is reached or a codeword $\hat{\mathbf{y}}$ is obtained: $\hat{\mathbf{y}} H^{\mathrm{T}} = 0$.

It should be noted that, similar to the decoding of binary LDPC codes, the step with the highest computational complexity is the MAP decoding of a NB single parity-check code - the update of R_{ji}^{β} for all checks in Eq. (3.10). Typically, it is implemented by using a full trellis (e.g. Fig. 14) of 2^m states with 2^m branches (for a code over GF(2^m)) both entering and leaving each state, on which BCJR-like decoding is performed as

$$R_{ji}^{\beta} = \sum_{\alpha,p:\ \alpha+p=\beta h_{ji}^{\mathrm{T}}} \Pr(\sigma_{j(i-i)} = \alpha) \Pr(\rho_{j(i+1)} = p), \tag{3.13}$$

where

$$\begin{aligned}
\rho_{ji} &= \sum_{k, i \in \mathcal{N}(j),\ k:\ k \geq i}^{i} y_k h_{kj}^{\mathrm{T}}, \\
\Pr(\rho_{ji} = \beta) &= \sum_{\alpha,p:\ p h_{ji}^{\mathrm{T}} + \alpha = \beta} \Pr(\rho_{jk} = \alpha) Q_{ki}^{p}, \\
\sigma_{ji} &= \sum_{k, i \in \mathcal{N}(j),\ k:\ k \leq i}^{i} y_k h_{kj}^{\mathrm{T}}, \\
\Pr(\sigma_{ji} = \beta) &= \sum_{\alpha,p:\ p h_{ji}^{\mathrm{T}} + \alpha = \beta} \Pr(\sigma_{jk} = \alpha) Q_{ki}^{p}.
\end{aligned} \tag{3.14}$$

The resulting computational complexity of the step, as given in Eq. (3.13), is $O(2^{2m})$, which would be prohibitive for most practical applications. In the case of binary LDPC codes Eq. (2.19), (2.20), complexity reduction was achieved by operating in the logarithmic domain. In the case of NB LDPC codes, this effect is achieved by using the Hadamard domain and the fast Hadamard transform (FHT). Based on [BD03], the order of complexity of the step is reduced to $O(q \log q)$ for codes over GF(q).

For a trellis section (Fig. 14), denote the set of all transitions between a trellis state pair (m', m) as $S_t(c)$ for a time moment t and symbol c. Performing symbol-MAP decoding (Sec. 1.3.2) on the trellis, the a posteriori probability of a symbol c in position t is computed according to Eq. (1.6), based on transition probabilities $\sigma_t(m', m)$, as in Eq. (1.9). Decoding then consists of the forward pass Eq. (1.11), backward pass Eq. (1.12) and computation of LLRs Eq. (1.13).

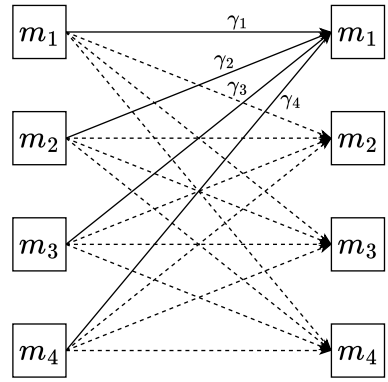

Figure 14. One level of a full trellis for codes over GF(4)

By construction of a matrix $\mathbf{\Gamma}_t$ from all transition probabilities of $m' \to m$ (probabilities of all length-m binary vectors) as

$$\mathbf{\Gamma}_t = \{\gamma_t(m', m)\},\ m, m' \in \{0, \ldots, q-1\}, \tag{3.15}$$

computations of Eq. (1.11) and (1.12) can be expressed in matrix form:

$$\boldsymbol{\alpha}_t = \boldsymbol{\alpha}_{t-1}\mathbf{\Gamma}_t; \quad \boldsymbol{\beta}_t = \mathbf{\Gamma}_{t+1}\boldsymbol{\beta}_{t+1}. \tag{3.16}$$

By constructing a transition matrix $\tilde{\mathbf{\Gamma}}_t$ corresponding to $\alpha_t = 1$ for both m and m', where rows are given by cyclic shifts of a standard vector

$$\tilde{\boldsymbol{\gamma}}_t = (\gamma_t(0,0), \gamma_t(0,1), \ldots, \gamma_t(0,q-1)), \tag{3.17}$$

any $\mathbf{\Gamma}_t$ corresponding to an arbitrary $\alpha_t \neq 0$ can be obtained as

$$\mathbf{\Gamma}_t = \tilde{\mathbf{\Gamma}}_t \mathbf{\Pi}_t, \tag{3.18}$$

where $\mathbf{\Pi}_t$ is a permutation matrix determined by the element α_t. This allows for efficient computation of

$$\boldsymbol{\alpha}_t = \mathrm{IFHT}\left(\mathrm{FHT}\left(\boldsymbol{\alpha}_t\right) .* \left(\tilde{\boldsymbol{\gamma}}_t \mathbf{\Pi}_t\right)\right), \tag{3.19}$$

and $\boldsymbol{\beta}_t$ analogously. Here, $.*$ denotes component-wise multiplication, FHT and IFHT denote the fast Hadamard transform, and its inverse, correspondingly. The Hadamard transform is defined by its recursive structure of order $q = 2^m$:

$$H_m = \frac{1}{\sqrt{2}} \begin{pmatrix} H_{m-1} & H_{m-1} \\ H_{m-1} & -H_{m-1} \end{pmatrix}, \tag{3.20}$$

or $H_m = H_1 \otimes H_{m-1}$, where $\otimes$ denotes the Kronecker product and

$$H_1 = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}. \tag{3.21}$$

The FHT algorithm is a divide-and-conquer implementation of the transform, achieving $O(q \log q)$ complexity.

3.1.4. Design of NB QC-LDPC Codes

It was shown in [LFK09], based on density evolution and simulation, that, for small alphabets $\mathrm{GF}(2^m)$, $m = 4, 5, 6$, the optimal average column weight of the parity-check matrix is in the interval $[2.2, 2.4]$. Furthermore, it was concluded that due to codes over $\mathrm{GF}(2^6)$ achieving a gap of only 0.06 dB from the Shannon limit for the binary AWGN channel, there is little incentive to even consider codes over larger alphabets. Throughout this dissertation, the used NB codes are irregular NB QC-LDPC codes over small ($m \leq 6$) alphabets.

The overarching design principles follow from the constructions used for binary QC-LDPC codes. Beginning with the construction of the base matrix, requiring a target degree distribution, the maximization of girth and optimization of other parameters, as described in Sec. 2.5.1, for example, methods from Sec. 2.5 can be used, followed by lifting of the base matrix, as for the binary case, in Sec. 2.5.3. The final step, assigning powers of a primitive element in the field $\beta \in \mathrm{GF}(q)$ to non-zero elements of the parity-check matrix is procedurally analogous to the lifting process, further eliminating cycles present in the base matrix according to Eq. 3.22.

Base matrices constructed by simulated annealing, as in [Boc+21], consisting of columns of weight $2, 3$ only, closely match the optimal average column weight given by density evolution. While these codes achieve near-capacity decoding performance, they exhibit notable error floors. The addition of a small number (depending on code length and rate) of high-weight columns results in improved error floors at the cost of a slightly increased gap to capacity. Methods of determining degree distributions from [BKJ16], applicable to a range of LDPC codes, including NB variants, offer a finite-length analysis, taking into account cycles in constructed codes. Based on an asymptotic analysis (such as DE) alone, a large number of high-weight columns leads to improved decoding FER performance while, in reality, creating many small cycles in the graph representation of the code. Well-optimized codes are able to mostly alleviate the negative impacts of the added high-weight columns. Ultimately, a singular approach to the design of base matrices remains elusive, requiring experimentation with multiple design approaches, to tailor codes to any specific application.

Conversely, lifting and assigning degrees to the constructed base matrix is a relatively straightforward process. Random labeling can be used in combination with efficient evaluation, as in Sec. 2.5.3, maximizing girth. As according to Eq. (2.4), a length-ℓ cycle with labels μ_i remains in the lifted graph only if $\sum_{i=1}^{\ell/2} \mu_{2i} = \sum_{i=1}^{\ell/2} \mu_{2i-1} \mod \hat{M}$, a cycle in the NB QC-LDPC

code with assigned degrees $\beta_i \in \mathrm{GF}(q)$ is canceled if

$$\prod_{i=1}^{\ell/2} \beta_{2i} \neq \prod_{i=1}^{\ell/2} \beta_{2i-1}. \tag{3.22}$$

Defined in [PFD06] as the *full rank condition*, an algorithmic approach to the cancellation of cycles by the condition is analogous to that applied in lifting (Sec. 2.5.3).

The resulting decoding performance of the described NB QC-LDPC code construction is presented by the simulation results in Fig. 18, using BPSK modulation, and throughout Chapters 4 and 5, using higher-order modulation, with and without coded modulation and shaping.

3.2. Generalized LDPC Codes

Generalized LDPC (GLDPC) codes stem from Tanner's paper [Tan81], and are therefore sometimes referred to as generalized low density (GLD) Tanner codes. Originally, GLDPC codes were targeted at the low-rate range since this construction requires additional parity checks to be introduced in the parity-check matrix of the underlying LDPC code. In other words, nonzero elements of the base parity-check matrix are replaced with columns of the parity-check matrix of a constituent code.

This idea was extensively studied in [LZ99] and [BPZ99], where regular GLDPC codes were introduced. It was shown that Hamming codes are among the most promising candidates for constituent codes and maintain a relatively high rate of the resulting GLDPC code construction. The random ensembles of GLDPC codes obtained from Gallager's ensemble of binary LDPC codes by replacing single parity-check codes with Hamming codes were analyzed. It was proven that there exist such GLDPC codes for which the asymptotic minimum distance of GLDPC codes grows linearly with length. An upper bound on the ML decoding error probability for the ensemble of GLDPC codes with column Hamming weight $J = 2$ was derived in [BPZ99]. Simulation results presented for the rate $R \approx 1/2$ GLDPC codes showed some coding gain over short binary LDPC and turbo codes. However, these GLDPC codes did not achieve the same bit error rate (BER) performance as long turbo codes.

Low-rate irregular GLDPC codes of short length with Hamming constituent codes were designed in [LR05]. A number of papers have studied both structured and unstructured regular, as well as irregular GLDPC codes of different rates based on different constituent codes (see *e.g.,* [DMV05; YPW07; LRC08; DW14; LOM18; ZLS21; ADR11] and references therein). It was shown in [YPW07] that extremely long (of length ($n > 10^6$)) and extremely low-rate codes are about 0.25 dB from the BICM Shannon limit.

In [DMV05], GLDPC codes were studied as candidates for high-speed optical communications. Weight distributions for the ensembles of irregular GLDPC codes were analyzed in [PCF08].

In the following sections, as originally covered in [BKM22], the focus is on the study of high-rate ($R > 1/2$) irregular GLDPC codes. The studied codes are restricted to constructions with low complexity of decoding, meaning a limit of 3 or 4 redundant symbols for the constituent codes used for GLDPC code construction and small alphabets of the NB LDPC codes used in performance comparisons.

Compared to prior works in the area, the proposed and studied approach differs by an increased focus on the optimization of underlying binary LDPC codes used as base matrices for construction of the GLDPC codes. Combined with optimized matchings of the non-zero elements of the base matrix with columns of the constituent codes with a minimum distance of two, similar to methods employed for design of NB LDPC codes (e.g. [Boc+22]), GLDPC codes with significantly improved BP decoding performance are obtained, at rates desirable for practical communication scenarios, while also achieving relatively low decoding complexity of MAP decoding of the constituent codes.

In the following sections, the two competiting constructions of low-complexity GLDPC and NB LDPC codes are compared.

3.2.1. GLDPC Description

A binary GLDPC code is determined by an $r_\mathrm{b} \times n$ sparse parity-check matrix H_b of the underlying binary LDPC code and the parity check matrix of the constituent code H_c of size $r_\mathrm{c} \times n_c$, $r_\mathrm{b} r_\mathrm{c} < n$. For simplicity, in the constructed code examples it is assumed that H_b is row-regular with Hamming weight of each row equal to n_c. The weights of columns can vary. The Hamming weight of the i-th column is denoted as J_i and the average column weight as:

$$J_\mathrm{av} = \frac{\sum_{i=1}^{n} J_i}{n}.$$

The binary $r_\mathrm{c} r_\mathrm{b} \times n$ parity-check matrix H of the row-regular GLDPC code is obtained by replacing zero elements in H_b by all-zero columns of length r_c and nonzero elements of rows of H_b by different columns of H_c.

Notice that for a row-regular base matrix, the average column weight can be computed as a ratio of the total number of nonzero elements $n_\mathrm{c} r_\mathrm{b}$ to the code length, *i.e.,* $J_\mathrm{av} = n_\mathrm{c} r_\mathrm{b}/n$. Taking into account that the number of redundant symbols of the GLDPC code is $r_\mathrm{c} r_\mathrm{b} = J_\mathrm{av} r_\mathrm{c} n/n_\mathrm{c}$, the rate R of the GLDPC code can be computed as:

$$R = 1 - \frac{r_\mathrm{c} r_\mathrm{b}}{n} = 1 - \frac{J_\mathrm{av} r_\mathrm{c}}{n_\mathrm{c}} = 1 - J_\mathrm{av}(1 - R_\mathrm{c}), \tag{3.23}$$

where $R_c = 1 - r_c/n_c$ denotes the rate of the constituent code.

In terms of the decoding complexity, the number of rows r_c of the constituent code parity-check matrix of GLDPC codes plays the same role as $m = \log_2 q$ for NB LDPC codes, where q is the alphabet size. In general, the complexity of the generalized BP decoding per bit for a binary image of the NB LDPC code is proportional to $q^2 = 2^{2m}$. Implementation in the fast Hadamard transform domain has complexity proportional to $m2^m$ (see *e.g.,* [DF07]) but requires higher arithmetic precision. Further simplifications in [ZC10] are achieved at the cost of a loss of 0.2 - 0.4 dB in the decoding performance. The complexity per bit of the trellis-based MAP decoding of the constituent code is of order 2^{r_c} in the case of GLDPC codes.

A precise comparison of decoding complexities for the two classes of codes is complex. If $r_c = m$, the per-bit decoding complexity is of the same order of magnitude, but NB LDPC codes require high-precision arithmetic, whereas the trellis representation of GLDPC codes can be more complex depending on the constituent codes used in their construction. A conclusive complexity analysis was not performed as part of this dissertation, although it was empirically evident from the work done that the overall decoding complexity for the two classes was close using matching parameters.

3.2.2. GLDPC Code Design

Design of GLDPC codes is performed in two steps, starting with the construction of a base matrix, following the descriptions and design principles from Sec. 2.31, also analogous to the design of the base matrices used for NB LDPC codes. Then, columns of a constituent code parity-check matrix are assigned to nonzero elements of the base matrix. As such, the following sections will only focus on the choice and construction of the used constitutent codes and the process of labeling the entries of the base matrix with columns of the constituent codes.

3.2.3. Constituent Codes

A common challenge with GLDPC codes for practical communication systems is to achieve a high enough code rate. Thus, while constituent codes, such as Hamming and BCH codes, can be used to achieve excellent decoding performance of the resulting GLDPC codes, especially at higher signal-to-noise ratios, simpler constructions are required to achieve more desirable rates.

Assuming a row-regular base matrix H_b with row weight n_c, a constituent code of a matching length is required, with the number of rows r_c dictated by the rate requirement of the resulting GLDPC code. As it was described earlier, constituent codes with a minimum distance equal to two are used in this construction. In order to reduce the number of weight-two codewords,

firstly all possible length-r_c non-zero vectors are chosen as columns of $H_{c;r_c\times n_c}$ and then repeated, each with as uniform number of repetitions as possible, until a desired constituent code rate is achieved. Parity-check matrices of two constituent codes used for simulations in Fig. 18 ($J_{\text{av}} = 2, n_c = 32$ and $J_{\text{av}} = 3, n_c = 48$) are given as Eq. (3.24) and (3.25):

$$H_{c;3\times 32} = \begin{pmatrix} 10110011011001101100110110011011 \\ 11010101101010110101011010101101 \\ 11101001110100111010011101001110 \end{pmatrix}, \tag{3.24}$$

$$H_{c;4\times 48} = \begin{pmatrix} 100010011010111100010011010111100010011010111100 \\ 010011010111100010011010111100010011010111100010 \\ 001001101011110001001101011110001001101011110001 \\ 000100110101111000100110101111000100110101111000 \end{pmatrix}. \tag{3.25}$$

3.2.4. Labeling

It is known that the existence of short cycles in the Tanner graph reduces the efficiency of BP decoding due to violating the hypothesis on independence of parity checks. Another interpretation of this phenomenon is that if the error combination covers a cycle, then check nodes involved in a cycle get zero syndrome values and do not help in error correction.

In this section, the technique for labeling base parity-check matrices of GLDPC codes is explained. The design goal is to match the parity-check matrices of hypergraphs of base binary QC-LDPC codes with columns of parity-check matrices of the constituent codes. Consider the following example.

Let H_b be the base matrix of the GLDPC code

$$H_b = \begin{array}{c} \\ c_1 \\ c_2 \\ c_3 \end{array} \begin{array}{c} \begin{array}{cccc} v_1 & v_2 & v_3 & v_4 \end{array} \\ \begin{pmatrix} 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \end{pmatrix} \end{array}. \tag{3.26}$$

Fig. 15 explains a procedure of labeling the base Tanner graph by columns of the parity-check matrix of the constituent code. For example, the base Tanner graph corresponding to H_b in Eq. (3.26) has the following cycle:

$$v_2 \to c_1 \to v_1 \to c_2 \to v_2. \tag{3.27}$$

Let $H_{\text{c}} = (\boldsymbol{c}_1, \boldsymbol{c}_2, ..., \boldsymbol{c}_{n_{\text{c}}})$ be a parity-check matrix of the constituent code. In a row-regular GLDPC code, each of the n_{c} nonzero elements in each row of the base matrix H_b is replaced by a column of H_{c}. Thus, rows of the base

parity-check matrix H_b are labeled by permutations of the set of columns of H_c.

By labeling the base matrix Eq. (3.26), a "coefficient" matrix corresponding to the labeled Tanner graph shown in Fig. 15 is obtained as

$$\begin{pmatrix} \boldsymbol{a}_1 & \boldsymbol{a}_4 & \mathbf{0} & \boldsymbol{a}_8 \\ \boldsymbol{a}_2 & \boldsymbol{a}_5 & \boldsymbol{a}_6 & \mathbf{0} \\ \boldsymbol{a}_3 & \mathbf{0} & \boldsymbol{a}_7 & \boldsymbol{a}_9 \end{pmatrix}, \tag{3.28}$$

where $\boldsymbol{a}_i$, $i = 1, ..., 9$, are chosen for each row j of H_b, denoted as H_b^j, as a subset of columns of the parity-check matrix $H_c = (\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3)$ of the constituent code. The algorithm for GLDPC construction is given as Alg. 3.

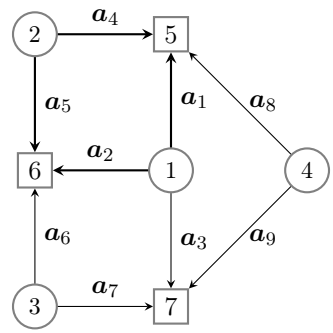

Figure 15. Labeled Tanner graph

Searching for harmful cycles is procedurally similar to what is done for labeling the base Tanner graph of QC-LDPC codes by powers of monomials [Fos04], using equations as in Sec. 2.5.3. However, the cycle existence condition in labeled GLDPC code graphs is not equivalent to the full-rank condition in case of NB-LDPC codes [PFD06]. In the case of GLDPC codes, a cycle in the base graph will remain problematic if the edges of the cycle in the labeled graph have a zero-sum of their assigned labels, resulting in a zero-syndrome component. For example, the cycle Eq. (3.27) will remain after labeling iff:

$$\boldsymbol{a}_4 + \boldsymbol{a}_1 + \boldsymbol{a}_2 + \boldsymbol{a}_5 = \mathbf{0}_{r_c \times 1}, \tag{3.29}$$

where addition of columns $\boldsymbol{a}_i$, $i = 1, ..., 9$ is performed in GF(2).

Notice that the columns $\boldsymbol{c}_i$ of H_c in Eq. (3.29) can be grouped pairwise as $((\boldsymbol{a}_1, \boldsymbol{a}_4), (\boldsymbol{a}_2, \boldsymbol{a}_5))$ by the check nodes they belong to in Eq. (3.28). Thus, in order to remove cycles from the labeled graph, it is sufficient to have nonequal columns $\boldsymbol{a}_i$ in at least one of the pairs corresponding to the cycle. This check is necessary, as constituent codes with minimum distance two are used in our construction. However, if all pairs consist of different columns, then the number of zero-syndrome components corresponding to the cycle is reduced.

Taking the above observations into account, it can be concluded that the optimization of the matching between the underlying binary LDPC code and a constituent code can be based on one of the following two criteria:

$\check{C}_1$: The total number N of remaining cycles in the Tanner graph of the underlying binary code after labeling of its parity-check matrix by columns of the parity-check matrix of the constituent code.

$\check{C}_2$: The total number N_{p} of satisfied subequations in the equations which determine a cycle the Tanner graph.

The described optimization approach is shown in Alg. 3.

Algorithm 3: Optimization of GLDPC codes

Input: H_b, H_c, I_{max}, g;
$I = 0; N_{best} = \infty$;
$\tilde{\mathcal{P}} = (0)_{r_{\mathrm{b}} \times n_{\mathrm{c}}}$;
Construct A_g, a $N_g \times g$ matrix of equations, describing all N_g cycles of length g in the base Tanner graph of H_b;
for $I = 1, \ldots, I_{max}$ **do**
    Choose a random $r_{\mathrm{b}} \times n_{\mathrm{c}}$ matrix $\mathcal{P}$ of permutations of column numbers $i = 1, ..., n_{\mathrm{c}}$ of $\boldsymbol{c}_i$ in H_{c}. Each row $\mathbf{r}_i$ of $\mathcal{P}$ corresponds to the i-th row of H_{b} and determines the labels assigned to nonzero elements in that row;
    For each equation in E_g compute the product $H_b^i H_{\mathrm{c}}^i$, where H_b^i is a submatrix of the base matrix H_{b} corresponding to nonzero elements in the i-th equation and H_{c}^i is the corresponding submatrix of H_{c};
    Evaluate the the number of cycles N using criterion $\check{C}_1$ or $\check{C}_2$;
    if $N < N_{\mathrm{best}}$ **then**
        $\tilde{\mathcal{P}} \leftarrow \mathcal{P}$;
        $N_{\mathrm{best}} \leftarrow N$;
return $\tilde{\mathcal{P}}$;

3.2.5. Decoding of GLDPC Codes

The key idea behind Tanner's proposed generalization of LDPC codes in [Tan81] is to replace the classical LDPC construction using rows of the parity-check matrix as single parity-check codes with larger constituent subcodes, achieving improved decoding performance as a result of the increased error-correcting capacity of the subcodes. While requiring increased complexity, the constituent codes can be decoded using the BCJR algorithm (Sec. 1.3.3), while exchanging extrinsic information in overlapping positions for the underlying BP decoder, same as in the decoding of binary LDPC codes, as described in Sec. 2.4.2.

Various approaches have been proposed to reduce complexity of decoding of the constituent codes. In [ZP01], Max-Log MAP is proposed for GLDPC decoding, as a simplification of the BCJR algorithm. Using Reed-Muller or Hamming codes for constituent codes, the Hadamard transform-based decoder from [AL04] can be used. The wrap-around Viterbi algorithm (WAVA) from [SLF03], which iterates over the TB trellis (Sec. 1.3.3), achieves near-ML performance with reduced complexity.

For simulations of GLDPC codes presented in this dissertation (Sec. 3.3.3), BCJR decoding was used (Sec. 1.3.3), applied to conventional trellises with complexity 2^3 and 2^4.

3.3. Bounds, Simulation and Comparison

In this section, in order to analyze and compare the two code classes, random ensembles of irregular NB LDPC and GLDPC codes, based on generalizations of Gallager's ensemble are considered. As analysis of LDPC codes by DE can only predict an optimal degree distribution for infinitely long codes, an approach analogous to [Boc+22] is applied for the analysis of finite length GLDPC codes, based on which, a finite-length random coding bound on the ML decoding error probability for the GLDPC code ensemble is derived and compared with the similar bound for the ensemble of NB LDPC codes introduced in [Boc+21]. The obtained bound can be interpreted as a target for BP decoding performance of the code ensembles.

In what follows, the ML decoding performance of the ensembles of irregular NB LDPC and binary GLDPC codes in the finite-length scenario is studied. Detailed overviews of finite-length upper bounds on the ML decoding error probability of general linear codes can be found in [SS06] and [PPV10], from which it follows that Poltyrev's TS bound [Pol94] is still the best benchmark for linear code ML decoding performance. As covered in Sec. 1.6.2, computation of the bound requires knowledge of the code weight spectrum.

3.3.1. Hamming Spectra for Ensembles of Irregular NB and GLDPC Codes

For the code ensembles considered below, the row-regularity requirement is relaxed. Consider an ensemble of irregular LDPC codes based on the generalization of Gallager's ensemble of binary (J, K)-regular LDPC codes in [Gal63]. In the generalized ensemble of the rate $1 - r_{\mathrm{b}}/n$ irregular binary LDPC codes, the parity-check matrix has the form $\begin{pmatrix} H_1 & H_2 & \dots & H_J \end{pmatrix}^{\mathrm{T}}$, where the i-th strip is chosen as a random permutation $\pi_i(\tilde{H}_i)$, where $\tilde{H}_i$

has the form:

$$\tilde{H}_i = (\underbrace{I_L \ ... \ I_L}_{K_i} \underbrace{\mathbf{0}_L \ ... \ \mathbf{0}_L}_{K-K_i}), \quad i = 1, ..., J, \tag{3.30}$$

that is, it contains a given number $K_i \leq K$ of identity matrices and $K - K_i$ of all-zero $L \times L$ submatrices, where $L = r_\mathrm{b}/J$, I_L denotes the identity matrix of order L, and $\mathbf{0}_L$ is the all-zero matrix of order L.

In this case, the strips in the generalized ensemble are permuted versions of the strips in the Gallager ensemble with some of the identity matrices replaced by all-zero matrices of the same order. By choosing the value of K_i, the column weight and row weight distributions could be adjusted. This base ensemble was introduced and used as a base ensemble for the random ensemble of irregular NB LDPC codes in [Boc+21].

First, consider the generalized Gallager's ensemble of binary irregular LDPC codes in Eq. (3.30). The average spectrum coefficients $\mathrm{E}\{A_{n,w}\}$ for this ensemble was defined in [Boc+22], where $A_{n,w}$ is a random variable representing the number of binary codewords of weight w and length n. $\mathrm{E}\{\cdot\}$ is the mathematical expectation over the code ensemble.

By following the approach in [Gal63], it was shown that for this binary ensemble, generating function of the number of binary sequences $\boldsymbol{x}$ of weight w and length n satisfying the equality $\boldsymbol{x}H_i^{\mathrm{T}} = \mathbf{0}$, $i \in \{1, 2, \ldots, J\}$, is equal to

$$G_i(s) = \sum_{w=0}^{n} G_{i,n,w} s^w = g_i(s)^L, \quad i = 1, ..., J, \tag{3.31}$$

where

$$\begin{aligned} g_i(s) &= (1+s)^{K-K_i} \sum_{j=0}^{K_i} g_{ij} s^j \\ &= (1+s)^{K-K_i} \frac{(1+s)^{K_i} + (1-s)^{K_i}}{2}, \end{aligned} \tag{3.32}$$

$g_{ij} = \binom{K_i}{j}$ if j is even, and $g_{ij} = 0$ otherwise.

NB LDPC Codes. In the NB case, the approach introduced in [Boc+22] is used. For an ensemble of NB LDPC codes over $q = 2^m$, denote the maximum number of q-ary symbols in each column and row as J and K, correspondingly. For the i-th strip of the parity-check matrix $\tilde{H}_i$, $i = 1, 2, \ldots, J$, let K_i denote the number of q-ary elements in the i-th strip. Then, similar to Eq. (3.31), the weight generating function of length-n sequences $\mathbf{x}$ satisfying $\mathbf{x}\tilde{H}_i^{\mathrm{T}} = \mathbf{0}$ is expressed as

$$G_i(s) = \sum_{w=0}^{n} G_{i,n,w} s^w = f_i(s)^L, \quad i = 1, ..., J, \tag{3.33}$$

where $f_i(s)$ denotes the symbol weight generating function of sequences $\mathbf{x}$ that satisfy the non-zero part of one q-ary parity-check equation, given as

$$f_i(s) = (1+(q-1)s)^{K-K_i} \frac{(1+(q-1)s)^{K_i} + (q-1)(1-s)^{K_i}}{q}. \quad (3.34)$$

In order to estimate the average bit weight for an ensemble of NB LDPC codes, binary images of NB LDPC codes must be considered. Let $\hat{H}_i$ denote the binary image of the i-th strip $\tilde{H}_i$, $i = 1, 2, \ldots, J$ of the q-ary parity check matrix of the NB LDPC code, and assume a binomial distribution of binary symbols in the m-dimensional binary image of a q-ary symbol. Based on a generalization of Eq. (1.3) in [Fel68, Chapter XII], the average bit weight generating function is given by the composition of two generating functions as

$$F_i(s) = \sum_{w=0}^{nm} F_{i,nm,w} s^w = f_i\left(\phi(s)\right)^L, \quad (3.35)$$

where $F_{i,nm,w}$ is the average number of weight-w, length-nm sequences $\mathbf{b}$ satisfying $\mathbf{b}\hat{H}_i^{\mathrm{T}} = \mathbf{0}$, and

$$\phi(s) = \sum_{i=1}^{m} \frac{1}{q-1} \binom{m}{i} s^i = \frac{(1+s)^m - 1}{q-1}. \quad (3.36)$$

Finally, the average bit weight for the ensemble can be computed as

$$\mathrm{E}\{A_{n,w,m}\} = \binom{nm}{w}^{1-J} \prod_{i=1}^{J} F_{i,nm,w}. \quad (3.37)$$

GLDPC Codes. Consider an ensemble of irregular GLDPC codes obtained by labeling nonzero elements of the i-th row in the base parity-check matrix by columns of the $[n_{\mathrm{c},i}, k_{\mathrm{c},i}]$ binary linear constituent code, substituting the values in the i-th strip of a random parity-check matrix of the base ensemble - replacing zeros by all-zero columns of length $r_{\mathrm{c},i} = n_{\mathrm{c},i} - k_{\mathrm{c},i}$ and ones by columns of a (non-random) parity-check matrix of a constituent code. Notice that $[n_{\mathrm{c},i}, k_{\mathrm{c},i}]$ constituent codes for some i can coincide. Then, the rate of the obtained GLDPC code is equal to $R = 1 - \sum_{i=1}^{J} \nu_i (1 - R_{\mathrm{c},i})$, where $\nu_i = \frac{n_{\mathrm{c},i}}{n}$, $J_{\mathrm{av}} = \sum_{i=1}^{J} \nu_i$, and $R_{\mathrm{c},i}$ is the rate of the i-th constituent code of length $n_{\mathrm{c},i}$.

Similar to Eq. (3.31) and (3.35), the generating function is expressed as $F_i(s)$, $i = 1, ..., J$, of the number of binary sequences $\boldsymbol{x}$ of weight w and length n satisfying the equality $\boldsymbol{x}H_i^{\mathrm{T}} = \mathbf{0}$, $i \in \{1, 2, ..., J\}$, where H_i is the i-th strip of the parity-check matrix of the GLDPC code, in the form

$$F_i(s) = \sum_{w=0}^{n} F_{i,n,w} s^w = (1+s)^{(K-n_{\mathrm{c},i})L} q_i(s)^L, \quad (3.38)$$

where

$$q_i(s) = \sum_{j=0}^{n_{\mathrm{c},i}} A_{i,j} s^j, \tag{3.39}$$

is the weight generating function of the i-th constituent code and $A_{i,j}$, $j = 0, 1, ..., n_{\mathrm{c},i}$ are the spectrum coefficients of the constituent code.

The probability that the binary sequence $\boldsymbol{x}$ of weight w and length n satisfies $\boldsymbol{x}H_i^{\mathrm{T}} = \mathbf{0}$ can be expressed as:

$$p_i(w) = \frac{F_{i,n,w}}{\binom{n}{w}}. \tag{3.40}$$

The average spectrum coefficients follow as:

$$\mathrm{E}\{A_{n,w}\} = \binom{n}{w}^{1-J} \prod_{j=1}^{J} F_{j,n,w}. \tag{3.41}$$

In Fig. 16, the average spectra of both regular and irregular GLDPC codes of rate $R = 3/4$ and length $n = 2016$ bits with $[n_{\mathrm{c}}, n_{\mathrm{c}} - 4]$ constituent codes are compared to the average spectra of the binary images of the rate $R = 3/4$ NB LDPC codes over GF(2^4) computed as in [Boc+21]. As follows from the presented plots, the average spectra of these two ensembles are rather close to each other. However, even for the case $J_{\mathrm{av}} = 3$, spectra for both ensembles have a visible gap to the average spectrum of the rate $R = 3/4$ random binary code presented in the same figure.

3.3.2. ML Decoding Bound for Ensembles of Irregular NB LDPC and GLDPC Codes

Assume that a linear $[n, k]$ code $\mathcal{C}$ is used with BPSK modulation and coherent detection to communicate over an AWGN channel. The binary code symbol $c_i \in \{0, 1\}$ is mapped onto the signal

$$v_i = (2c_i - 1)\sqrt{E_{\mathrm{s}}}, \quad i = 1, 2, \ldots, n,$$

where E_{s} is the signal energy. Thus the codewords

$$\boldsymbol{c}_j = (c_1^{(j)}, c_2^{(j)}, \ldots, c_n^{(j)}), \quad j = 0, 1, \ldots, 2^k - 1,$$

are mapped onto bipolar sequences

$$\boldsymbol{v}_j = (v_1^{(j)}, v_2^{(j)}, ..., v_n^{(j)}).$$

Assume that $\boldsymbol{v}_0$ is transmitted. Then, the received signal is

$$\boldsymbol{r} = \boldsymbol{v}_0 + \boldsymbol{n},$$

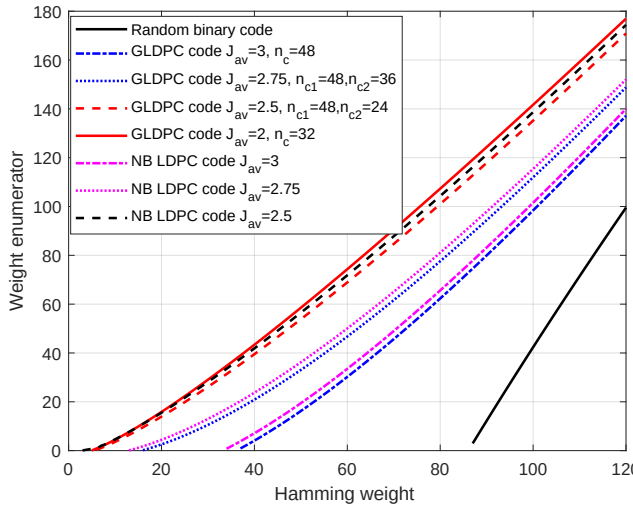

Figure 16. Spectra of the rate $R = 3/4$ GLDPC codes of length ≈ 2000 bits with $[n_c, n_c - 4]$ constituent codes versus spectra of the rate $R = 3/4$ NB LDPC codes over $GF(2^4)$.

where the noise vector $\boldsymbol{n}$ consists of independent zero-mean Gaussian random variables with variance $\sigma^2 = N_0/2$.

From here, the finite length random coding bound on the ML decoding error probability for the binary input-AWGN channel can be computed by substituting the average spectrum for the ensemble of the irregular NB or GLDPC codes, given by Eq. (3.37) or (3.41), correspondingly, to the TS bound [Pol94] (Sec. 1.6.2).

Fig. 17, shows the comparison of the finite length random coding bounds on the ML decoding error probability for two classes of rate $R = 3/4$ codes: GLDPC codes with $[n_c, n_c - 4]$ constituent codes and NB LDPC codes over GF(2^4). Although spectra for these classes of codes, presented in Fig. 16, are rather close to each other, the corresponding bounds on the ML decoding error probability for the same average column weight J_{av} have a gap increasing as J_{av} decreases. In the following, the derived bound on ML decoding performance is interpreted as a lower bound on the FER performance of the BP decoding for the two classes of codes.

3.3.3. Decoding Performance of NB and GLDPC Codes

In this section, the FER performance of sum-product BP decoding for a set of rate $R = 3/4$ optimized GLDPC codes of length ≈ 2000 bits with different average column weights J_{av} of their base parity-check matrices is compared with the performance of two optimized NB QC-LDPC codes of the same rate and length over GF(2^m), $m = 4; 5$. The simulations were performed until fifty frame errors were encountered with a maximum of 50

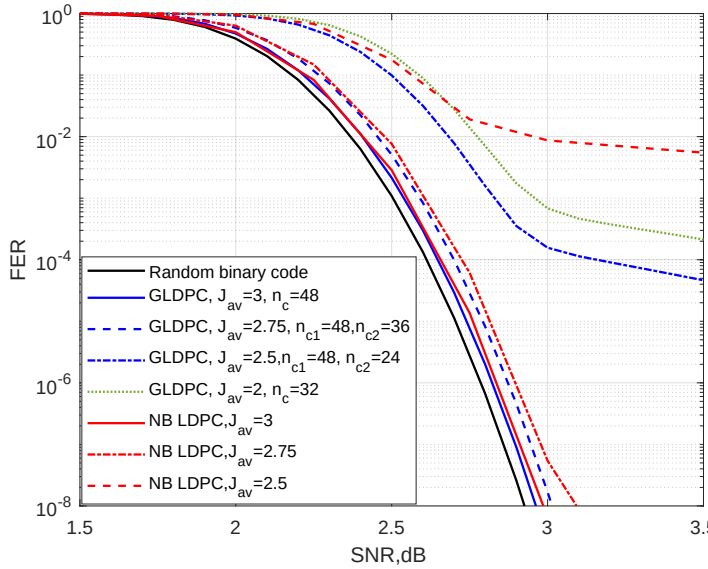

Figure 17. Bound [Pol94] for the rate $R = 3/4$ GLDPC codes of length ≈ 2000 bits with $[n_c, n_c - 4]$ constituent codes versus the same bound for the rate $R = 3/4$ NB LDPC codes over $GF(2^4)$.

iterations. The studied QC-GLDPC code with $[n_c, n_c - 3]$ constituent codes is constructed using the 25×300 base parity-check matrix with variable node degree distribution

$$\lambda(x) = \frac{125}{300}x + \frac{150}{300}x^2 + \frac{25}{300}x^7, \tag{3.42}$$

i.e., $J_{\text{av}} = 3.0$. The lifting factor $\hat{M}$ is equal to 7. The QC-GLDPC code with $[n_c, n_c - 4]$ constituent codes is determined by the 31×496 base matrix with variable node degree distribution

$$\lambda(x) = \frac{310}{496}x + \frac{155}{496}x^2 + \frac{31}{496}x^{10}, \tag{3.43}$$

and thus, $J_{\text{av}} = 2.875$. The lifting factor $\hat{M}$ is equal to 4. Although the bounds on the ML decoding performance of the ensemble codes determined by base matrices with only weight two and three columns do not show error floor if J_{av} is large enough, it is not the case for the FER performance of BP decoding of practical codes. In order to avoid the error floor, higher weight columns were added to the base matrices. The precise degree distributions in Eq. (3.42) and (3.43) were obtained empirically through optimization by simulation.

For comparison, NB QC-LDPC codes determined by the 14×56 base matrix with column weights two and three are considered. The lifting factor $\hat{M}$ is equal to 9, and 8 for the codes over GF(2^4) and GF(2^5), respectively.

The average column weight is $J_{\text{av}} = 2.62$ and $J_{\text{av}} = 2.5$. The finite length random coding bounds for the ensemble of GLDPC codes of the same length with $r_{\text{c}} = 3$, $J_{\text{av}} = 3$ and $r_c = 4$, $J_{\text{av}} = 2.75$ are shown in the same figure.

It can be seen from Fig. 18 that the simulated GLDPC and NB QC-LDPC codes outperform binary LDPC codes used in WiFi and 5G standards. Furthermore, the two code classes exhibit nearly identical decoding error rate performance at lower SNRs, with a difference emerging in favor of the NB QC-LDPC code over GF(2^5) in the error floor region.

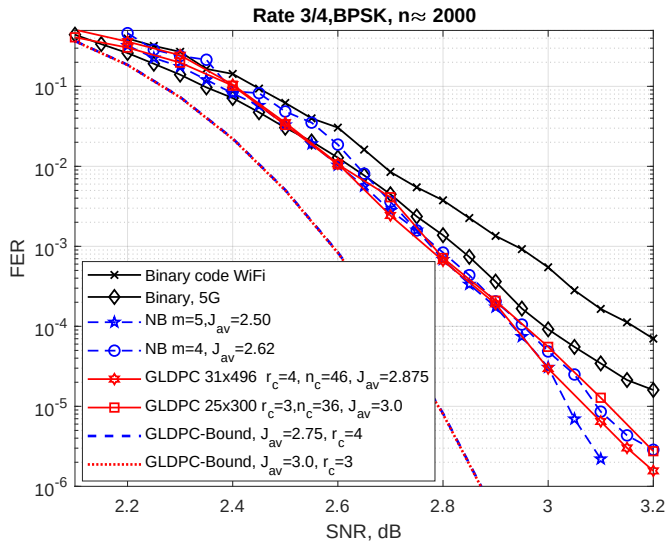

Figure 18. BP decoding FER performance of the rate $R = 3/4$ GLDPC codes with $[n_{\text{c}}, n_{\text{c}} - 3]$ and $[n_{\text{c}}, n_{\text{c}} - 4]$ constituent codes versus the FER performance of BP decoding for the rate $R = 3/4$ irregular NB QC-LDPC codes over GF(2^3)–GF(2^4).

4. ANALYSIS OF SHAPED AND UNSHAPED CODED MODULATION

Transmission of data over a physical channel requires transformations from sequences over a finite alphabet to parameters of physical harmonic waveform signals, and back to the code alphabet sequences. These transformations are known as modulation and demodulation, respectively.

In the context of digital communication, modulation can be summarized as the process of determining the mappings of input binary sequences to sequences of transmittable signals over a communication channel, and demodulation, as its inverse process. Based on a discrete set of signals, chosen as to satisfy the requirements of a communication system, such as achieved transmission rates, computational complexity, and restrictions on signal power, the goal is still to reach the greatest reliability of communication achievable.

The basics of modulation were covered in Sec. 1.5. From hereafter, the focus is on coded modulation (CM), used with PAM signaling, for which the methods and analysis will be designed for. While offering a simplified analysis, the approaches are generalizable and applicable to a wider range of types of modulation.

4.1. Coded Modulation

BICM, from [CTB98], has since its introduction become standard in modern communication systems due to the simplicity of its implementation. A high-level description of the communication model using BICM is given in Fig. 19, where $\mathbf{u}, \hat{\mathbf{u}}$ are binary vectors, $\mathbf{c}$ is a codeword, Π denotes the BICM interleaver and $\mathbf{c}'$ its output, $\mathbf{x}$ is a sequence of channel symbols, $\mathbf{n}$ is noise in the AWGN channel, $\mathbf{y}$ is a vector of received noisy channel symbols, $\mathbf{L}, \mathbf{L}'$ are bit-LLR estimates and $\hat{\mathbf{u}}$ is the decoded output.

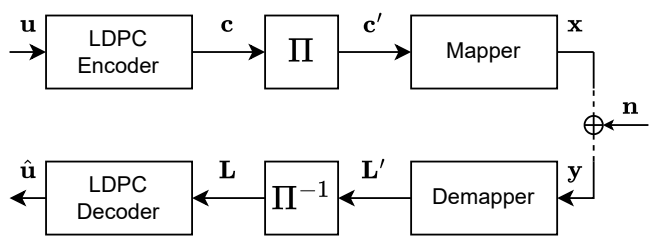

Figure 19. BICM system

The intuition behind BICM is to design a modulation system - the combined interleaver Π and mapper, creating an "ideal" memoryless channel from the decoder perspective. To achieve this, local dependencies in LLRs

resulting from an erroneous received high-order modulation symbol need to be "broken" by the interleaver. The interleaver Π, in theoretical analysis of the BICM model, is assumed to be "ideal", meaning, of infinite depth and fully random.

More formally, BICM is constructed as a concatenation of an encoder of a binary code $\mathcal{C}$ and an N-dimensional memoryless modulator. For a signal set $\mathcal{X}, |\mathcal{X}| = M = 2^p$, a binary labeling map $\mu : \{0,1\}^m \to \mathcal{X}$ determines a one-to-one correspondence between length-m binary sequences and symbols of $\mathcal{X}$. A codeword $\mathbf{c}$ is interleaved, denoted as $\pi(\mathbf{c}')$, and split into subsequences of length m for the mapper μ. The interleaver is defined as another one-to-one mapping $\pi : k \to (k', i)$, where k, k' denote a time ordering of the input $\mathbf{c}_k$ or output $\mathbf{x}_k$, accordingly, and i denotes the position of $\mathbf{c}_k$ in the label of $\mathbf{x}_{k'}$.

While BICM has become a *de facto* standard, the widespread acceptance of coded modulation (or channel coding), as a prerequisite for near-capacity achieving communications, followed from [Ung82]. The proposed coding scheme, working with probabilistic coding and decoding of modulated signals, using rate $R = m/(m+1)$ binary convolutional codes mapping onto 2^{m+1} channel signals, resulted in performance gains of $3 - 4$dB using 8-PSK modulation compared to an uncoded 4-PSK system. While resulting in major gains, the scheme ultimately proved difficult to practically implement due to issues of matching with ECCs.

In the context of NB LDPC codes, as demonstrated in [Boc+23c], BICM is outperformed by other types of coded modulation. Specific to NB codes, the matching of code alphabet size and modulation order, and mappings more sophisticated than BICM yield improved decoding performance. In this dissertation, four mappings of binary images of NB QC-LDPC code codewords are considered, as described in [Boc+23c]: *(A)* BICM, *(B)* symbol-interleaved coded modulation (SICM) (Fig. 20), *(C)* bit-plane coded modulation (BPCM) (Fig. 21) and *(D)* amplitude-sign coded modulation (ASCM) (Fig. 22).

For each of the mappings, $M = 2^p$-PAM signaling is used. Consider a binary linear $[n, k]$ code of length n and dimension k with systematic encoding be used for transmission over the AWGN channel. When 2^p-PAM signaling is used, each signal point represented by p bits. A codeword $\boldsymbol{c} = (c_1, c_2, ..., c_n)$ is mapped onto the signal sequence $\boldsymbol{x} = (x_1, x_2, ..., x_{N_p})$, $N_p = n/p$, which is transmitted over the channel. The sign- and amplitude bits s_i, a_{ij} are obtained from $\mathbf{c}$ based on the type of coded modulation used (*e.g.,* Figures 20, 21, 22), where bits of codewords that are mapped to sign bits s_i are highlighted in red. The PAM signal x_i is computed as

$$x_i = (2s_i - 1)A(a_{i1}, \ldots, a_{i(p-1)}), \tag{4.1}$$

where s_i, treated as an integer, determines the sign of x_i, and $A(\cdot)$ denotes

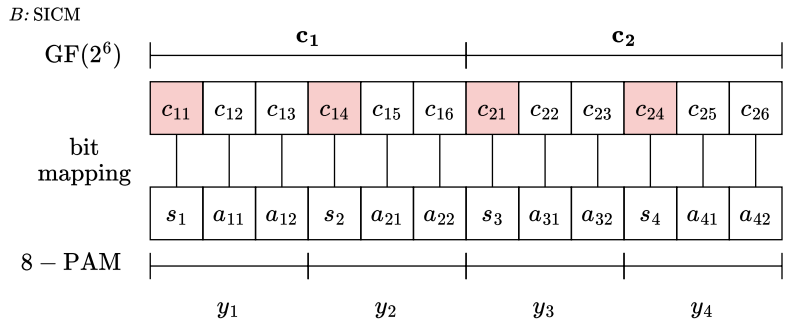

Figure 20. SICM mapping for code alphabet GF(2^6) with 8-PAM signaling

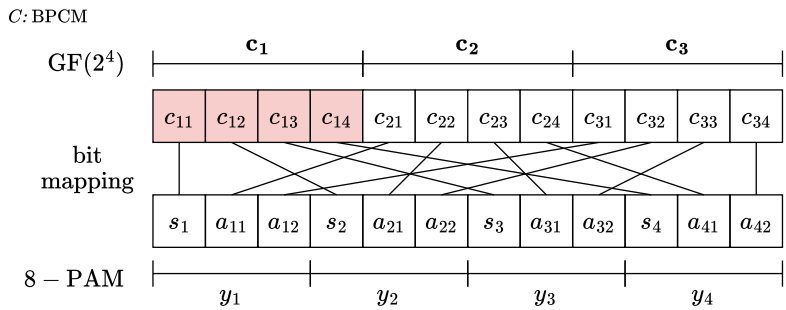

Figure 21. BPCM mapping for code alphabet GF(2^4) with 8-PAM signaling

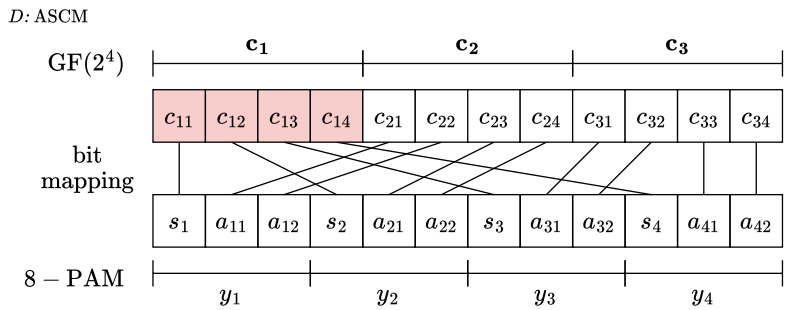

Figure 22. ASCM mapping for code alphabet GF(2^4) with 8-PAM signaling

a mapping of the sequence of amplitude bits to the amplitude value of the signal point performed according to the Gray rule (Def. 1.5.1), $i = 1, ..., N_p$, and $N_p = n/p$ denotes the length of the sequence in PAM signals. While Gray coding is a common method used for the mapping, specific design and optimization of mappings $A(\cdot)$ is described extensively in Sec. 5.

Three of the four considered coded modulation type mappings are visualized in Figures 20, 21, 22. The three have specific structured bit mappings, depending on the pairing of the code alphabet and modulation order sizes, while the BICM design uses a pseudorandom interleaver instead.

B: SICM, for a pairing of a code over GF(2^m) and 2^p-PAM, requires p to

be a divisor of m, resulting in some inflexibility of the modulation system in terms of applicability. As the number of sign and amplitude bits (s_i, a_{ij}, respectively, in Fig. 20) per each code symbol $\mathbf{c}$ is equal, each symbol of a codeword is equally reliable. Typically SICM offers the best decoding performance out of all considered types of coded modulation, but it is only applicable if $p \mid m$.

C: BPCM offers increased flexibility of system parameters, with a codeword split into segments with size dependent on m and p. Every $(p+1)$-th, $(2p+1)$-th, ... group of m bits determining the signs s_i of m PAM signals and every j-th group of m bits in between the sign bits determining amplitude bits a_{ij}.

D: ASCM assumes that $p-1$ is a divisor of m. With sign bits assigned the same as with BPCM, with one code symbol defining m signs. With $m = a(p-1)$, each other code symbol is used to determine (sequentially) the amplitude bits of a PAM signals.

4.2. Analysis of NB LDPC Code Ensemble SEDS with Coded PAM Signaling

In order to determine the SEDS of ensembles of NB LDPC codes used with coded PAM signaling, it is necessary to account for nonlinearity of the Euclidean space formed by modulation signals. Furthermore, as the Euclidean distances of PAM signal points are dependent on the mappings of code symbols to PAM signals, a method of relating the Hamming weight spectrum of the NB LDPC code ensemble to the SED of PAM signals is required. To achieve this, the methods introduced in [Boc+23c] and [BKS23] can be used.

Consider a NB LDPC code over GF(q), $q = 2^m$, used with $2^p = M$-PAM signaling. Denote as n_s and n_p the number of q-ary code symbols and number of M-PAM signals grouped together, correspondingly. The size of the grouping is determined such that

$$n_s m = n_p p = n_b, \tag{4.2}$$

where n_b denotes the resulting number of jointly processed codeword bits for the analysis. These values depend not only on m and p, but also on the choice of coded modulation being used. Considering the four types of CM introduced before: for BICM, $n_p = 1, n_b = p$, for SICM, $n_s = 1, n_p = m/p, n_b = m$. For ASCM and BPCM, they are chosen as integers that satisfy Eq. (4.2).

Consider a sequence $\mathbf{s}_i$ of M-PAM signals of length n_p and the length-n_b binary sequence $\mathbf{b}_i$ that corresponds to $\mathbf{s}_i$, based on the used mapping. Then, for all possible pairs $(\mathbf{s}_i, \mathbf{s}_j)$, and corresponding $(\mathbf{b}_i, \mathbf{b}_j)$, the MGF

(Def. 1.6.3)

$$\alpha_{n,d}(\lambda) = \sum_{i=1}^{M^{n_p}} \sum_{j \neq i} \Pr\{d_{\mathrm{E}}^2(\mathbf{s}_i, \mathbf{s}_j), d_{\mathrm{H}}(\mathbf{b}_i, \mathbf{b}_j) | n, d\} \lambda^{\delta_{ij}}, \tag{4.3}$$

was introduced in [Boc+23c], where d denotes the Hamming distance between two length-n codewords $\mathbf{v}_i, \mathbf{v}_j$ with binary images $\mathbf{b}_i, \mathbf{b_j}$ and

$$\delta_{ij} = \frac{d_{\mathrm{E}}^2(\mathbf{s}_i, \mathbf{s}_j)}{d_{\mathrm{H}}(\mathbf{b}_i, \mathbf{b}_j)}$$

is the *normalized SED*, which is the ratio of SED between signal sequences $\mathbf{s}_i, \mathbf{s}_j$ corresponding to codewords $\mathbf{v}_i, \mathbf{v}_j$ and Hamming distance between the binary images $\mathbf{b}_i$ and $\mathbf{b}_j$.

Then, for two codewords at Hamming distance d, the average MGF of the SEDs between the corresponding coded QAM sequences was defined in [BKS23] as

$$G_d(\lambda) = \alpha_{n,d}^d(\lambda), \tag{4.4}$$

where, for simplification, d_{E}^2 is approximated as a sum of i.i.d. variables.

Finally, the SED spectrum of coded QAM signal sets is computed as

$$A_{\mathrm{E}}^{\mathrm{bit}}(\lambda) = \sum_{w} A_{\mathrm{H}}(w) G_w(\lambda), \tag{4.5}$$

where $A_{\mathrm{H}}(\rho)$ is the average Hamming spectrum of an ensemble of binary images of NB LDPC codes (Sec. 3.3.1), computed as Eq. (3.35).

4.3. Shaping

It is well known that shaping of modulated signaling can be used to approach the Shannon limit on the signal-to-noise ratio for reliable coded modulation transmission over the AWGN channel. Capacity of the discrete time memoryless channel with restriction on the input signal energy E and additive Gaussian noise with variance σ^2 is defined as

$$C(\mathrm{SNR}) = \max_{\{p(x)\}: \mathrm{Var}(X) \leq E} I(X;Y) = \frac{1}{2} \log_2 (1 + \mathrm{SNR}) \tag{4.6}$$

bits per dimension, where $\mathrm{SNR} = E/\sigma^2$, $I(X;Y)$ denotes the average mutual information between the input X and output Y, and $\{p(x)\}$ denotes the input probability distribution on X. If X is a continuous random variable, then the maximum in Eq. (4.6) is achieved when $\{p(x)\}$ approaches the Gaussian distribution, that is, when X is the Gaussian variable with zero mean and variance E.

An ultimate shaping gain of up to 1.53 dB is theoretically achievable ([For+84]) by shaping the used signal constellation to match the Gaussian distribution, compared to the uniform input distribution. Using M-PAM (or M^2-QAM) signaling, X is a discrete random variable, the optimal distribution $\{p(x)\}$ for which can be approximated by the Maxwell-Boltzmann distribution for some signal alphabet size M. The achievable shaping gain decreases as M decreases.

The "separation principle" in [FW89] motivated the search for good shaping techniques independent of the coding problem (see, for example, [For92; Kai+07; CO90; LFT94]). However, in general, shaping reduces the average signal energy at the cost of additional redundancy. For a fixed data transmission rate, this redundancy should be compensated by increasing the rate of the ECC. In what follows, shaping is considered in conjunction with error-correcting coding.

4.3.1. Probabilistic Amplitude Shaping

Probabilistic shaping is a commonly used shaping technique based on using uniformly spaced signal points with different probabilities chosen to approach the optimal distribution. Existing systems using these shaping techniques can be divided into two large classes - shaping before coding (SBC), where a shaping encoder precedes the ECC encoder ([BSS15; SB16; Gül+18; Feh+19; Feh+15; SLB17]) and communication systems with shaping after coding (SAC), where shaping techniques are applied to the output of ECC encoder ([KP93; Ung02; RG04; BB04; SH05; BB06; Le +07; VX12; Yan+14; Bou+14; Fen+15; PK16; Zho+16]).

Probabilistic amplitude shaping (PAS) used for providing energy-efficient amplitude distribution in [BSS15; SB16; Gül+18; Feh+19; Feh+15; SLB17] follows ideas similar to those in [CO90], [LFT94]. In [BSS15], Böcherer *et al.* study a scheme for coded modulation systems with binary LDPC code of length 64800 used in the DVB-S2 standard. The suggested scheme is based on enumerative decoding of constant-composition codes [Cov73]. In the recent paper [Gül+18], *enumerative sphere shaping codes* were used in combination with binary LDPC codes from IEEE802.11 (WiFi) standard and PAM modulation. Larger shaping gain is achieved due to the use of not only sequences with fixed optimal (sub-optimal) composition, but also all sequences inside a sphere whose radius is determined by a given energy constraint. This approach is an improvement of the shaping technique in [BSS15]. For moderate lengths, coding gain larger than 1 dB compared to constant composition shaping codes was claimed in [Gül+20a]. The constant-composition code based approach in combination with NB LDPC codes was studied in [SLB17]. In that paper, NB LDPC codes of length 576 and 1008 bits over GF(2^8) determined by parity-check matrices with two nonzero elements in each column were used with 8-PAM and 16-PAM

signaling.

4.3.2. Shaping Before Coding

Shaping schemes in which a target signal amplitude distribution is achieved by shaping operations preceding coding by the used ECC, such as the original proposed PAS in [BSS15], can be classified as SBC. In this type of architecture (Fig. 23), shaping can be considered as an instance of outer layer coding, where code parity symbols are then added by the ECC in the inner layer.

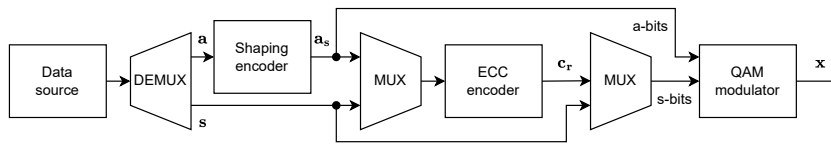

Figure 23. Block-scheme of SBC-ECC encoder used with QAM signaling

In the SBC scheme, first a k_a-bit input is encoded by the shaper into a $k_a + k_{as} = k_{af}$-bit vector $\mathbf{a}_s$, where $k_{af} > k_a$. Assume an ECC with systematic encoding is used. Then, the probability distribution of the k_{af} information bits does not change in the ECC encoder, whereas the check bits $\mathbf{c}_r$, typically, are distributed uniformly. This allows the usage of information bits to determine the signal amplitudes (a-bits) and the shaping of their probability distribution (resulting in shaping redundancy, as visualized in Fig. 24), in order to approach the theoretical limit. In Fig. 24, the ideas behind mapping an $n = Np$-bit codeword onto N 2^p-PAM signals are shown. Each signal point $x_i = (2s_i - 1)A(a_{i,1}, \ldots, a_{i,(p-1)}), i = 1, \ldots, N$, is produced by a mapper $A(\cdot)$, taking $p-1$ a-bits to determine the amplitude, and a sign bit s_i.

In practice, in an SBC PAS architecture, a target transmission rate R_t is obtained by further added redundancy on top of coding redundancy produced by the ECC. Figure 24 shows the distribution and origin of Np bits used to determine N 2^p-PAM signals using SBC. Assuming systematic encoding, the parity check bits (coding redundancy) produced by the ECC will be

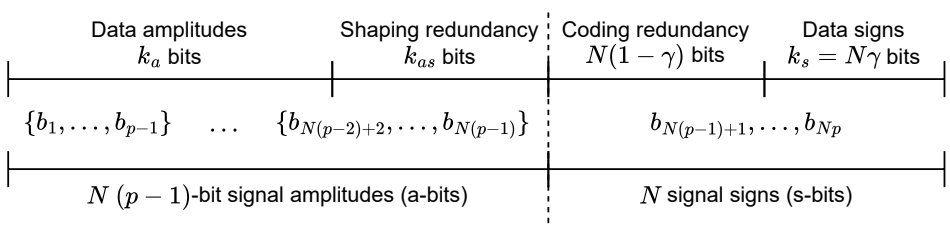

Figure 24. SBC PAS bits per N symbols

used to determine the signs of $N(1-\gamma)$ signals, where $\gamma = 1-(1-R_{ECC})p$. In the case of $\gamma = 0$, which holds only if $R_{ECC} = (p-1)/p$, only check bits are used for signs. In [BSS15], a construction is proposed allowing for code rates $R_{ECC} > (p-1)/p$, to be used with this scheme. Then an additional $k_s = N\gamma$ data bits must be used for signs.

In Fig. 23, the codeword $\boldsymbol{c}$ of the systematic encoder has the form $\boldsymbol{c} = (\boldsymbol{c}_i, \boldsymbol{c}_r) = (c_1, ..., c_{(k_{af}+k_s)}, \boldsymbol{c}_r)$. The merged sequence of the k_s message s-bits and the parity-check bits of the ECC encoder are used as signs of the M-PAM signals.

Assuming uniformly distributed codeword parity bits, the signs will be equiprobable, as required for optimal distribution of modulation signals. Amplitude bits matching the target probability distribution are produced by the shaping encoder by the addition of shaping redundancy. The shaping redundancy per bit can be expressed as

$$R_{\mathrm{ECC}} - R = (1 - R_{\mathrm{SH}})\,\frac{(p-1)}{p}, \tag{4.7}$$

where $R_{\mathrm{SH}} = k_a/k_{af}$ is the rate of the shaper, $R_{\mathrm{ECC}} = \frac{k_{af}+k_s}{n}$ and $R = k/n = (k_a+k_s)/n$ is the overall (target) code rate. As follows from Eq. (4.7), if SBC scheme is used then an additional redundancy has to be introduced in order to provide the required capacity-achieving input distribution. This leads to the transmission rate loss.

Example 1: Let $R = 12/20$. Consider an $[20, 15]$ binary ECC used with a rate $R_{\mathrm{SH}} = 4/5$ shaping encoder and 16-PAM ($p = 4$) signaling. First, a $k_a = 12$-bit input is encoded by the shaping encoder, outputting a $k_{af} = 15$-bit message to be encoded by the ECC. The resulting $n = 20$ bits are mapped onto $N = 5$ 16-PAM signal points. As $R_{ECC} = (p-1)/p = 3/4$, $\gamma = 0$ and no data (information) bits are required to determine the signs of the signal points.

Example 2: Alternatively, for the same rate, consider the same ECC and shaping with 4-PAM signaling. Then $k_a = 7, k_{as} = 3$, $\gamma = 1/2$, $p = 2$ and the 20 bits would be mapped onto $k_{af} = N(p-1) = N = 10$ PAM signal points, requiring $k_s = N\gamma = 5$ data bits to be used as sign bits. The resulting construction has $R_{\mathrm{SH}} = 7/10, R_{\mathrm{ECC}} = 15/20$ and $R = 12/20$.

Denoting the total length of shaped a-bits as $k_{af} = N(p-1)$, strategies of transforming the k_a-bit input to a k_{af}-bit output matching the target probability distribution when mapped to signal amplitudes, are varied. In essence, these methods can be considered as variable-length source decoding techniques and can be implemented as decompressors in a compression scheme, while also facing typical challenges associated with these schemes, such as buffering (including overflow), synchronization and error propagation

issues. In [BSS15], a constant composition distribution matching (CCDM) is used to provide an invertible source encoding, producing the target signal distribution. CCDM, described in [SB16], uses arithmetic decoding to produce the required fixed-to-fixed length distribution matcher. In [Feh+19], multiset partition distribution matching (MPDM) is proposed as an improvement of CCDM, targeting the average target distribution in an n-dimensional signal space.

4.3.3. Shaping After Coding

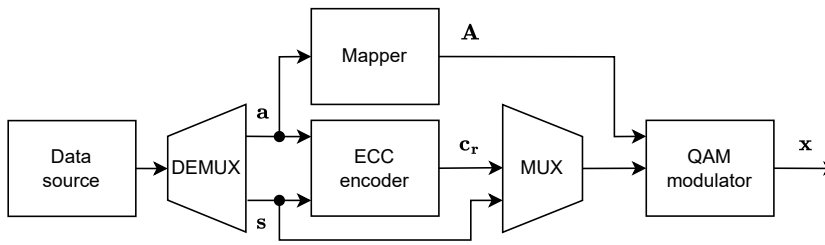

Figure 25. Block-scheme of ECC-SAC encoder used with QAM signaling

SAC studied in [KP93] and [Ung02] belongs to the class of variable-length coding techniques. It is based on using signal points with different probabilities according to the optimal distribution and splitting input data stream into codewords of the Huffman code which are then mapped onto the constellation points. However, this approach suffers from typical problems of variable-rate coding schemes such as a need for buffering and possible loss of synchronization, etc. SAC can be implemented as a fixed-rate scheme as well (see, for example, [PK16]). The corresponding approach is called many-to-one (M2O), or Gallager mapping [Gal68]. It maps bit sequences onto the constellation points, with more than one sequence mapped onto points which should have larger probability. Possible errors caused by M2O are resolved by an error-correcting code (for example, an LDPC code). Various M2O SAC techniques combined with turbo-codes, binary and NB LDPC codes are studied in [RG04; BB04; SH05; BB06; Yan+14]. Another approach, called constellation extension (CE) (or alphabet extension), introduces additional signal points that can be used by the modulation system - for example, extending 4-PAM to use 6 signal points instead.

A turbo-coded system with redundant SAC is presented in [Le +07]. Low complexity turbo-coded SAC shaped modulation technique with very low shaping redundancy (1-bit) is studied in [Bou+14]. The SAC scheme similar to [Le +07; Bou+14] but based on LDPC coding and using non-uniformly spaced, instead of uniformly spaced, signal points is presented in [VX12].

Examples of M2O and CE are given as *shaping books* in Table 4, with

$\mathcal{B}_{a1}$ defining a M2O mapping, repeating a amplitude vector 1311, and $\mathcal{B}_{a2}$ combining both M2O and CE, extending the constellation to 6-PAM and repeating the vector 1111.

A shaping book (or code book) serves as a lookup table (LUT), providing a mapping of $bin(i)$ to a vector $\mathbf{A}_i$ of signal amplitudes, where i denotes the index of the amplitude vector in $\mathcal{B}$.

If SAC is used, amplitudes $A(\cdot)$ will be determined by a mapper in a block-wise manner, with achieved gain being limited by complexity restrictions on the block length of the shaped signals. Each of these blocks is mapped onto the amplitude indices of M-PAM signals with a proper distribution.

For the AWGN channel, the capacity-achieving input distribution on the signal points is symmetric. This results in uniform distribution for the sign bit, whereas a probability distribution of the amplitude bits is non-uniform, and it influences both the mutual information $I(X;Y)$ and the average energy of the signal constellation.

Fig. 25 shows the encoder with SAC based communication system. The message data stream is split into a-bits $\mathbf{a}$ and s-bits $\mathbf{s}$. Both the message a-bits and s-bits enter the ECC encoder, where check bits are computed. These check bits are merged with the data s-bits and then are used as the sign bits by the QAM modulator. From the data a-bits short blocks $\mathbf{A}$ of a given length are formed in the mapper. Each of these blocks is mapped into the amplitude indices of M-PAM signals with a proper distribution. The mapper in Fig. 25 consists of block former followed by a shaper. SAC does not lead to any rate loss but it is applied in a block-wise manner. The achieved gain is limited by complexity restrictions on the block length of the shaped signals.

4.4. Coded Modulation with Shaping After Coding

It is well-known that both shaping and matching of PAM signals with NB LDPC codes can significantly influence the performance of BP decoding (*e.g.,* [DCG04]). In order to achieve low-complexity shaped coded modulation, encoder-modulator schemes must be designed, matching NB LDPC codes with shaped M-PAM signaling.

To illustrate the construction, in this section, the focus is on so-called 'ultra-sparse' NB LDPC codes over GF(2^8) and GF(2^6) used with 8-PAM signaling and SAC. For $m = 8$, SAC BPCM and SAC ASCM encoder-modulator schemes are illustrated in Fig. 26, 27.

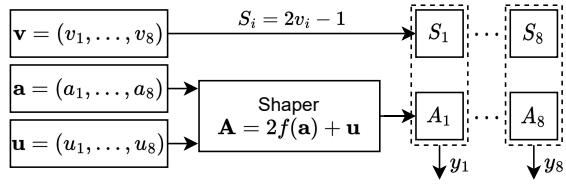

Figure 26. Block-scheme of BPCM-SAC-ECC encoder used with 8-PAM

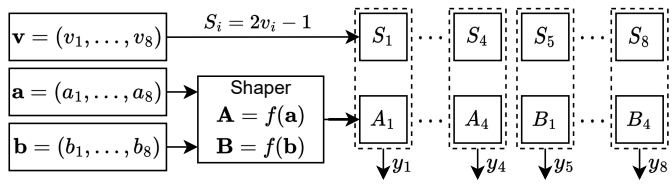

Figure 27. Block-scheme of ASCM-SAC-ECC encoder used with 8-PAM

The first scheme is based on the BPCM mapping in [Boc+23c]. It is applied to the three sequential NB symbols $\boldsymbol{v}$, $\boldsymbol{a}$, and $\boldsymbol{u}$. Bits of $\boldsymbol{v}$ are converted into signs of the eight PAM signals. Bits of $\boldsymbol{a}$ are considered as binary indices of 256×8 shaping code book $\mathcal{B}_1$. The output vector of $\mathcal{B}_1$ is $f(\boldsymbol{a})$. Together with unshaped $\boldsymbol{u}$, it is converted into the eight amplitudes of PAM signals $y_1, ..., y_8$. If the scheme in Fig. 26 is used, then only one out of two amplitude bits of the PAM signal is subjected to shaping.

In the scheme shown in Fig. 27, shaping is combined with ASCM mapping. In this case, two amplitude bits are shaped. The vector $\boldsymbol{a}$ is considered as the binary index of the vector in the 256×4 shaping book $\mathcal{B}_2$. The output of $\mathcal{B}_2$ is length four vector $f(\boldsymbol{a})$. The amplitudes $A_1, ..., A_4$ of the first four PAM signals $y_1, ..., y_4$ are obtained as elements of $f(\boldsymbol{a})$. Similarly, the symbol $\boldsymbol{b}$ is shaped and converted into the amplitudes $B_1, ..., B_4$ of the next four PAM signals.

For $m = 6$, when using the NB LDPC code with 8-PAM, a shaping scheme is constructed based on the SICM mapping shown in Fig. 28. The first and the fourth bits of the six-bit long representation of the NB symbol are converted into the sign bits of two PAM signals y_1 and y_2. The second and third bits and the fifth and sixth bits are shaped by using a 16×2 shaping book $\mathcal{B}_3$ and converted into the amplitudes of y_1 and y_2, respectively.

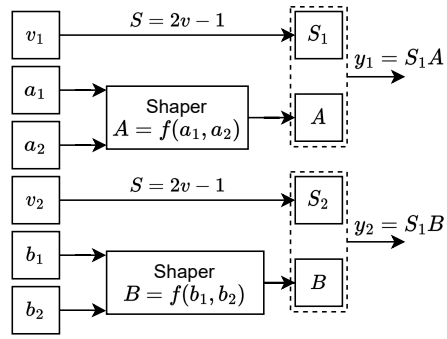

Figure 28. Block-scheme of SICM-SAC-ECC encoder used with 8-PAM

4.5. Bounds, Simulation and Comparison

In this section, an approach is presented for estimating the SED spectrum of ensembles of NB LDPC codes with both SBC and SAC, using a two-variable MGF. The methods are further refined to specific types of coded modulation used, allowing for theoretical analysis and comparison of communication systems using NB LDPC codes used with shaped signaling.

Analagous to the case of BPSK modulation (Sec. 3.3.2) with NB and GLDPC codes, the estimated SED spectra are used for the computation of the TS bound (Sec. 1.6.2).

Lastly, comparisons of SBC and SAC, as well as different types of coded modulation are provided by simulation of decoding performance.

4.5.1. SED Spectra for Ensembles of Irregular NB LDPC Codes with Shaped Coded Modulation

The SED spectrum (SEDS) of a coded PAM signal set can be estimated based on a two-variable generating function of the Hamming and squared Euclidean distances. As described in Sec. 1.6.2, Poltyrev's TS bound is one of the tightest bounds on the ML decoding performance. Based on the SED spectrum, this allows for comparative analysis of combined codes and modulation schemes. While ML decoding is not utilized in practical decoding, relative performance differences of the bounds between studied codes hold for their BP decoding performance.

Let $\mathcal{B} = \{(p_i, \boldsymbol{x}_i, \boldsymbol{b}_i)\}$, $i = 1, 2, \ldots, M^\ell$, be a set of signal sequences $\boldsymbol{x}_i \in X^l$ of length l, where $\boldsymbol{b}_i \in \{0,1\}^{p\ell}$ is a binary index of $\boldsymbol{x}_i$ and p_i is its probability. A two-variable MGF of the pairwise Hamming and Euclidean distances $\left(d_{\mathrm{H}}(\boldsymbol{b}_i, \boldsymbol{b}_j), d^2_{\mathrm{E}}(\boldsymbol{x}_i, \boldsymbol{x}_j)\right)$ is defined on the set $\mathcal{B}$ as

$$g(\lambda, \rho) = \sum_{i=1}^{M^\ell} \sum_{j=1}^{M^\ell} p_i p_j \lambda^{d_{\mathrm{H}}(\boldsymbol{b}_i, \boldsymbol{b}_j)} \rho^{d^2_{\mathrm{E}}(\boldsymbol{x}_i, \boldsymbol{x}_j)}. \tag{4.8}$$

For a code length n divisible by ℓ, the two-variable MGF for sequences of $N = n/\ell$ blocks of signal sequences $\boldsymbol{x}$ is given as

$$G^{(N)}(\lambda, \rho) = \sum_h \sum_e G_{h,e}^{(N)} \lambda^h \rho^e = g(\lambda, \rho)^N, \tag{4.9}$$

where $G_{h,e}^{(N)}$ is a joint probability of the Hamming distance h and SED e.

In order to compute the SED spectrum of the coded shaped and non-shaped PAM signal sets, a conditional MGF of the SED spectrum for sequences of blocks of signal sequences $\boldsymbol{x}$ whose index sequences are at Hamming distance h is introduced as

$$\Theta_h^{(N)}(\rho) = \sum_e \rho^e \theta_{e|h}^{(N)} = \sum_e \rho^e \frac{G_{h,e}^{(N)}}{\sum_d G_{h,d}^{(N)}}.$$

Let $A_{\mathrm{H}}(s) = \sum_{h=0}^{n} A_h s^h$ denote the Hamming weight enumerator of a linear $[n, k]$-code or the average Hamming weight enumerator of an ensemble of linear $[n, k]$-codes. By assuming that nonzero bits of all codewords of weight h are uniformly distributed over n positions, the dependence of $\Theta_h^{(N)}(\rho)$ on the choice of the pair of coded signal sequences at Hamming distance h will be ignored. This assumption, called *perfect interleaver (PI) assumption*, can be interpreted as a conjecture of existence of a perfect interleaver between the ECC encoder and the modulator. Then the average over code bit permutations SED spectrum can be represented as follows

$$A_w(\rho) = \sum_{h=0}^{n} A_h \Theta_h^{(N)}(\rho). \tag{4.10}$$

Notice that according to Eq. (4.7), for the SBC scheme, an $[n, k']$ ECC code $\mathcal{C}'$ of higher rate $R_{\mathrm{ECC}} > R$ is used. Its nonlinear subcode consisting of 2^k low-energy codewords is used for data transmission. By bounding from above the subcode spectrum by spectrum of $\mathcal{C}'$, an estimate of the SED spectrum is obtained as

$$A'_w(\rho) \le \sum_{h=0}^{n} A'_h \Theta_h^{(N)}(\rho), \tag{4.11}$$

where A'_h are Hamming weight enumerators of $\mathcal{C}'$.

Non-shaped PAM signaling. Consider 4-PAM signaling used with Gray mapping. The set $\mathcal{B}$ has the form

$$\mathcal{B} = \{(1/4, -1, 00), (1/4, -3, 01), (1/4, 1, 10), (1/4, 3, 11)\}.$$

Table 4. Examples of shaping books for SAC

M	Book	Signal amplitudes $\boldsymbol{a} = (a_1, a_2, a_3, a_4)$
4	$\mathcal{B}_{a1}$	1111;1113;1131;1133;1311;1313;1331;1333; 3111;3113;3131;3133;1311;3113;3311;3331
6	$\mathcal{B}_{a2}$	1111;1113;1115;1133;1131;1151;1331;1311; 1511;1111;5111;3111;3113;1313;3311;3131

The MGF $g(\lambda, \rho)$ is determined as

$$g(\lambda, \rho) = \frac{1}{4} + \frac{3}{8}\lambda\rho^4 + \frac{1}{8}\lambda\rho^{36} + \frac{1}{4}\lambda^2\rho^{16}.$$

SBC. Consider 4-PAM signaling used with optimized probability distribution $\boldsymbol{p} = (\alpha, 1-\alpha)$, where α denotes the probability of the a-bit of high-energy signals -3 and 3 be equal to one. The s-bits of 4-PAM signals are distributed uniformly. The set $\mathcal{B}$ is determined as follows

$$\left\{\left(\frac{1-\alpha}{2}, -1, 00\right), \left(\frac{\alpha}{2}, -3, 01\right), \left(\frac{1-\alpha}{2}, 1, 10\right), \left(\frac{\alpha}{2}, 3, 11\right)\right\}.$$

The MGF has the form

$$g(\lambda, \rho) = (\alpha^2 + (1-\alpha)^2 + 2\alpha(1-\alpha) + (1-\alpha)^2\lambda\rho^4)/2$$
$$+\alpha^2\lambda\rho^{36}/2 + \alpha(1-\alpha)\lambda^2\rho^{16}.$$

SAC. Consider sequences of 4-PAM signal blocks of length $\ell = 4$. The binary indices $\boldsymbol{b}_i$, $i = 1, 2, ..., M^\ell$ are of length 8. Let $\boldsymbol{a}_i = (a_{i1}, a_{i2}, a_{i3}, a_{i4})$ be the amplitudes of $\boldsymbol{x}_i = (x_{i1}, x_{i2}, x_{i3}, x_{i4})$. Ordered according to their binary indices, elements $\boldsymbol{a}_i$, $i = 1, 2, ..., 2^\ell$ of the shaping books $\mathcal{B}_{ai}$, are given in Table 4. The shaping book $\mathcal{B}_{a1}$ is obtained by M2O method, and $\mathcal{B}_{a2}$ requires the alphabet extension to $M = 6$.

For example, consider $\boldsymbol{b} = (0, 1, 0, 1, 0, 0, 0, 0)$. Extracting the first 4 bits and obtain a binary representation of value 5, using this value as the index 5 of the codeword in $\mathcal{B}_{a1}$. The corresponding entry is $\boldsymbol{a} = (1, 3, 1, 3)$. By using the second group of 4 s-bits in $\boldsymbol{b}$, $\boldsymbol{x} = (-1, -3, -1, -3)$ is obtained.

Notice that some large-weight sequences $\boldsymbol{a}$ of length 4 are replaced by lower-weight sequences. Since $\mathcal{B}_{a1}$ is obtained by M2O mapping, not all entries of the shaping code are different. It is easy to verify that the provided probability distribution of the amplitude bit is $p_1 = 0.359$, $p_2 = 0.141$. By using 2^{16} possible pairs of signal blocks of length 4, following MGF $g(\lambda, \rho)$ is obtained

$$g(\lambda, \rho) = 1$$

$$
\begin{aligned}
&+ \quad \lambda\left(1+20\rho^4+3\rho^8+\rho^{12}+7\rho^{36}\right)/32\\
&+ \quad \lambda^2\left(15\rho^4+83\rho^8+15\rho^{12}+23\rho^{16}+12\rho^{20}+...\right)/224\\
&+ \quad \lambda^3\left(68\rho^8+163\rho^{12}+38\rho^{16}+194\rho^{20}+80\rho^{24}+...\right)/896\\
&+ \quad \lambda^4\left(60\rho^{12}+63\rho^{16}+84\rho^{20}+294\rho^{24}+60\rho^{28}+...\right)/1120\\
&+ \quad \lambda^5\left(18\rho^{16}+106\rho^{20}+156\rho^{24}+24\rho^{28}+172\rho^{32}+...\right)/896\\
&+ \quad \lambda^6\left(21\rho^{28}+28\rho^{36}+71\rho^{40}+16\rho^{48}+20\rho^{52}+...\right)/224\\
&+ \quad \lambda^7\left(8\rho^{40}+2\rho^{48}+14\rho^{52}+2\rho^{64}+4\rho^{72}+2\rho^{84}\right)/32\\
&+ \quad \lambda^8\left(\rho^{52}+\rho^{64}\right)/2.
\end{aligned}
$$

4.5.2. Mapping-specific 2D MGF for SAC

It follows from Eq. (4.10) and (4.9) that the SEDS of the coded modulation system is determined by the MGF $g(\lambda,\rho)$ computed for one code symbol. In this section, examples of computing $g(\lambda,\rho)$ for different code alphabet sizes and mappings are presented. In order to represent MGF in a compact form, the 8-PAM signal set to have a minimum SED between the PAM signal points is normalized to be equal to one: $\{-\frac{7}{2},...,-\frac{1}{2},\frac{1}{2},...,\frac{7}{2}\}$. For unshaped transmission, the average signal energy for 8-PAM is

$$E_{\text{uni}} = (M^2-1)/12 = 5.25. \tag{4.12}$$

Example 1: MGF for the 2^p-PAM signals can be represented as a matrix of size $(p+1)\times t$, $t=(2^p-1)^2$. Denote $\boldsymbol{\lambda}_p=(1,\lambda,...,\lambda^p)$ and $\boldsymbol{\rho}_p=(1,\rho,\rho^4,...,\rho^{t^2})^{\mathrm{T}}$. Direct computations for the 8-PAM signaling give the following formulas

$$g_{8\text{PAM}}(\lambda,\rho) \;=\; \frac{1}{32}\boldsymbol{\lambda}_3\begin{pmatrix}4&0&0&0&0&0&0&0\\0&7&0&3&0&1&0&1\\0&0&6&0&4&0&2&0\\0&0&0&2&0&2&0&0\end{pmatrix}\boldsymbol{\rho}_3. \tag{4.13}$$

For the code of length $N=n/p$ in PAM signals for unshaped SICM:

$$G_{h,e}^{\text{SICM}}(\lambda,\rho) = g_{8\text{PAM}}^N(\lambda,\rho). \tag{4.14}$$

Example 2: Consider the unshaped BPCM. For the 8-PAM signaling, the bit-wise MGFs for the sign bit $g_s(\lambda,\rho)$ and two amplitude bits $g_a(\lambda,\rho)$, and $g_u(\lambda,\rho)$, respectively, are equal to

$$g_s(\lambda,\rho) \;=\; \frac{1}{8}(4+\lambda(\rho+\rho^4+\rho^9+\rho^{16})), \tag{4.15}$$

$$g_a(\lambda,\rho) \;=\; \frac{1}{4}(2+\lambda(\rho+\rho^4)), \tag{4.16}$$

$$g_u(\lambda,\rho) \;=\; \frac{1}{2}(1+\lambda\rho). \tag{4.17}$$

For the code of length N the following is obtained:

$$G_{h,e}^{\mathrm{BICM}}(\lambda,\rho) = [g_s(\lambda,\rho)g_a(\lambda,\rho)g_u(\lambda,\rho)]^N. \tag{4.18}$$

Consider PAM signaling used with SAC. For each $\boldsymbol{b} \in \mathcal{B}$, considering all possible combinations of sign bits, there are 2^ℓ different signal sequences $\boldsymbol{s} \in \mathcal{S}$ consisting of the same amplitude bits as $\boldsymbol{b}$. Denote by $g_{\mathcal{B}}(\lambda,\rho)$ and $g_{\mathcal{S}}(\lambda,\rho)$ the corresponding MGFs Eq. (4.8) over the sets $\mathcal{B}$ and $\mathcal{S}$, respectively. Then for the shaped SICM (Fig. 28) and ASCM (Fig. 27), the Eq. (4.9) becomes

$$G_{h,d}^{\mathrm{SICM-SAC}}(\lambda,\rho) = g_{\mathcal{S}}(\lambda,\rho)^N. \tag{4.19}$$

$$G_{h,d}^{\mathrm{ASCM-SAC}}(\lambda,\rho) = \left[g_{\mathcal{B}}(\lambda,\rho)g_s(\lambda,\rho)^\ell\right]^N, \tag{4.20}$$

where $N = n/pl$, $g_s(\lambda,\rho)$ is determined by Eq. (4.15).

Similarly, for the shaped BPCM in Fig. 27:

$$G_{h,e}^{\mathrm{BPCM-SAC}}(\lambda,\rho) = \left[g_{\mathcal{B}}(\lambda,\rho)g_u(\lambda,\rho)^l g_s(\lambda,\rho)^l\right]^N,$$

where $g_s(\lambda,\rho)$ is determined by Eq. (4.15), and $g_u(\lambda,\rho)$ is determined by Eq. (4.17).

4.5.3. ML Decoding Bound of NB LDPC Codes with SBC and SAC

First, consider a toy example of a short linear code. For this code the "true" pairwise SED spectrum can be computed by an exhaustive search over all pairs of modulated and shaped codewords. By comparison of the "true" spectra and its estimate Eq. (4.10) the influence of the PI assumption on the precision of the SED spectrum estimate can be verified.

Next, consider the $[24, 18]$ binary linear code with minimum distance $d = 4$. Its Hamming weight generating function is

$$A(s) = 378s^4 + 4032s^6 + 23439s^8 + 60480s^{10} + \dots \ .$$

For SBC, a higher rate $[24, 19]$-code with $d = 3$ is used. Its Hamming weight generating function is

$$A(s) = 71s^3 + 371s^4 + 1288s^5 + 4088s^6 + 10949s^7 + \dots \ .$$

First coefficients of the "true" and estimated SED spectrum for three communication scenarios are presented in Table 5.

In Fig. 29, the TS bounds Eq. (1.20) for coded 4-PAM signaling without shaping, with SBC, and with SAC are shown. The presented bounds are computed by using the estimated SED spectra from Table 5. For comparison, simulated ML decoding FER performance for the three scenarios is given in

Table 5. Weight enumerators and estimates for $R = 3/4$, $n = 24$

Scenario	SED spectrum	
No shaping	True	$91\rho^{16} + 365\rho^{24} + 1011\rho^{32} + \ldots$
	Estimated	$89\rho^{16} + 368\rho^{24} + 1007\rho^{32} + \ldots$
SBC	True	$23\rho^{12} + 81\rho^{16} + 190\rho^{20} + \ldots$
	Estimated	$36\rho^{12} + 134\rho^{16} + 318\rho^{20} + \ldots$
SAC	True	$4\rho^{8} + 8\rho^{12} + 77\rho^{16} + 110\rho^{20} + \ldots$
	Estimated	$3\rho^{8} + 8\rho^{12} + 77\rho^{16} + 108\rho^{20} + \ldots$

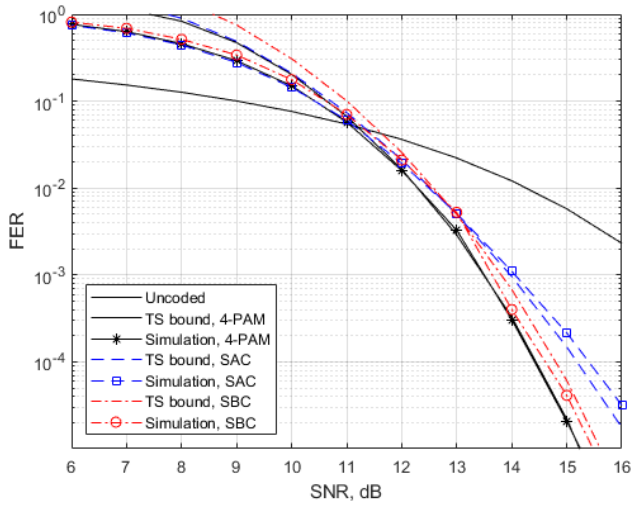

Figure 29. Simulation results and TS bounds for coded modulation without shaping, with SBC, and with SAC, $n = 24$, $R_{\mathrm{T}} = 3/2$ bits/dimension

Fig. 29. It follows from the plots in Fig. 29 that the TS bounds based on the estimated SED spectra are rather tight. However, for $n = 24$ shaping does not improve performance since for such a short code shaping gain is not large enough to overcome the negative impact of shaping on the code performance (reducing minimum distance for SBC or additional errors for SAC).

Fig. 30 shows the TS bound with and without shaping for the ensemble of rate $R = 3/4$ and length $n \approx 2000$ NB LDPC codes along with simulation results of the BP decoding for a set of optimized, as in [Boc+22], NB QC-LDPC codes of the same rate and length. Three scenarios are considered: no shaping, SBC, and SAC. In Fig. 30, the BP decoding performance of the binary LDPC code in the 5G standard is presented for comparison. It follows from the presented plots that:

- The random coding bound on the ML decoding error probability for coded 4-PAM modulation with SBC and with SAC gain about 1 dB and 0.5 dB, respectively, compared to the non-shaped coded modulation.
- The simulation of the BP decoding for NB QC-LDPC codes shows 0.5 dB gain of the shaped transmission with respect to the non-shaped

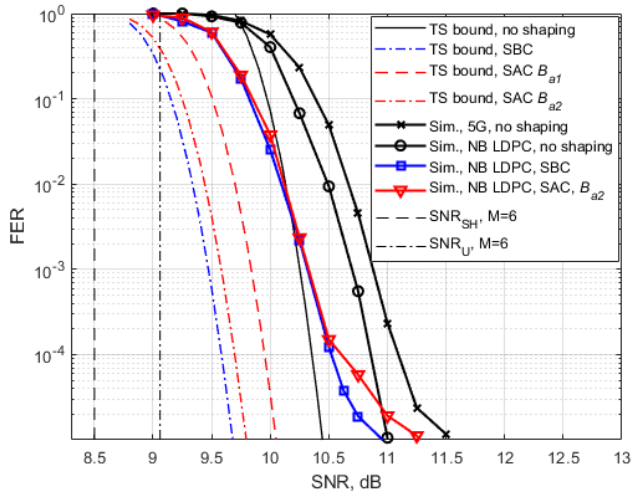

Figure 30. Simulation results and TS bounds for coded modulation without shaping, with SBC, and SAC, $n \approx 2000$, $R_{\mathrm{T}} = 3/2$ bits/dimension

one. The obtained gain does not depend on the type of shaping.

4.5.4. Decoding Performance of NB LDPC Codes with Shaped and Unshaped Coded Modulation

Numerical comparison of the shaping schemes in Fig. 26 and Fig. 27 for the NB LDPC code of length $n = 512$ over GF(2^8) used with 8-PAM signaling is presented in Fig. 31. It follows from the FER performance plots of the BP decoding that BPCM-SAC based shaping outperforms ASCM-SAC based technique. The shaping gain with respect to the unshaped signaling is about 0.25 and 0.5 dB for BPCM-SAC and ASCM-SAC, respectively.

Simulation results for the FER of BP decoding for the NB LDPC coded unshaped and shaped 8-PAM signals are shown in Fig. 32. The NB LDPC codes over GF(2^8) in [Dol+14] of length n =128, 256, 512, were simulated with maximum 100 iterations of BP decoding. Simulations were performed up to 100 error events. The achieved gain over uniform signaling varies from 0.25 dB for $n = 128$ to 0.5 dB for $n = 512$.

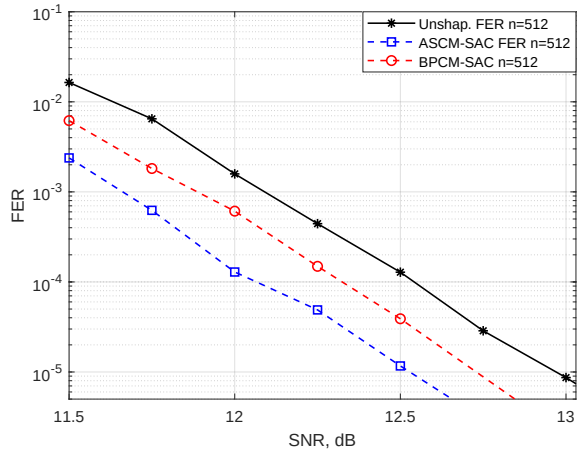

Figure 31. The FER performance of BP decoding of unshaped 8-PAM vs ASCM/BPCM-SAC for the NB LDPC code over GF(2^8)

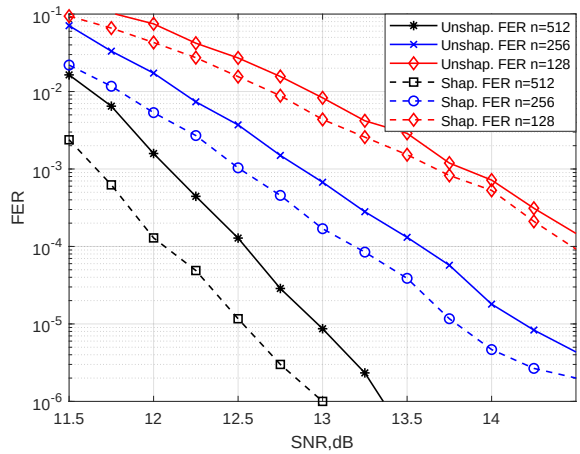

Figure 32. The FER performance of BP decoding of unshaped 8-PAM vs ASCM-SAC for the NB LDPC code over GF(2^8)

5. OPTIMIZATION OF SHAPED CODED MODULATION

5.1. Cost Function Based Optimization of SAC

In this section, a new algorithm for optimization of shaping books $\mathcal{B}_a$ is presented, as originally proposed in [Boc+23b]. For complexity reasons, similarly to the previous works in this area, the optimization of the used ECCs was not considered in the optimization procedure proposed in the paper. The influence of the ECC on the shaping parameters is partially taken into account by the optimization criterion. Similarly to the Gray mapping used for one-dimensional signal sets, the optimization process prioritizes shaping books which correspond to a larger SED for a given Hamming distance between codewords. The MGF Eq. (4.8) can be rewritten as

$$g(\lambda, \rho) = \sum_{d_{\mathrm{H}}} \lambda^{d_{\mathrm{H}}} \sum_{d_{\mathrm{E}}^2} \Pr(\mathrm{d_H}, \mathrm{d_E^2}) \rho^{\mathrm{d_E^2}}, \tag{5.1}$$

where $\Pr(\mathrm{d_H}, \mathrm{d_E^2})$ denotes a joint probability of the event that the Hamming distance between indices $\boldsymbol{b}$ is equal to d_{H} and the SED between the corresponding signal blocks $\boldsymbol{x}$ is equal to d_{E}^2. The optimization algorithm minimizes the cost function

$$\mu(\mathcal{B}_a) = \sum_{d_{\mathrm{H}}} \sum_{d_{\mathrm{E}}^2 \leq 4 d_{\mathrm{H}}} \Pr(d_{\mathrm{H}}, d_{\mathrm{E}}^2). \tag{5.2}$$

The main steps of the algorithm are given in Alg. 4.

5.2. Two-Step Optimization Procedure

A particularly challenging task in the topic of shaping, in the context of NB LDPC codes, is to obtain a noticeable shaping gain at short and moderate code lengths. In [BKM24], which this section is based on, NB LDPC codes over GF(2^6) and GF(2^8), as proposed in [CSN17; Dol+14], at a transmission rate of 1.5 bits/PAM signal were studied. These codes are recommended for satellite navigation [CSN17] and transmitting commands in satellite communications [Spa15]. Note that a potential gain for this scenario is about 0.55 dB.

In order to achieve a noticeable shaping gain in this setting, a refined analysis of the SEDS of the coded PAM signal system used with SAC and different mappings is proposed. It is empirically confirmed in [Boc+23c] that despite the suboptimality of BP decoding, the SEDS can serve as an optimization criterion for signal systems combined with NB LDPC codes.

To reduce the computation complexity of the optimization procedure, it is split into two steps, as follows. Firstly, by utilizing a genetic algorithm

Algorithm 4: SED-based shaping book optimization

Input: $\mathcal{B}_{\text{init}}, I_{\max}$;
// Chosen capacity-optimal length-l vectors, $|\mathcal{B}_{\text{init}}| = 2^l$
$\mathcal{B} = \mathcal{B}_{\text{init}}, I = 0, F = 1, \hat{\mu} = \infty$; // Initialize
while $F = 1$ & $I < I_{max}$ **do**
  $F = 0$; // Success flag
  $\hat{\mu} = \mu(\mathcal{B})$; // Score (Eq. (5.2))
  Choose a list $\mathcal{L}$ of pairs $(\boldsymbol{a}_i, \boldsymbol{a}_j) \in \mathcal{B}$ with the smallest ratio $d_{\text{E}}^2(\boldsymbol{a}_i, \boldsymbol{a}_j)/d_{\text{H}}(\boldsymbol{a}_i, \boldsymbol{a}_j)$;
  // Find pairs with smallest pairwise distance ratio
  for $(\boldsymbol{a}_i, \boldsymbol{a}_j) \in \mathcal{L}$ **do** // Iterate over pairs
    $\mathcal{B}' = \mathcal{B}$; // Temporary copy
    Choose a vector $\boldsymbol{a}_t \in \mathcal{B}$ such that $d_{\text{H}}(\mathbf{a}_i, \mathbf{a}_t) \leq t$;
    Swap $\boldsymbol{a}_i, \boldsymbol{a}_t$ in $\mathcal{B}'$;
    $\hat{\mu}' = \mu(\mathcal{B}')$;
    if $\hat{\mu}' < \hat{\mu}$ **then**
      $\mathcal{B} = \mathcal{B}', \hat{\mu} = \hat{\mu}', F = 1$;
      go to 4;
  if $F = 1$ **then**
    $I = I + 1$;
return $\mathcal{B}$;

to optimize the so-called "shaping books" for 8-PAM signals shaped with SAC, which are used in conjunction with the NB LDPC codes over GF(2^8) and GF(2^6). Next, the constructed shaping books can be improved by using an approximation of the SEDS for the coded modulation system.

5.2.1. Optimization by Genetic Algorithm

In this section, a new method of shaping book $\mathcal{B} = \{\boldsymbol{b}_i\}, i \in \{0, ..., M^\ell - 1\}$, optimization is introduced, using a specialized genetic algorithm. The algorithm is designed to overcome issues of local extrema encountered by deterministic optimization algorithms and to have a low computational complexity required for optimization of large shaping books. The steps of the algorithm can be summarized as:

1. The starting population is defined, consisting of books that are permutations of the same row vectors $\{\boldsymbol{b}_i\}$.
2. The population is assessed and the highest-scoring (based on Eq. (5.3)) books are chosen for crossover.
3. Crossover of two randomly paired parent books randomly selects a row vector $\boldsymbol{b}_i$ from either parent for each position of the new book. Duplicate row vectors are replaced by randomly assigned missing vectors.

The procedure is repeated to produce a predetermined number of new books from each parent pair. To introduce additional randomness to each genetic iteration, crossover is repeated for both parents using a book of randomly permuted row vectors.

4. Mutation of new books from the crossover procedure has a probability μ of swapping the positions of a pair of row vectors of a produced book.
5. The new population, consisting of the previous generation and all new books produced by crossover, is evaluated, and reduced to a defined maximum size, keeping only the highest-scoring books of the generation.

The number of genetic iterations (steps $2-5$) used for optimization depends on the size of the shaping book. Additional crossover using random permutations in step 3) increases the number of iterations that can be performed without converging to a stagnant population.

The quality of the produced shaping books is dependent on the scoring function

$$\text{score} = \sum_{w=1}^{\ell} \frac{\gamma(w)}{w} \binom{\ell}{w}^{-1} \sum_{(i,j):d_H(i,j)=w} d_{\mathrm{E}}^2(\boldsymbol{b}_i, \boldsymbol{b}_j), \tag{5.3}$$

where $\gamma(w)$ is a weighting function assigning larger weights to smaller Hamming distances. A simple linear weighting function $\gamma(w) = \ell - w + 1$ is used. To reduce computational complexity, signs of signal sets (book row vectors $\boldsymbol{b}_i$) are excluded from calculations.

5.2.2. SED-Based Optimization

Previously, a limiting factor for SED-based optimization (Sec. 5.2.2) was the complexity of SED spectrum (SEDS) computation according to Eq. (4.10), with order n^4 complexity of Eq. (4.9) in particular. In order to analyze SEDS and, especially, for involving SED in the shaping book optimization process, an approximation method of Eq. (4.9) is proposed, by using a generalization of saddle point approximation ([LR80], [See92]) for the PDF of the sum of independent random vectors.
The MGF Eq. (4.8) can be rewritten as

$$g(\lambda, \rho) = \sum_{w=0}^{mp} \sum_{\ell \geq 0} \pi_{w,\ell} \lambda^w \rho^\ell, \tag{5.4}$$

and in the vector form

$$g(\boldsymbol{s}) = \sum_{\boldsymbol{x}} \pi(\boldsymbol{x}) e^{(\boldsymbol{s},\boldsymbol{x})} = \boldsymbol{E}[e^{(\boldsymbol{s},\boldsymbol{x})}], \tag{5.5}$$

where $\pi_{w,\ell}$ denotes a joint probability of the Hamming distance w between indices and the SED ℓ between the corresponding signal sequences, $\boldsymbol{x} = (w, \ell)$, $\boldsymbol{s} = (\ln \lambda, \ln \rho)$, and $(\cdot, \cdot)$ denotes the scalar product of the two vectors.

Consider Eq. (4.9) as the MGF of the sum $\boldsymbol{y} = \sum_{i=1}^{N} \boldsymbol{x}_i$ of N independent identically distributed random vectors $\boldsymbol{x}_i$ with probability distribution $\pi(\boldsymbol{y})$. This MGF can be expressed as

$$G_{h,d}(\boldsymbol{s}) = g_N(\boldsymbol{s}) = \sum_{\boldsymbol{y}} \pi(\boldsymbol{y}) e^{(\boldsymbol{s},\boldsymbol{y})}, \tag{5.6}$$

where $\pi(\boldsymbol{y}_0)$ is the probability of a given pair of Hamming and Euclidean distances $\boldsymbol{y}_0$ for the codeword of length N blocks. Similar to the one-dimensional case in [Gal68, Appendix 5A], the so-called *tilted* distribution on $\boldsymbol{x}$ is defined as:

$$q_{\boldsymbol{s}}(\boldsymbol{x}) = \frac{\pi(\boldsymbol{x}) e^{(\boldsymbol{s},\boldsymbol{x})}}{\sum_{\boldsymbol{x}'} \pi(\boldsymbol{x}') e^{(\boldsymbol{s},\boldsymbol{x}')}} = \frac{\pi(\boldsymbol{x}) e^{(\boldsymbol{s},\boldsymbol{x})}}{g(\boldsymbol{s})}. \tag{5.7}$$

Denote as $\mu(\boldsymbol{s}) = \ln g(\boldsymbol{s})$. It is easy to show that $\frac{d\mu(\boldsymbol{s})}{d\boldsymbol{s}} = \boldsymbol{E}_{\boldsymbol{s}}[\boldsymbol{x}]$ and the Hessian of $\mu(\boldsymbol{s})$ is

$$\boldsymbol{H}(\boldsymbol{s}) = \left\{ \frac{\partial^2 \mu(\boldsymbol{s})}{\partial s_i \partial s_j} \right\} = \boldsymbol{E}_{\boldsymbol{s}}[\boldsymbol{x}^{\mathrm{T}} \boldsymbol{x}] - \boldsymbol{E}_{\boldsymbol{s}}[\boldsymbol{x}^{\mathrm{T}}] \boldsymbol{E}_{\boldsymbol{s}}[\boldsymbol{x}] = \mathrm{Cov}_{\boldsymbol{s}}(\boldsymbol{x}), \tag{5.8}$$

where $\boldsymbol{E}_{\boldsymbol{s}}[\cdot]$ denotes the mathematical expectation on the tilted distribution. Then, for the variable $\boldsymbol{y}$, its MGFs $g_N(\boldsymbol{s})$, $\mu_N(\boldsymbol{s}) = \ln g_N(\boldsymbol{s})$, and the Hessian matrix $\boldsymbol{H}_N(\boldsymbol{s})$ are

$$g_N(\boldsymbol{s}) = g(\boldsymbol{s})^N, \quad \mu_N(\boldsymbol{s}) = N\mu(\boldsymbol{s}), \tag{5.9}$$

$$\boldsymbol{H}_N(\boldsymbol{s}) = N\boldsymbol{H}(\boldsymbol{s}). \tag{5.10}$$

Next, to express the distribution on $\boldsymbol{y}$ via the tilted distribution $q_{\boldsymbol{s},N}(\boldsymbol{y})$, we have:

$$q_{\boldsymbol{s},N}(\boldsymbol{y}) = \frac{\pi_N(\boldsymbol{y}) e^{(\boldsymbol{s},\boldsymbol{y})}}{g_N(\boldsymbol{s})}. \tag{5.11}$$

$$\pi_N(\boldsymbol{y}) = q_{\boldsymbol{s},N}(\boldsymbol{y}) e^{N\mu(\boldsymbol{s}) - (\boldsymbol{s},\boldsymbol{y})}. \tag{5.12}$$

Choose $\boldsymbol{s} = \boldsymbol{s}_0$ as a solution of the equation

$$\mu'(\boldsymbol{s}_0) = \frac{d\mu(\boldsymbol{s})}{d\boldsymbol{s}}|_{\boldsymbol{s}=\boldsymbol{s}_0} = \frac{\boldsymbol{y}_0}{N}. \tag{5.13}$$

Note that $\boldsymbol{y}$ is the sum of numerous independent vectors, and therefore the tilted distribution $q_{\boldsymbol{s}_0,N}$ can be approximated as the multivariable normal distribution $\mathcal{N}(\boldsymbol{y}_0, N\boldsymbol{H}(\boldsymbol{s}_0))$ and

$$q_{\boldsymbol{s}_0,N}(\boldsymbol{y}_0) \approx \frac{1}{(2\pi)^{t/2} N (\det(\boldsymbol{H}(\boldsymbol{s}_0))^{1/2}},$$

where t is the dimension of $\boldsymbol{y}$. The final expression for $\pi_N(\boldsymbol{y}_0)$ is

$$\pi_N(\boldsymbol{y}_0) \approx \frac{e^{N(\mu(\boldsymbol{s}_0)-(s_0,\mu'(\boldsymbol{s}_0))}}{(2\pi)^{t/2}N(\det(\boldsymbol{H}(\boldsymbol{s}_0))^{1/2}}, \tag{5.14}$$

where $\boldsymbol{y}_0$ and $\boldsymbol{s}_0$ satisfy Eq. (5.13).

For the signal set $\mathcal{S}$ in the schemes in Figs. 26-28, the average energy is computed as $E_{\mathcal{S}} = M^{-\ell}\sum_i ||\boldsymbol{s}_i||^2$. Coding gain for a given scheme is equal to $g_{\mathcal{S}} = E_{\text{uni}}/E_{\mathcal{S}}$. We measure the minimum SED d^2_{minSED} as the first nontrivial index e, such that $\text{SEDS}(e) \geq 1$. For a fair comparison of schemes with different gains, the *equivalent squared Euclidean distance* (ESED) is used, defined as

$$d^2_{\text{eq}} = d^2_{\text{minSED}} \times \text{g}_{\mathcal{S}}. \tag{5.15}$$

By substituting the optimization criterion (cost function), given in Eq. (5.2) used in Alg. 4, with maximization of the minimum SED, the fast computation for estimation of the SEDS based on saddle point approximation allows for increased refinement in optimization.

5.3. Simulation and Comparison

5.3.1. Cost Function Based Optimization

In this section, examples of shaping books are given for SAC optimized by Algorithm 4. Sequences of 4-PAM signal blocks of length $\ell = 5$ with binary indices $\boldsymbol{b}_i$, $i = 0, 1, ..., 2^{2\ell} - 1$, of length $p\ell = 10$ are considered. Let $\boldsymbol{a}_i = (a_{i,1}, a_{i,2}, a_{i,3}, a_{i,4}, a_{i,5})$ be a-bits of $\boldsymbol{x}_i = (x_{i,1}, x_{i,2}, x_{i,3}, x_{i,4}, x_{i,5})$. Ordered according to their binary indices $\boldsymbol{\beta}_i$ of length $\ell = 5$, the codewords $\boldsymbol{a}_i$, $i = 0, 1, ..., 2^{\ell} - 1$, are given in Table 6.

Table 6. Examples of shaping books for SAC matched with NB LDPC codes over GF(2^5) used with 4-PAM signaling

M	Book	Signal amplitudes $\boldsymbol{a} = (a_1, a_2, a_3, a_4, a_5)$
6	$\mathcal{B}_{a1}$	11111; 11113; 11115; 11133; 11131; 11151; 11331; 11333; 11313; 11311; 11511; 13111; 15111; 13113; 13133; 13131; 13331; 13311; 13313; 31313; 31113; 31133; 31131; 31111; 51111; 31311; 31331; 33131; 33113; 33111; 33311 11111;
6	$\mathcal{B}_{a2}$	33311; 11111; 11111; 11113; 11131; 11311; 13111; 31111; 11133; 11331; 11313; 13113; 13131; 13311; 31113; 31131; 31311; 33111; 13313; 11333; 13133; 11115; 13331; 11151; 11511; 31133; 31331; 31313; 15111; 51111; 33131; 33113;

Both the shaping books $\mathcal{B}_{a1}$ and $\mathcal{B}_{a2}$ were obtained by combining M2O with alphabet extension CE to $M = 6$. $\mathcal{B}_{a1}$ was obtained by probability distribution optimization over PAM-signal amplitudes. $\mathcal{B}_{a2}$ was optimized using Algorithm 1. The shaping-modulation mapping is explained in Fig. 33.

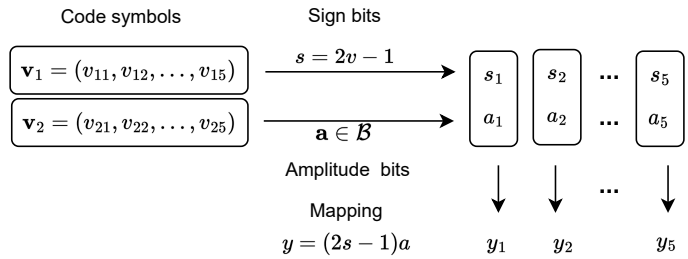

Figure 33. Shaping-modulation mapping for 4-PAM signaling used with NB LDPC codes over GF(2^5)

A sequence of codeword bits is split into blocks of length $\ell = m = 5$. Each odd-numbered block is interpreted as s-bits of $m = 5$ 4-PAM signals. Decimal representations of even-numbered blocks are used as indices of the shaping book entries. For example, let the sequence of codeword bits be $\boldsymbol{b} = (1010111001)$, then the index $\boldsymbol{\beta} = (11001)$ corresponds to the word $\boldsymbol{a}_{25} = (31111)$ in $\mathcal{B}_{a1}$ or $\boldsymbol{a}_{25} = (11151)$ in $\mathcal{B}_{a2}$. The s-bits are 10101, that is, shaped 4-PAM signals $(y_1, y_2, y_3, y_4, y_5)$ are either $(3, -1, 1, -1, 1)$ for $\mathcal{B}_{a1}$ or $(1, -1, 1, -5, 1)$ for $\mathcal{B}_{a2}$.

In Fig. 34, TS bounds are presented for the rate $R = 3/4$ NB LDPC codes of the ensemble in [Boc+22] used without shaping and with SAC 4-PAM signaling. The code length in bits is $n \approx 2000$, the average column weight for the ensemble codes is $w = 2.5$. In the same figure, the FER performance of the BP decoding is shown for NB QC-LDPC codes with $w = 2.5$ of the same length $n \approx 2000$ and rate $R = 3/4$ used with shaped 4-PAM signaling. For comparison, the FER performance of the unshaped and SBC NB QC-LDPC coded 4-PAM signaling are given. In order to provide transmission rate $R_{\mathrm{T}} = 3/2$ when using coded SBC signaling, the NB QC-LDPC code of rate $R = 4/5$ was simulated. The used NB QC-LDPC codes were optimized by using techniques in [Boc+22]. The maximum number of iterations was restricted by 50.

In the case of 8-PAM signaling, there are a few possible modulation-shaping mappings depending on the parameter m. Fig. 35, shows the FER performance of BP decoding for the mapping which corresponds to M2O shaping according to the scheme in Fig. 33 applied to the most significant a-bit of the 8-PAM signal. The second (least significant) a-bit of the PAM signal is used unshaped. Both shaping books $\mathcal{B}_{a3}$ and $\mathcal{B}_{a4}$ are obtained by the M2O method. The book $\mathcal{B}_{a3}$ is obtained by optimizing probability distribution over PAM-signal amplitudes. Then this shaping book was additionally optimized by Algorithm 4 to obtain the shaping book $\mathcal{B}_{a4}$.

From the presented results, it can be concluded that:

- The TS bound and simulations of the BP decoding confirm that optimization of SAC books provides about 0.2 dB gain in the FER

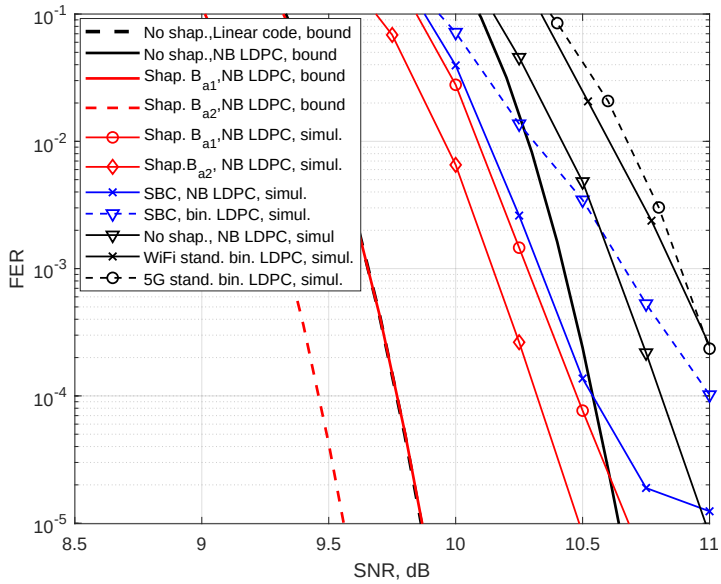

Figure 34. TS bounds and simulation results for general linear and NB QC-LDPC codes over GF(2^5) used with shaped 4-PAM signaling. Theoretical gain is ≈ 0.56 dB (see Table 3)

performance for binary images of NB LDPC codes over GF(2^m). Simulation results in Fig. 34 show that for 4-PAM, the achieved coding gain is rather close to the theoretically predicted value.

- SAC with a proper optimization can outperform SBC regardless if it is applied to rather short blocks and has lower implementation complexity.
- Although the proposed optimization criterion does not directly rely on the code spectra, it reduces the FER for both ML and BP decoding by improving the SED spectrum Eq. (4.10).

5.3.2. Two-Step Optimization

As clearly seen from the plots in Fig. 36, optimization of the shaping book by the genetic algorithm improves the FER performance of the BP decoding in the waterfall region (solid curves). The performance improvement increases with increasing $g_{\mathcal{S}}$. On the other hand, the corresponding SEDS shown in Fig. 37 are worsening with increasing $g_{\mathcal{S}}$ resulting in the FER performance deterioration in the error floor region.

Further optimization of the shaping books by the Alg. 4, according to the criterion of maximizing the minimum SED significantly improves the SEDS of the coded shaped signal set. This leads to the FER performance improvement in the error floor region (Fig. 36, dashed lines).

In Fig. 38, a comparison of the FER performance of BP decoding of the

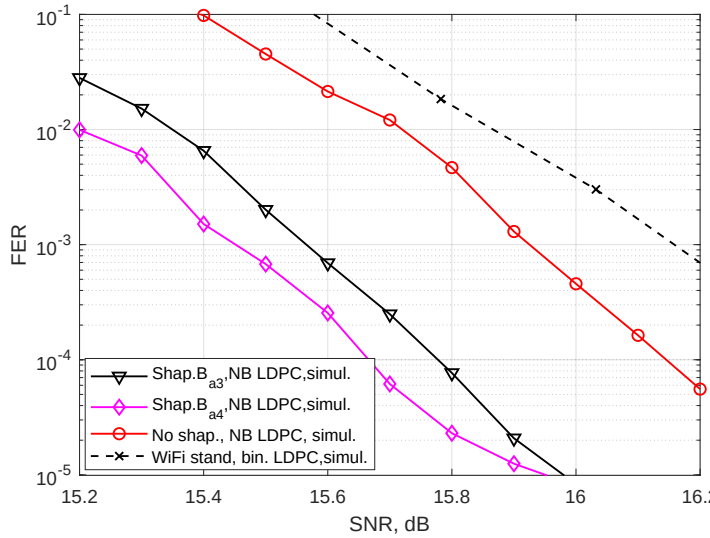

Figure 35. Simulation results for NB QC-LDPC codes over GF(2^4) used with shaped 8-PAM signaling. Theoretical gain is ≈ 0.80 dB (see Table 3)

rate $R = 1/2$ NB LDPC code over GF(2^6) of length $n = 576$ used with and without shaping is given. The achieved shaping gain is about 0.3 dB.

Comparison of optimized ASCM-SAC and two CCDM schemes from [SLK20], together with their $(192, 96)$ 5G LDPC code baseline is provided in Fig. 39. Both decoders from [SLK20] use a combined BP decoder and CCDM processor with computational complexity dependent on code trellis size.

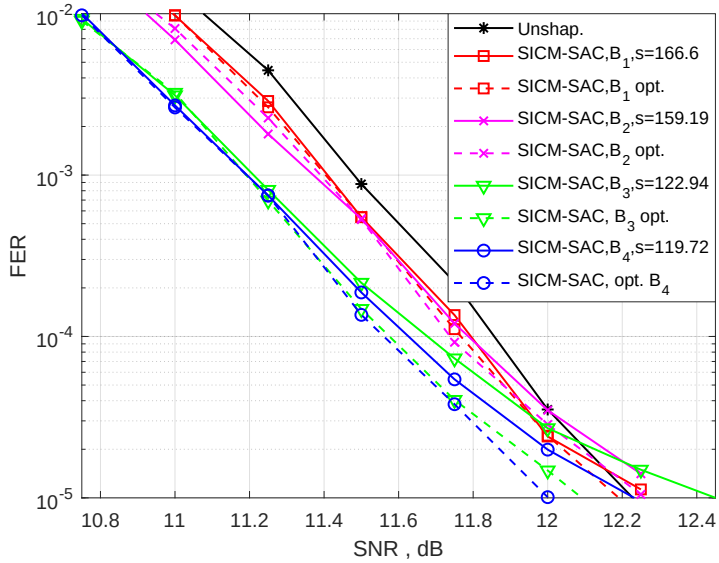

Figure 36. The FER performance of BP decoding of unshaped 8-PAM vs SICM-SAC with optimized books for the NB LDPC code over GF(2^6)

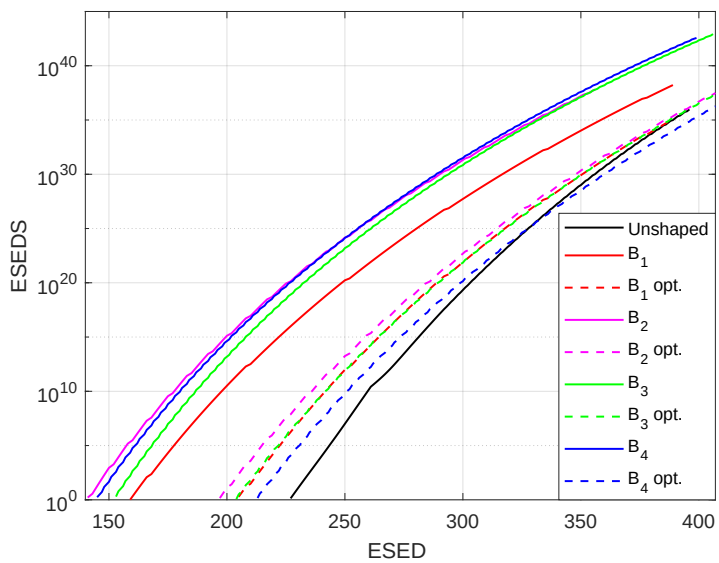

Figure 37. Comparison of unshaped 8-PAM SEDS vs SICM-SAC SEDS for the rate $R = 1/2$ NB LDPC code over GF(2^6)

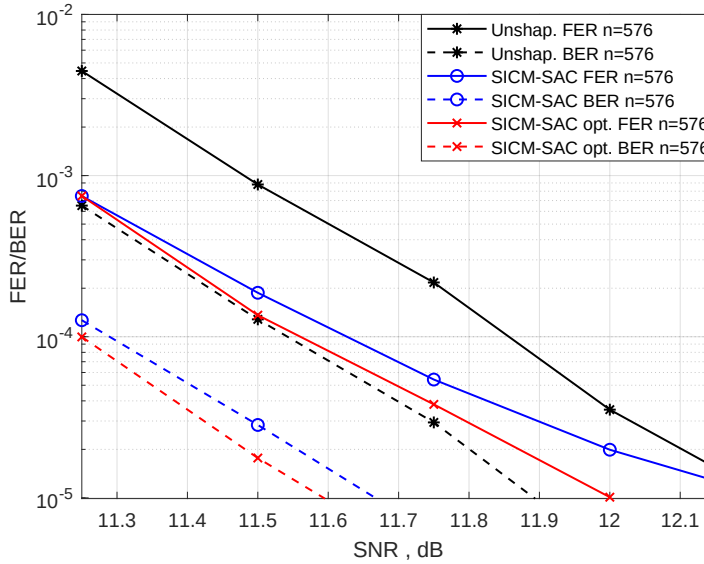

Figure 38. The FER performance of BP decoding of unshaped 8-PAM vs SICM-SAC for the NB LDPC code over GF(2^6)

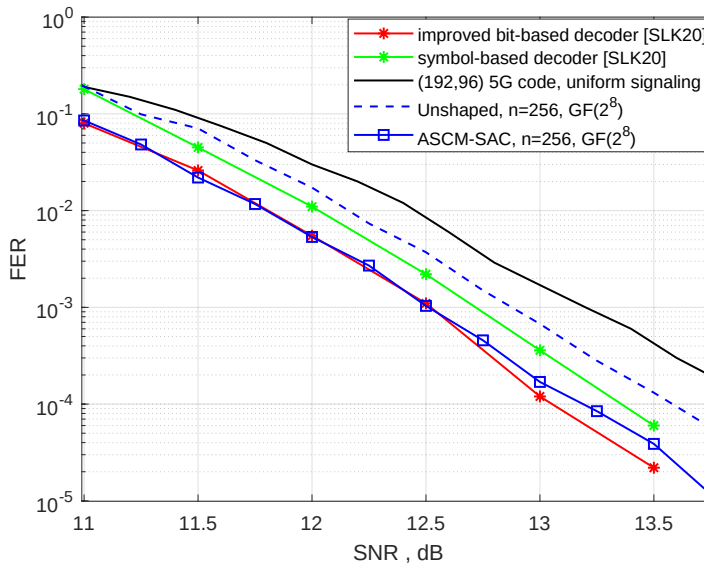

Figure 39. BP decoding FER performance of ASCM-SAC 8-PAM vs two CCDM schemes [SLK20] with transmission rate 1.5 bits/signal dimension.

6. CONCLUSION

The research presented in this dissertation can be broadly categorized as:

- **Code search techniques**,
  including binary, quasi-cyclic, generalized, and non-binary LDPC codes. The presented methods can be used to design codes that surpass those used in modern telecommunications standards in terms of decoding error rates.
- **System design**,
  combining powerful error-correcting codes with tailored modulation and shaping, optimized to specific codes.
- **Analysis**,
  Poltyrev's tangential sphere bound was used to evaluate ML decoding performance. The proposed methods allow the bound to be computed for generalized LDPC codes and non-binary LDPC codes combined with shaped coded modulation.
- **Coding gain in important practical scenarios**,
  all aspects of code and shaped modulation scheme design proposed in the dissertation aim to satisfy the restrictive nature of practically useful communications systems. This entails designing codes and shaped modulation schemes that allow for computationally efficient encoding and decoding processes.

A detailed overview of the work presented in each chapter and the following conclusions are presented below.

In Chapter 2, binary LDPC codes were covered, starting with an introduction to the required preliminaries, including encoding and decoding. A main focus of the chapter, QC-LDPC codes are a powerful class of error-correcting codes, used for further constructions and study in the following chapters of the dissertation.

The methods of code construction and optimization, including choice of code parameters, and algorithms for the search of high-performance QC-LDPC codes, presented in the chapter were further employed for the construction base matrices of generalized- and non-binary LDPC codes – two powerful generalizations of Gallager's LDPC codes.

In Chapter 3, GLDPC codes with low-complexity decoded constituent codes were compared to NB LDPC codes over small alphabets. The finite-length random coding bound on the ML decoding error probability for both classes is rather close, however, the random GLDPC codes outperform the random NB LDPC codes with the same average column weight of their base parity-check matrices. Based on analysis presented in the chapter and Fig. 18, it can be concluded that:

- Both classes – GLDPC codes with low-complexity decoded constituent

codes and NB LDPC codes over small alphabets – outperform their binary counterparts in terms of achieved frame error rates.

- While GLDPC and NB LDPC codes are two competing solutions, the advantage of GLDPC codes is a simpler iterative decoding procedure, which implies exchange of binary log-likelihoods.
- Irregular NB QC-LDPC codes provide better decoding performance than the QC-GLDPC codes with low-complexity decoded constituent codes only if $m \geq 4$, that is, if their decoding complexity is impractically high.

A new algorithm for labeling the base matrices of QC-GLDPC codes by the columns of a parity-check matrix of a constituent code was proposed in Sec. 3.2.4. It was demonstrated that an improved BP decoding performance of QC-GLDPC codes, at least in the error-floor region, can be achieved by the proposed optimized labeling of underlying binary QC-LDPC codes, while keeping the same low-complexity decoded constituent codes.

While the designed QC-GLDPC codes achieve a desirable decoding complexity and error rate performance, their main limitation lies in the lack of an efficient encoding procedure. This was a limiting factor in the analysis conducted as part of this dissertation, restricting simulations to $n \approx 2000$. As LDPC codes used in practical scenarios typically utilize large block lengths, this shortcoming is a severe issue limiting practical application of GLDPC codes and thus is deserving of further study as part of potential future work.

In Chapter 4, an approach to estimating the ML decoding error probability for both shaped and unshaped signaling with multiple variants of coded modulation was proposed. The developed approach was applied to both the ensemble of long linear codes and short linear codes for a theoretical analysis. It was shown that the derived estimates are rather tight and allow for an adequate comparison of communication systems with both shaped and unshaped coded modulation. The proposed methods of analysis of shaped modulation are used to optimize the BP decoding performance of NB LDPC codes with shaping in Chapter 5.

Multiple variants of coded modulation used for NB LDPC codes were studied: BICM, SICM, BPCM and ASCM. The two modulation techniques most commonly used with NB LDPC codes are BICM and SICM, however, the two techniques have limitations. In the case of BICM, reliabilities of PAM signals are recomputed into reliabilities of bits of signal point indices, thus specific information about the dependence between index bits is inevitably lost, resulting in loss of BP decoding performance. The use of SICM solves this problem, but it can be applied only if, for a modulation order $M = 2^p$ and NB codes over GF(2^m), p is a divisor of m. BPCM and ASCM mappings, proposed in [Boc+23c], are suitable for arbitrary pairs of parameters (M, m).

In the case that p is a divisor of m, SICM provides the best performance of BP decoding among all four mappings.

In Chapter 5, techniques for optimization of a shaped coded modulation systems using NB LDPC codes were proposed. For a given spectrum of the binary image of the code, methods using the estimated SEDS of the system and equivalent minimum distance were shown to be efficient tools for optimization. The SEDS of a code reduces the optimization procedure of a communication system using shaped coded modulation to a numerical problem suitable for optimization techniques, such as the proposed genetic algorithm.

Optimization of the shaping book is organized as a two-step procedure. First, optimization by a genetic algorithm is performed, using a simple score function. Next, the equivalent minimum distance is used for finer system optimization. Simulation results confirming the consistency of the approach were presented.

The shaping techniques used, developed and optimized as part of this work were designed specifically for NB LDPC codes. As such, their applicability in a wider context, including binary (possibly generalized) LDPC codes remains unstudied. As part of this work, preliminary analysis was conducted using binary QC-LDPC codes with the studied SAC methods, which yielded marginal gains in performance. Whether the designed methods could be applied to different classes of codes remains an open research question.

Other potential directions of future study include the combination of SBC and SAC techniques in one system, as well as considering additional types and higher orders of modulation. This work focused on irregular NB LDPC codes over small alphabets using BP decoding, in order to achieve a desirable computational complexity. As such, further research into codes over large alphabets, including ultra-sparse NB LDPC code constructions, could yield more impressive gains in decoding performance using the proposed optimized shaping techniques. Such codes could also be more naturally combined with higher orders of modulation, resulting in additional interesting results.

BIBLIOGRAPHY

[ADR11] Shadi Abu-Surra, Dariush Divsalar, and William E. Ryan. "Enumerators for protograph-based ensembles of LDPC and generalized LDPC codes". In: *IEEE Transactions on Information Theory* 57.2 (2011), pp. 858–886. DOI: 10.1109/TIT.2010.2094819.

[Agr+04] Erik Agrell, Johan Lassing, Erik G. Strom, and Tony Ottosson. "On the optimality of the binary reflected Gray code". In: *IEEE Transactions on Information Theory* 50.12 (2004), pp. 3170–3182. DOI: 10.1109/TIT.2004.838367.

[AKR19] Rajagopal Anantharaman, Karibasappa Kwadiki, and Vasundara Rao. "Hardware implementation analysis of Min-Sum decoders". In: *Advances in Electrical and Electronic Engineering* 17 (2019). DOI: 10.15598/aeee.v17i2.3042.

[AL04] Alexei Ashikhmin and Simon Litsyn. "Simple MAP decoding of first-order Reed-Muller and Hamming codes". In: *IEEE Transactions on Information Theory* 50.8 (2004), pp. 1812–1818. DOI: 10.1109/TIT.2004.831835.

[Ara+09] Murat Arabaci, Ivan B. Djordjevic, Ross Saunders, and Roberto M. Marcoccia. "High-rate nonbinary regular quasi-cyclic LDPC codes for optical communications". In: *Journal of Lightwave Technology* 27.23 (2009), pp. 5261–5267. DOI: 10.1109/JLT.2009.2029062.

[Ari09] Erdal Arikan. "Channel polarization: a method for constructing capacity-achieving codes for symmetric binary-input memoryless channels". In: *IEEE Transactions on Information Theory* 55.7 (2009), pp. 3051–3073. DOI: 10.1109/TIT.2009.2021379.

[Bah+74] Lalit R. Bahl, John Cocke, Frederick Jelinek, and Josef Raviv. "Optimal decoding of linear codes for minimizing symbol error rate". In: *IEEE Transactions on Information Theory* 20.2 (1974), pp. 284–287. DOI: 10.1109/TIT.1974.1055186.

[Bar98] Alexander Barg. "Complexity issues in coding theory". In: *Handbook of Coding Theory* 1 (1998), pp. 649–754. URL: https://cir.nii.ac.jp/crid/1572543025521778688.

[BB04] Amir Bennatan and David Burshtein. "On the application of LDPC codes to arbitrary discrete-memoryless channels". In: *IEEE Transactions on Information Theory* 50.3 (2004), pp. 417–438. DOI: 10.1109/TIT.2004.824917.

[BB06] Amir Bennatan and David Burshtein. "Design and analysis of nonbinary LDPC codes for arbitrary discrete-memoryless

channels". In: *IEEE Transactions on Information Theory* 52.2 (2006), pp. 549–583. DOI: 10.1109/TIT.2005.862080.

[BD03] Loïc Barnault and David Declercq. "Fast decoding algorithm for LDPC over GF(2^q)". In: *Proceedings of IEEE Information Theory Workshop (ITW)*. 2003, pp. 70–73. DOI: 10.1109/ITW.2003.1216697.

[Ber10] Claude Berrou. *Codes and Turbo Codes*. 1st. Berlin, Heidelberg: Springer-Verlag, 2010. ISBN: 2817800389.

[BGT93] Claude Berrou, Alain Glavieux, and Punya Thitimajshima. "Near Shannon limit error-correcting coding and decoding: Turbo-codes. 1". In: *IEEE International Conference on Communications (ICC)*. Vol. 2. 1993, pp. 1064–1070. DOI: 10.1109/ICC.1993.397441.

[BKJ16] Irina E. Bocharova, Boris D. Kudryashov, and Rolf Johannesson. "Searching for binary and nonbinary block and convolutional LDPC codes". In: *IEEE Transactions on Information Theory* 62.1 (2016), pp. 163–183. DOI: 10.1109/TIT.2015.2496213.

[BKM22] Irina E. Bocharova, Boris D. Kudryashov, and Sander Mikelsaar. "Irregular generalized LDPC codes in practical communication scenarios". In: *Proceedings of IEEE Information Theory Workshop (ITW)*. 2022, pp. 517–522. DOI: 10.1109/ITW54588.2022.9965833.

[BKM24] Irina E. Bocharova, Boris D. Kudryashov, and Sander Mikelsaar. "Analysis of coded shaped QAM signaling at short and moderate lengths". In: *Proceedings of IEEE International Symposium on Information Theory (ISIT)*. 2024, pp. 641–646. DOI: 10.1109/ISIT57864.2024.10619186.

[BKS23] Irina E. Bocharova, Boris D. Kudryashov, and Vitaly Skachek. "LDPC coded QAM signaling: mapping and shaping". In: *12th International Symposium on Topics in Coding (ISTC)*. 2023, pp. 1–6. DOI: 10.1109/ISTC57237.2023.10273506.

[Boc+04] Irina E. Bocharova, Marc Handlery, Rolf Johannesson, and Boris D. Kudryashov. "A BEAST for prowling in trees". In: *IEEE Transactions on Information Theory* 50.6 (2004), pp. 1295–1302. DOI: 10.1109/TIT.2004.828093.

[Boc+17] Irina E. Bocharova, Boris D. Kudryashov, Vitaly Skachek, and Yauhen Yakimenka. "Average spectra for ensembles of LDPC codes and applications". In: *Proceedings of IEEE International Symposium on Information Theory (ISIT)*. 2017, pp. 361–365. DOI: 10.1109/ISIT.2017.8006550.

[Boc+21] Irina E. Bocharova, Boris D. Kudryashov, Evgenii P. Ovsyannikov, Vitaly Skachek, and Tähvend Uustalu. "Optimization

of irregular NB QC-LDPC block codes over small alphabets". In: *Proceedings of IEEE Information Theory Workshop (ITW)*. 2021, pp. 1–5. DOI: 10.1109/ITW46852.2021.9457656.

[Boc+22] Irina E. Bocharova, Boris D. Kudryashov, Evgenii P. Ovsyannikov, Vitaly Skachek, and Tähvend Uustalu. "Design and analysis of NB QC-LDPC codes over small alphabets". In: *IEEE Transactions on Communications* 70.5 (2022), pp. 2964–2976. DOI: 10.1109/TCOMM.2022.3160176.

[Boc+23a] Irina E. Bocharova, Boris D. Kudryashov, Sander Mikelsaar, and Vitaly Skachek. "Bound on the ML decoding error probability for coded QAM signals with shaping". In: *Proceedings of IEEE International Symposium on Information Theory (ISIT)*. 2023, pp. 270–275. DOI: 10.1109/ISIT54713.2023.10206612.

[Boc+23b] Irina E. Bocharova, Boris D. Kudryashov, Sander Mikelsaar, and Vitaly Skachek. "Shaping for NB QC-LDPC coded QAM signals". In: *12th International Symposium on Topics in Coding (ISTC)*. 2023, pp. 1–5. DOI: 10.1109/ISTC57237.2023.10273467.

[Boc+23c] Irina E. Bocharova, Boris D. Kudryashov, Evgenii P. Ovsyannikov, and Vitaly Skachek. "Nonbinary LDPC coded QAM signals with optimized mapping: Bounds and simulation results". In: *IEEE Transactions on Information Theory* (2023). DOI: 10.1109/TIT.2023.3264489.

[Bou+14] Boubakar S. Bouazza, Stéphane Y. Le Goff, Ahmed Garadi, Clency Perrine, and Rodolphe Vauzelle. "A novel constellation shaping technique for bit-interleaved coded modulation". In: *Wireless Personal Communications* 74.2 (2014), pp. 519–528. DOI: 10.1007/s11277-013-1303-9.

[BPZ99] Joseph Boutros, Olivier Pothier, and Gilles Zemor. "Generalized low density (Tanner) codes". In: *IEEE International Conference on Communications (ICC)*. Vol. 1. 1999, pp. 441–445. DOI: 10.1109/ICC.1999.767979.

[Bri00] Stephan ten Brink. "Rate one-half code for approaching the Shannon limit by 0.1 dB". In: *Electronics Letters* 36 (2000), pp. 1293–1294. DOI: 10.1049/el:20000953.

[Bri01] Stephan ten Brink. "Convergence behavior of iteratively decoded parallel concatenated codes". In: *IEEE Transactions on Communications* 49.10 (2001), pp. 1727–1737. DOI: 10.1109/26.957394.

[BSS15] Georg Böcherer, Fabian Steiner, and Patrick Schulte. "Bandwidth efficient and rate-matched low-density parity-check coded

modulation". In: *IEEE Transactions on Communications* 63.12 (2015), pp. 4651–4665. DOI: 10.1109/TCOMM.2015.2494016.

[CDD12] Ben-Yue Chang, Dariush Divsalar, and Lara Dolecek. "Non-binary protograph-based LDPC codes for short block-lengths". In: *Proceedings of IEEE Information Theory Workshop (ITW)*. 2012, pp. 282–286. DOI: 10.1109/ITW.2012.6404676.

[CF02a] Jinghu Chen and Marc P. C. Fossorier. "Density evolution for two improved BP-Based decoding algorithms of LDPC codes". In: *IEEE Communications Letters* 6.5 (2002), pp. 208–210. DOI: 10.1109/4234.1001666.

[CF02b] Jinghu Chen and Marc P. C. Fossorier. "Near optimum universal belief propagation based decoding of low-density parity check codes". In: *IEEE Transactions on Communications* 50.3 (2002), pp. 406–414. DOI: 10.1109/26.990903.

[Chu+01] Sae-Young Chung, G. David Forney, Thomas J. Richardson, and Rüdiger L. Urbanke. "On the design of low-density parity-check codes within 0.0045 dB of the Shannon limit". In: *IEEE Communications Letters* 5.2 (2001), pp. 58–60. DOI: 10.1109/4234.905935.

[Chu00] Sae-Young Chung. "On the construction of some capacity-approaching coding schemes". PhD thesis. Massachusetts Institute of Technology, 2000. URL: http://hdl.handle.net/1721.1/8981.

[CO90] A. Robert Calderbank and Lawrence H. Ozarow. "Nonequiprobable signaling on the Gaussian channel". In: *IEEE Transactions on Information Theory* 36.4 (1990), pp. 726–740. DOI: 10.1109/18.53734.

[Cov73] Thomas M. Cover. "Enumerative source encoding". In: *IEEE Transactions on Information Theory* 19.1 (1973), pp. 73–77. DOI: 10.1109/TIT.1973.1054929.

[CRU01] Sae-Young Chung, Thomas J. Richardson, and Rüdiger L. Urbanke. "Analysis of sum-product decoding of low-density parity-check codes using a Gaussian approximation". In: *IEEE Transactions on Information Theory* 47.2 (2001), pp. 657–670. DOI: 10.1109/18.910580.

[CSN17] China Satellite Navigation Office CSNO. *Beidou navigation satellite system signal in space interface control document - open service signal B2a (version 1.0)*. 2017.

[CTB98] Giuseppe Caire, Giorgio Taricco, and Ezio Biglieri. "Bit-interleaved coded modulation". In: *IEEE Transactions on Information Theory* 44.3 (1998), pp. 927–946. DOI: 10.1109/18.669123.

[DCG04] David Declercq, Maxime Colas, and Guillaume Gelle. "Regular GF (2^q)-LDPC modulations for higher order QAM-AWGN channels". In: *Proceedings of International Symposium on Information Theory and Its Applications (ISITA)*. 2004, pp. 1–6. URL: https://perso.etis-lab.fr/declercq/PDF/ConferencePapers/Declercq_2004_ISITA.pdf.

[DF07] David Declercq and Marc P. C. Fossorier. "Decoding algorithms for nonbinary LDPC codes over GF(q)". In: *IEEE Transactions on Communications* 55.4 (2007), pp. 633–643. DOI: 10.1109/TCOMM.2007.894088.

[Di+02] Changyan Di, David Proietti, I. Emre Telatar, Thomas J. Richardson, and Rüdiger L. Urbanke. "Finite-length analysis of low-density parity-check codes on the binary erasure channel". In: *IEEE Transactions on Information Theory* 48.6 (2002), pp. 1570–1579. DOI: 10.1109/TIT.2002.1003839.

[Dio+15] Madiagne Diouf, David Declercq, Samuel Ouya, and Bane Vasic. "A PEG-like LDPC code design avoiding short trapping sets". In: *Proceedings of IEEE International Symposium on Information Theory (ISIT)*. 2015, pp. 1079–1083. DOI: 10.1109/ISIT.2015.7282621.

[Dio+16] Madiagne Diouf, David Declercq, Marc P. C. Fossorier, S. Ouya, and Bane V. Vasic. "Improved PEG construction of large girth QC-LDPC codes". In: *9th International Symposium on Topics in Coding (ISTC)*. 2016. DOI: 10.1109/ISTC.2016.7593094.

[DM98] Matthew C. Davey and David J. C. MacKay. "Low density parity check codes over GF (q)". In: *Proceedings of IEEE Information Theory Workshop (ITW)*. 1998, pp. 70–71. DOI: 10.1109/ITW.1998.706440.

[DMV05] Ivan B. Djordjevic, Olgica Milenkovic, and Bane Vasic. "Generalized low-density parity-check codes for optical communication systems". In: *Journal of Lightwave Technology* 23.5 (2005), pp. 1939–1946. DOI: 10.1109/JLT.2005.846892.

[Dol+14] Lara Dolecek, Dariush Divsalar, Yizeng Sun, and Behzad Amiri. "Non-binary protograph-based LDPC codes: Enumerators, analysis, and designs". In: *IEEE Transactions on Information Theory* 60.7 (2014), pp. 3913–3941. DOI: 10.1109/TIT.2014.2316215.

[DW14] Ivan B. Djordjevic and Ting Wang. "Multiple component codes based generalized LDPC codes for high-speed optical transport". In: *Optics Express* 22.14 (2014), pp. 16694–16705. DOI: 10.1364/OE.22.016694.

[EE09a] European Telecommunications Standards Institute (ETSI) and European Broadcasting Union (EBU). *ETSI EN 302 307*. 2009. URL: https://www.etsi.org/deliver/etsi_en/302300_302399/302307/01.02.01_60/en_302307v010201p.pdf.

[EE09b] Institute of Electrical and Electronics Engineers. *IEEE Standard for Information technology– Local and metropolitan area networks– Specific requirements– Part 11: Wireless LAN Medium Access Control (MAC)and Physical Layer (PHY) Specifications Amendment 5: Enhancements for Higher Throughput*. 2009. DOI: 10.1109/IEEESTD.2009.5307322.

[EE21] Institute of Electrical and Electronics Engineers. *IEEE Standard for Information Technology–Telecommunications and Information Exchange between Systems Local and Metropolitan Area Networks–Specific Requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 1: Enhancements for High-Efficiency WLAN*. 2021. DOI: 10.1109/IEEESTD.2021.9442429.

[Eli55] Peter Elias. "Coding for two noisy channels". In: *Proceedings of 3rd London Symposium on Information Theory*. Buttersworth's Scientific Publications, 1955, pp. 61–67. URL: https://web.mit.edu/6.441/www/reading/hd2.pdf.

[ETS18] European Telecommunications Standards Institute (ETSI). *LTE; Evolved Universal Terrestrial Radio Access (E-UTRA); Multiplexing and channel coding (3GPP TS 36.212 version 15.2.1 Release 15)*. 2018. URL: https://www.etsi.org/deliver/etsi_ts/136200_136299/136212/15.02.01_60/ts_136212v150201p.pdf.

[ETS21] European Telecommunications Standards Institute (ETSI). *5G; NR; Multiplexing and channel coding (3GPP TS 38.212 version 16.5.0 Release 16)*. 2021. URL: https://www.etsi.org/deliver/etsi_ts/138200_138299/138212/16.05.00_60/ts_138212v160500p.pdf.

[Feh+15] Tobias Fehenberger, Georg Böcherer, Alex Alvarado, and Norbert Hanik. "LDPC coded modulation with probabilistic shaping for optical fiber systems". In: *Optical Fiber Communication Conference and Exposition (OFC)*. 2015, pp. 1–3. DOI: 10.1364/OFC.2015.Th2A.23.

[Feh+19] Tobias Fehenberger, David S. Millar, Toshiaki Koike-Akino, Keisuke Kojima, and Kieran Parsons. "Multiset-partition distribution matching". In: *IEEE Transactions on Communications* 67.3 (2019), pp. 1885–1893. DOI: 10.1109/TCOMM.2018.2881091.

[Fel68] William Feller. *An Introduction to Probability Theory and Its Applications*. Vol. 1. Wiley, 1968. ISBN: 0471257087.

[Fen+15] Dan Feng, Qi Li, Baoming Bai, and Xiao Ma. "Gallager mapping based constellation shaping for LDPC-coded modulation systems". In: *IEEE International Workshop on High Mobility Wireless Communications (HMWC)*. 2015, pp. 116–120. DOI: 10.1109/HMWC.2015.7354347.

[FMI99] Marc P. C. Fossorier, Miodrag J. Mihaljevic, and Hideki Imai. "Reduced complexity iterative decoding of low-density parity check codes based on belief propagation". In: *IEEE Transactions on Communications* 47.5 (1999), pp. 673–680. DOI: 10.1109/26.768759.

[For+84] G. David Forney, Robert G. Gallager, Gordon R. Lang, Fred M. Longstaff, and Shahid U. Qureshi. "Efficient modulation for band-limited channels". In: *IEEE Journal on Selected Areas in Communications (JSAC)* 2.5 (1984), pp. 632–647. DOI: 10.1109/JSAC.1984.1146101.

[For92] G. David Forney. "Trellis shaping". In: *IEEE Transactions on Information Theory* 38.2 (1992), pp. 281–300. DOI: 10.1109/18.119687.

[Fos04] Marc P. C. Fossorier. "Quasicyclic low-density parity-check codes from circulant permutation matrices". In: *IEEE Transactions on Information Theory* 50.8 (2004), pp. 1788–1793. DOI: 10.1109/TIT.2004.831841.

[FW89] G. David Forney and Lee-Fang Wei. "Multidimensional constellations. I. Introduction, figures of merit, and generalized cross constellations". In: *IEEE Journal on Selected Areas in Communications (JSAC)* 7.6 (1989), pp. 877–892. DOI: 10.1109/49.29611.

[FZ99] Alberto J. Felstrom and Kamil S. Zigangirov. "Time-varying periodic convolutional codes with low-density parity-check matrix". In: *IEEE Transactions on Information Theory* 45.6 (1999), pp. 2181–2191. DOI: 10.1109/18.782171.

[Gal62] Robert G. Gallager. "Low-density parity-check codes". In: *IRE Transactions on Information Theory* 8.1 (1962), pp. 21–28. DOI: 10.1109/TIT.1962.1057683.

[Gal63] Robert G. Gallager. *Low-Density Parity-Check Codes*. The MIT Press, 1963. ISBN: 9780262256216. DOI: 10.7551/mitpress/4347.001.0001.

[Gal68] Robert G. Gallager. *Information Theory and Reliable Communication*. John Wiley & Sons, Inc., 1968. ISBN: 0471290483.

[Gei+23] Marvin Geiselhart, Felix Krieg, Jannis Clausius, Daniel Tandler, and Stephan ten Brink. "6G: A welcome chance to unify channel coding?" In: *IEEE BITS the Information Theory Magazine* 3.1 (2023), pp. 67–80. DOI: 10.1109/MBITS.2023.3322974.

[Gil52] Edgar N. Gilbert. "A comparison of signalling alphabets". In: *The Bell System Technical Journal* 31.3 (1952), pp. 504–522. DOI: 10.1002/j.1538-7305.1952.tb01393.x.

[GMC08] Albert Guillén i Fàbregas, Alfonso Martinez, and Giuseppe Caire. *Bit-interleaved coded modulation*. Vol. 5. Foundations and trends in communications and information theory 1–2. Now Publishers, 2008, pp. 1–153. DOI: 10.1561/0100000019.

[Gra53] Frank Gray. "Pulse Code Communication". U.S. pat. US2632058A. Mar. 17, 1953. URL: https://patents.google.com/patent/US2632058A/en.

[GSD10] Matteo Gorgoglione, Valentin Savin, and David Declercq. "Optimized puncturing distributions for irregular non-binary LDPC codes". In: *Proceedings of International Symposium on Information Theory and Its Applications (ISITA)*. 2010, pp. 400–405. DOI: 10.1109/ISITA.2010.5649264.

[Gül+18] Yunus Can Gültekin, Wim J. van Houtum, Frans M. J. Willems, Luuk Spreeuwers, and Jasper Goseling. "On constellation shaping for short block lengths". In: *Symposium on Information Theory and Signal Processing in the Benelux (SITB)*. University of Twente. 2018, pp. 86–96. URL: https://research.tue.nl/en/publications/on-constellation-shaping-for-short-block-lengths.

[Gül+20a] Yunus Can Gültekin, Tobias Fehenberger, Alex Alvarado, and Frans M. J. Willems. "Probabilistic shaping for finite blocklengths: Distribution matching and sphere shaping". In: *Entropy* 22.5 (2020), p. 581. DOI: 10.3390/e22050581.

[Gül+20b] Yunus Can Gültekin, Wim J. van Houtum, Arie G. C. Koppelaar, and Frans M. J. Willems. "Enumerative sphere shaping for wireless communications with short packets". In: *IEEE Transactions on Wireless Communications* 19.2 (2020), pp. 1098–1112. DOI: 10.1109/TWC.2019.2951139.

[Ham50] Richard W. Hamming. "Error detecting and error correcting codes". In: *The Bell System Technical Journal* 29.2 (1950), pp. 147–160. DOI: 10.1002/j.1538-7305.1950.tb00463.x.

[HE04] Xiao-Yu Hu and Evangelos Eleftheriou. "Binary representation of cycle Tanner-graph GF(2^b) codes". In: *IEEE International Conference on Communications (ICC)*. Vol. 1. 2004, pp. 528–532. DOI: 10.1109/ICC.2004.1312545.

[HEA01] Xiao-Yu Hu, Evangelos Eleftheriou, and Dieter-Michael Arnold. "Progressive edge-growth Tanner graphs". In: *IEEE Global Communications Conference (GLOBECOM)* 2 (2001), pp. 995–1001. DOI: 10.1109/GLOCOM.2001.965567.

[HEA05] Xiao-Yu Hu, Evangelos Eleftheriou, and Dieter-Michael Arnold. "Regular and irregular progressive edge-growth Tanner graphs". In: *IEEE Transactions on Information Theory* 51 (2005), pp. 386–398. DOI: 10.1109/TIT.2004.839541.

[Hoc04] Dale E. Hocevar. "A reduced complexity decoder architecture via layered decoding of LDPC codes". In: *IEEE Workshop on Signal Processing Systems (SIPS)*. 2004, pp. 107–112. DOI: 10.1109/SIPS.2004.1363033.

[HZW08] Jie Huang, Shengli Zhou, and Peter Willett. "Nonbinary LDPC coding for multicarrier underwater acoustic communication". In: *IEEE Journal on Selected Areas in Communications (JSAC)* 26.9 (2008), pp. 1684–1696. DOI: 10.1109/JSAC.2008.081208.

[JK05] Sudhanshu John and Hyuck M. Kwon. "Approximate cycle extrinsic message degree regular quasi circulant LDPC codes". In: *IEEE Military Communications Conference (MILCOM)*. 2005, pp. 2877–2881. DOI: 10.1109/MILCOM.2005.1606100.

[Kai+07] Sunil Kaimalettu, Andrew Thangaraj, Matthieu Bloch, and Steven W. McLaughlin. "Constellation shaping using LDPC codes". In: *Proceedings of IEEE International Symposium on Information Theory (ISIT)*. 2007, pp. 2366–2370. DOI: 10.1109/ISIT.2007.4557573.

[KLF01] Yu Kou, Shu Lin, and Marc P. C. Fossorier. "Low-density parity-check codes based on finite geometries: a rediscovery and new results". In: *IEEE Transactions on Information Theory* 47.7 (2001), pp. 2711–2736. DOI: 10.1109/18.959255.

[KP93] Frank R. Kschischang and Subbarayan Pasupathy. "Optimal nonuniform signaling for Gaussian channels". In: *IEEE Transactions on Information Theory* 39 (1993), pp. 913–929. DOI: 10.1109/18.256499.

[Kyu+16] Kim Kyung-Joong et al. "Low-density parity-check codes for ATSC 3.0". In: *IEEE Transactions on Broadcasting* 62 (2016), pp. 189–196. DOI: 10.1109/TBC.2016.2515538.

[LC83] Shu Lin and Daniel J. Costello. *Error Control Coding: Fundamentals and Applications*. Computer applications in electrical engineering series. Prentice-Hall, 1983. ISBN: 9780132837965. URL: https://books.google.ee/books?id=autQAAAAMAAJ.

[Le +07] Stéphane Y. Le Goff, Boon Kien Khoo, Charalampos C. Tsimenidis, and Bayan S. Sharif. "Constellation shaping for bandwidth-

efficient turbo-coded modulation with iterative receiver". In: *IEEE Transactions on Wireless Communications* 6.6 (2007), pp. 2223–2233. DOI: 10.1109/TWC.2007.05780.

[LFK09] Ge Li, Ivan J. Fair, and Witold A. Krzymien. "Density evolution for nonbinary LDPC codes under Gaussian approximation". In: *IEEE Transactions on Information Theory* 55.3 (2009), pp. 997–1015. DOI: 10.1109/TIT.2008.2011435.

[LFT94] Rajiv Laroia, Nariman Farvardin, and Steven A. Tretter. "On optimal shaping of multidimensional constellations". In: *IEEE Transactions on Information Theory* 40.4 (1994), pp. 1044–1056. DOI: 10.1109/18.335969.

[Lin+08] Yi-Kai Lin, Chih-Lung Chen, Yen-Chin Liao, and Hsie-Chia Chang. "Structured LDPC codes with low error floor based on PEG Tanner graphs". In: *IEEE International Symposium on Circuits and Systems (ISCAS)*. 2008, pp. 1846–1849. DOI: 10.1109/ISCAS.2008.4541800.

[LK04] Zongwang Li and B. Kumar. "A class of good quasi-cyclic low-density parity check codes based on progressive edge growth graph". In: *Asilomar Conference Signals, Systems, and Computers*. Vol. 2. 2004, pp. 1990–1994. DOI: 10.1109/ACSSC.2004.1399513.

[LM05] Stefan Landner and Olgica Milenkovic. "Algorithmic and combinatorial analysis of trapping sets in structured LDPC codes". In: *International Conference on Wireless Networks, Communications and Mobile Computing*. Vol. 1. 2005, pp. 630–635. DOI: 10.1109/WIRLES.2005.1549481.

[LOM18] Yanfang Liu, Pablo M. Olmos, and David G. M. Mitchell. "On generalized LDPC codes for 5G ultra reliable communication". In: *Proceedings of IEEE Information Theory Workshop (ITW)*. 2018, pp. 1–5. DOI: 10.1109/ITW.2018.8613515.

[LR05] Gianluigi Liva and William E. Ryan. "Short low-error-floor Tanner codes with Hamming nodes". In: *IEEE Military Communications Conference (MILCOM)*. 2005, pp. 208–213. DOI: 10.1109/MILCOM.2005.1605687.

[LR80] Robert Lugannani and Stephen Rice. "Saddle point approximation for the distribution of the sum of independent random variables". In: *Advances in Applied Probability* 12.2 (1980), pp. 475–490. DOI: 10.2307/1426607.

[LRC08] Gianluigi Liva, William E. Ryan, and Marco Chiani. "Quasi-cyclic generalized LDPC codes with low error floors". In: *IEEE Transactions on Communications* 56.1 (2008), pp. 49–57. DOI: 10.1109/TCOMM.2008.050600.

[Lub+01] Michael G. Luby, Michael Mitzenmacher, M. Amin Shokrollahi, and Daniel A. Spielman. "Improved low-density parity-check codes using irregular graphs". In: *IEEE Transactions on Information Theory* 47.2 (2001), pp. 585–598. DOI: 10.1109/18.910576.

[Lub+97] Michael G. Luby, Michael Mitzenmacher, M. Amin Shokrollahi, Daniel A. Spielman, and Volker Stemann. "Practical loss-resilient codes". In: *Proceedings of ACM Symposium on Theory of Computing (STOC)*. 1997, pp. 150–159. DOI: 10.1145/258533.258573.

[LV96] Alec Lafourcade and Alexander Vardy. "Optimal sectionalization of a trellis". In: *IEEE Transactions on Information Theory* 42.3 (1996), pp. 689–703. DOI: 10.1109/18.490504.

[LZ99] Michael Lentmaier and Kamil Sh. Zigangirov. "On generalized low-density parity-check codes based on Hamming component codes". In: *IEEE Communications Letters* 3.8 (1999), pp. 248–250. DOI: 10.1109/4234.781010.

[MMM04] David J. C. MacKay, Graeme Mitchison, and Paul L. McFadden. "Sparse-graph codes for quantum error correction". In: *IEEE Transactions on Information Theory* 50.10 (2004), pp. 2315–2330. DOI: 10.1109/tit.2004.834737.

[MN96] David J. C. Mackay and Radford M. Neal. "Near Shannon limit performance of low density parity check codes". In: *Electronics Letters*. Vol. 32. 1996, pp. 1645–1646. DOI: 10.1049/el:19961141.

[MS77] Florence Jessie MacWilliams and Neil James Alexander Sloane. *The theory of error-correcting codes*. Vol. 16. Elsevier, 1977. ISBN: 978-0-444-85010-2.

[MTB15] Latifa Mostari, Abdelmalik Taleb-Ahmed, and Abdennacer Bounoua. "Simplified soft output demapper for non-binary LDPC codes". In: *Optik* 126.24 (2015), pp. 5074–5076. DOI: 10.1016/j.ijleo.2015.09.210.

[MW86] H. Ma and Jack K. Wolf. "On tail biting convolutional codes". In: *IEEE Transactions on Communications* 34.2 (1986), pp. 104–111. DOI: 10.1109/TCOM.1986.1096498.

[PCF08] Enrico Paolini, Marco Chiani, and Marc P. C. Fossorier. "On the growth rate of irregular GLDPC codes weight distribution". In: *IEEE International Symposium on Spread Spectrum Techniques and Applications*. 2008, pp. 790–794. DOI: 10.1109/ISSSTA.2008.154.

[PFD06] Charly Poulliat, Marc P. C. Fossorier, and David Declercq. "Using binary images of non binary LDPC codes to improve

overall performance". In: *4th International Symposium on Turbo Codes & Related Topics; 6th International ITG-Conference on Source and Channel Coding*. 2006, pp. 1–6. URL: https://ieeexplore.ieee.org/document/5755957.

[PFD08] Charly Poulliat, Marc P. C. Fossorier, and David Declercq. "Design of regular (2, d_c)-LDPC codes over GF(q) using their binary images". In: *IEEE Transactions on Communications* 56.10 (2008), pp. 1626–1635. DOI: 10.1109/TCOMM.2008.060527.

[Pfl+09] Stephan Pfletschinger, Alain Mourad, Eduardo Lopez, David Declercq, and Giacomo Bacci. "Performance evaluation of non-binary LDPC codes on wireless channels". In: *Proceedings of ICT Mobile Summit*. Santander. 2009, pp. 1–8. DOI: 20.500.12860/16013.

[PK16] Chunpo Pan and Frank R. Kschischang. "Probabilistic 16-QAM shaping in WDM systems". In: *Journal of Lightwave Technology* 34.18 (2016), pp. 4285–4292. DOI: 10.1109/JLT.2016.2594296.

[Pol94] Gregory Poltyrev. "Bounds on the decoding error probability of binary linear codes via their spectra". In: *IEEE Transactions on Information Theory* 40.4 (1994), pp. 1284–1292. DOI: 10.1109/18.335935.

[PPV10] Yury Polyanskiy, H. Vincent Poor, and Sergio Verdú. "Channel coding rate in the finite blocklength regime". In: *IEEE Transactions on Information Theory* 56.5 (2010), pp. 2307–2359. DOI: 10.1109/TIT.2010.2043769.

[Ran+14] Sudarsan V. S. Ranganathan, Dariush Divsalar, Kasra Vakilinia, and Richard D. Wesel. "Design of high-rate irregular non-binary LDPC codes using algorithmic stopping-set cancellation". In: *Proceedings of IEEE International Symposium on Information Theory (ISIT)*. 2014, pp. 711–715. DOI: 10.1109/ISIT.2014.6874925.

[RC89] Marc Rouanne and Daniel J. Costello. "An algorithm for computing the distance spectrum of trellis codes". In: *IEEE Journal on Selected Areas in Communications (JSAC)* 7.6 (1989), pp. 929–940. DOI: 10.1109/49.29615.

[Rez+20] Gada Rezgui, Asma Maaloui, Iryna Andryanova, Charly Poulliat, and Cyril Méasson. "NB-LDPC Codes with high rates achieving low BER over the AWGN channel with QAM signaling". In: *Proceedings of International Symposium on Information Theory and Its Applications (ISITA)*. 2020, pp. 235–239. URL: https://ieeexplore.ieee.org/document/9366200.

[RG04] Dan Raphaeli and Assaf Gurevitz. "Constellation shaping for pragmatic turbo-coded modulation with high spectral efficiency".

In: *IEEE Transactions on Communications* 52.3 (2004), pp. 341–345. DOI: 10.1109/TCOMM.2004.823564.

[Ric03] Thomas J. Richardson. "Error floors of LDPC codes". In: *Annual Allerton Conference on Communication, Control, and Computing* (2003), 1426–1435. URL: https://web.stanford.edu/class/ee388/papers/ErrorFloors.pdf.

[RSU01] Thomas J. Richardson, M. Amin Shokrollahi, and Rüdiger L. Urbanke. "Design of capacity-approaching irregular low-density parity-check codes". In: *IEEE Transactions on Information Theory* 47.2 (2001), pp. 619–637. DOI: 10.1109/18.910578.

[RU01] Thomas J. Richardson and Rüdiger L. Urbanke. "The capacity of low-density parity-check codes under message-passing decoding". In: *IEEE Transactions on Information Theory* 47.2 (2001), pp. 599–618. DOI: 10.1109/18.910577.

[SB16] Patrick Schulte and Georg Böcherer. "Constant composition distribution matching". In: *IEEE Transactions on Information Theory* 62.1 (2016), pp. 430–434. DOI: 10.1109/TIT.2015.2499181.

[See92] Gilg U.H. Seeber. "Saddlepoint approximations for generalized linear models: A gentle introduction". In: *Advances in GLIM and Statistical Modelling*. Springer. 1992, pp. 195–200. DOI: 10.1007/978-1-4612-2952-0_30.

[SH05] Frank Schreckenbach and Patrick Henkel. "Signal shaping using non-unique symbol mappings". In: *Proceedings of Annual Allerton Conference on Communication, Control, and Computing*. 2005.

[SH15] Feijin Shi and Shuangshuang Han. "New PEG algorithm with low error floor for construction of irregular LDPC codes". In: *International Conference on Computer and Communications (ICCC)*. 2015, pp. 290–294. DOI: 10.1109/CompComm.2015.7387584.

[Sha48] Claude E. Shannon. "A mathematical theory of communication". In: *The Bell System Technical Journal* 27.3 (1948), pp. 379–423. DOI: 10.1002/j.1538-7305.1948.tb01338.x.

[Sin64] Richard C. Singleton. "Maximum distance q-nary codes". In: *IEEE Transactions on Information Theory* 10.2 (1964), pp. 116–118. DOI: 10.1109/TIT.1964.1053661.

[SLB17] Fabian Steiner, Gianluigi Liva, and Georg Böcherer. "Ultra-sparse non-binary LDPC codes for probabilistic amplitude shaping". In: *IEEE Global Communications Conference (GLOBECOM)*. 2017, pp. 1–5. DOI: 10.1109/GLOCOM.2017.8254155.

[SLF03] Rose Shao, Shu Lin, and Marc P. C. Fossorier. "Two decoding algorithms for tailbiting codes". In: *IEEE Transactions on Communications* 51 (2003), pp. 1658–1665. DOI: 10.1109/TCOMM.2003.818084.

[SLK20] Patrick Schulte, Wafa Labidi, and Gerhard Kramer. "Joint decoding of distribution matching and error control codes". In: *International Zurich Seminar on Information and Communication (IZS)*. 2020, pp. 53–57. DOI: 10.3929/ethz-b-000402706.

[Spa15] Consultative Committee for Space Data Systems. *Short Block Length LDPC Codes for TC Synchronization and Channel Coding. CCSDS, 288 231.1-O-1 Orange Book.* 2015. URL: https://ccsds.org/Pubs/231x1o1s.pdf.

[SS06] Igal Sason and Shlomo Shamai. *Performance Analysis of Linear Codes under Maximum-Likelihood Decoding: A Tutorial*. Now Publishers Inc, 2006. DOI: 10.1561/0100000009.

[Sv79] Gustave Solomon and Henk C. A. van Tilborg. "A connection between block and convolutional codes". In: *SIAM Journal on Applied Mathematics* 37 (1979), pp. 358–369. DOI: 10.1137/0137027.

[Sv93] Feng-Wen Sun and Henk C. A. van Tilborg. "Approaching capacity by equiprobable signaling on the Gaussian channel". English. In: *IEEE Transactions on Information Theory* 39.5 (1993), pp. 1714–1716. DOI: 10.1109/18.259663.

[Tan+05] Heng Tang, Jun Xu, S. Lin, and Khaled A. S. Abdel-Ghaffar. "Codes on finite geometries". In: *IEEE Transactions on Information Theory* 51.2 (2005), pp. 572–596. DOI: 10.1109/TIT.2004.840867.

[Tan81] R. Michael Tanner. "A recursive approach to low complexity codes". In: *IEEE Transactions on Information Theory* 27.5 (1981), pp. 533–547. DOI: 10.1109/TIT.1981.1056404.

[TB21] Alireza Tasdighi and Emmanuel Boutillon. "Integer ring sieve for constructing compact QC-LDPC codes with girths 8, 10, and 12". In: *IEEE Transactions on Information Theory* 68.1 (2021), pp. 35–46. DOI: 10.1109/TIT.2021.3116655.

[Tia+03] Tao Tian, Christopher R. Jones, John D. Villasenor, and Richard D. Wesel. "Construction of irregular LDPC codes with low error floors". In: *IEEE International Conference on Communications (ICC)*. Vol. 5. 2003, pp. 3125–3129. DOI: 10.1109/ICC.2003.1203996.

[Tia+04] Tao Tian, Christopher R. Jones, John D. Villasenor, and Richard D. Wesel. "Selective avoidance of cycles in irregular LDPC code

construction". In: *IEEE Transactions on Communications* 52.8 (2004), pp. 1242–1247. DOI: 10.1109/TCOMM.2004.833048.

[TW67] Richard L. Townsend and Edward J. Weldon. "Self-orthogonal quasi-cyclic codes". In: *IEEE Transactions on Information Theory* 13 (1967), pp. 183–195. DOI: 10.1109/TIT.1967.1053974.

[Ung02] Gottfried Ungerboeck. "Huffman shaping". In: *Codes, Graphs, and Systems*. Springer, 2002, pp. 299–313. DOI: 10.1007/978-1-4615-0895-3_17.

[Ung82] Gottfried Ungerboeck. "Channel coding with multilevel/phase signals". In: *IEEE Transactions on Information Theory* 28.1 (1982), pp. 55–67. DOI: 10.1109/TIT.1982.1056454.

[Var57] Rom R. Varshamov. "The evaluation of signals in codes with correction of errors". Russian. In: *Doklady Akademii Nauk SSSR* 117 (1957), pp. 739–741. ISSN: 0002-3264.

[Vit+89] Andrew J. Viterbi, Jack K. Wolf, Ephraim Zehavi, and Roberto Padovani. "A pragmatic approach to trellis-coded modulation". In: *IEEE Communications Magazine* 27.7 (1989), pp. 11–19. DOI: 10.1109/35.31452.

[Vit67] Andrew J. Viterbi. "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm". In: *IEEE Transactions on Information Theory* 13.2 (1967), pp. 260–269. DOI: 10.1109/TIT.1967.1054010.

[VM04] Bane V. Vasic and Olgica Milenkovic. "Combinatorial constructions of low-density parity-check codes for iterative decoding". In: *IEEE Transactions on Information Theory* 50.6 (2004), pp. 1156–1176. DOI: 10.1109/TIT.2004.828066.

[VS08] Dejan Vukobratovic and Vojin Senk. "Generalized ACE constrained progressive edge-growth LDPC code design". In: *IEEE Communications Letters* 12.1 (2008), pp. 32–34. DOI: 10.1109/LCOMM.2008.071457.

[VX12] Matthew C. Valenti and Xingyu Xiang. "Constellation shaping for bit-interleaved LDPC coded APSK". In: *IEEE Transactions on Communications* 60.10 (2012), pp. 2960–2970. DOI: 10.1109/TCOMM.2012.070912.110533.

[Wib96] Niclas Wiberg. "Codes and decoding on general graphs". PhD thesis. Linköping University, 1996. URL: https://www.essrl.wustl.edu/~jao/itrg/wiberg.pdf.

[Wol78] Jack K. Wolf. "Efficient maximum likelihood decoding of linear block codes using a trellis". In: *IEEE Transactions on Information Theory* 24.1 (1978), pp. 76–80. DOI: 10.1109/TIT.1978.1055821.

[WSM04] Henk Wymeersch, Heidi Steendam, and Marc E. Moeneclaey. "Log-domain decoding of LDPC codes over GF(q)". In: *IEEE International Conference on Communications (ICC)*. Vol. 2. 2004, pp. 772–776. DOI: 10.1109/ICC.2004.1312606.

[XB04] Hua Xiao and Amir H. Banihashemi. "Improved progressive-edge-growth (PEG) construction of irregular LDPC codes". In: *IEEE Communications Letters* 8.12 (2004), pp. 715–717. DOI: 10.1109/LCOMM.2004.839612.

[Yan+14] Metodi Yankov, Søren Forchhammer, Knud J. Larsen, and Lars P. B. Christensen. "Rate-adaptive constellation shaping for near-capacity achieving turbo coded BICM". In: *IEEE International Conference on Communications (ICC)*. 2014, pp. 2112–2117. DOI: 10.1109/ICC.2014.6883635.

[YPW07] Guosen Yue, Li Ping, and Xiaodong Wang. "Generalized low-density parity-check codes based on Hadamard constraints". In: *IEEE Transactions on Information Theory* 53.3 (2007), pp. 1058–1079. DOI: 10.1109/TIT.2006.890694.

[ZC10] Xinmiao Zhang and Fang Cai. "Partial-parallel decoder architecture for quasi-cyclic non-binary LDPC codes". In: *IEEE International Conference on Acoustics, Speech and Signal Processing*. 2010, pp. 1506–1509. DOI: 10.1109/ICASSP.2010.5495502.

[Zho+16] Lin Zhou, Weicheng Huang, Shengliang Peng, Yan Chen, and Yucheng He. "An improved design of Gallager mapping for LDPC-coded BICM-ID system". In: *Electronics* 20.1 (2016), pp. 16–21. DOI: 10.7251/ELS1620016Z.

[ZLS21] Peng W. Zhang, Francis C. M. Lau, and Chiu-Wing Sham. "Protograph-based LDPC Hadamard codes". In: *IEEE Transactions on Communications* 69.8 (2021), pp. 4998–5013. DOI: 10.1109/TCOMM.2021.3077939.

[ZP01] T. Zhang and Keshab K. Parhi. "High-performance, low-complexity decoding of generalized low-density parity-check codes". In: *IEEE Global Communications Conference (GLOBECOM)*. Vol. 1. 2001, pp. 181–185. DOI: 10.1109/GLOCOM.2001.965103.

ACKNOWLEDGEMENTS

I would like to express my sincerest gratitude to my supervisors – Irina Bocharova, Boris Kudryashov and Vitaly Skachek. I feel incredibly lucky to have been a part of their team during my PhD studies. Regardless of how busy their schedules were, they always managed to find the time to help me throughout this journey. Their relentless support, encouragement, and guidance have been invaluable throughout all aspects of this work. The depth of their expertise, the endless stream of new ideas and interests, and the tenacity of their work ethic, while almost frightening, will always be things I look up to and strive towards. I will always fondly remember our long talks and be grateful for everything you have done to help me reach this milestone – thank you!

I am also grateful to the reviewers and opponents, Prof. Emmanuel Boutillon, Prof. Michael Lentmaier, and Dr Maiara Francine Bollauf, for their attention to detail, insightful feedback and comments, which undoubtedly helped to improve this dissertation.

I would like to thank other members, past and present, of the coding theory group – Henk, Junming, and Ago. The time spent together at conferences and workshops, as well as here in Tartu, has added a lot to this journey, both professionally and personally. Always looking forward to Henk's visits, I am grateful for having shared in your wisdom and truly appreciate all your advice and encouragement over the years. I also want to thank Sedat for the unexpected neighbourly visits, words of motivation, and sense of humour. I am also thankful to have had the opportunity to meet and share an office with some amazing people – Baran, Bora, Daniel, Erki, Handy, Karan, Kübra, and Modar. Having the chance to get to know all of you, the diversity of your personalities and backgrounds, has made my time studying here so much more interesting and fun. Your friendship and support has meant a lot to me.

I am thankful to the staff at the Institute of Computer Science for the academic environment, interesting courses, and support throughout my studies. I am also grateful for the financial support I have received from the Estonian Research Council grants PRG49 and PRG2531, University of Tartu ASTRA Project PER ASPERA ICT Doctoral School, the Kristjan Jaak national scholarship programme and travel grants from the IEEE Information Theory Society.

Last but definitely not least, I am grateful to my friends and family, but in particular to my mother, for being unwavering in their support and encouragement. None of this would have been possible without everything you have done for me.

SISUKOKKUVÕTE

Iteratiivselt dekodeeritavate koodide analüüs ja optimeerimine

Tagamaks vigadeta kommunikatsiooni seadmete vahel on tarvis usaldusväärnet digitaalset andmesidet. Praktiliste sidesüsteemide disainimine nõuab tipptasemel veaparanduskoodide ja füüsiliste andmesignaalide edastamise meetodite kombineerimist ja ühist optimeerimist. Oluline väljakutse selliste süsteemide väljatöötamisel on kasutatavate meetodite piiratud arvutuslik keerukus, eriti mobiilseadmete puhul, mille puhul lisanduv keerukus tõlgendub otse suuremasse aku kasutusse. Kuigi kodeerimisteooria ja digitaalses kommunikatsioonis kasutatavad koodid on juba pikalt väljakujunenud uurimisvaldkonnad, on digitaalsete sidesüsteemide areng olnud viimaste aastate jooksul siiski kiire, näiteks 5G telekommunikatsioonivõrkude laialdasem kasutuselevõtt eelmise aastakümne lõpul. Käesoleva doktoritöö peamiseks sihiks on nende tehnoloogiate uurimine ja arendamine tulevaste sidestandardite tarbeks. Suures pildis on käesolev töö jaotatud kaheks: koodid ja modulatsioon.

Esiteks, et saavutada robustset veaparandust mürarohketes kanalites toimuva andmeside jaoks, keskendub doktoritöö mitmetele madala tihedusega paarsuskontrolli (ingl. k. LDPC) koodide klassidele. Antud koodiklass on leidnud laialdast rakendust kaasaegsetes telekommunikatsioonistandardites, sealhulgas 5G-s, tänu oma võimekusele pakkuda suurepärast veaparandust, samal ajal säilitades madalat arvutuslikku keerukust. Madal keerukus tuleneb antud koodide iteratiivsest graafipõhisest dekodeerimisest ning spetsiaalselt struktureeritud maatriksist, mis võimaldavad kiiret kodeerimisprotsessi.

Selleks, et võrreldes kasutusel olevate standarditega saavutada kõrgemat veaparandusjõudlust, on antud lõputöös keskendutud üldistatud (GLDPC) ja mittebinaarsete (NB LDPC) koodi variantide disainimisele. Lisaks on mõlema koodiklassi variandi teoreetiliseks analüüsiks lõputöös uuritud vastavate veaparandusvõimekuse hindamiseks mõeldud meetodeid, mis põhinevad lõpliku pikkusega juhusliku kodeerimise veatõenäosuse piirmääradel. Koodide disainimiseks on väljatöötatud graafipõhised lähenemised, mis optimeerivad koodide baasmaatriksite graafilisi struktuure, suurendades nende jõudlust. Kuna koodidisain on suure keerukusega protsess, on antud töös rõhk arvutuslikult tõhusate algoritmide loomisel. Mõlema koodiklassi variandi puhul keskendutakse spetsiifiliselt madala keerukusega disainidele, rakendades üldistatud koodide puhul vaid lihtsaid komponentkoode ning mittebinaarsete koodide loomisel kasutades vaid madala suurusega Galois' korpuseid. Antud piirangud raskendavad koodidisaini protsessi, kuid on vajalikud tagamaks loodud koodide praktilist väärtust ning eristavad käesolevat lõputööd suurest hulgast eelnevalt loodud disainidest.

Teiseks, et läheneda mürarohketes kanalites teoreetiliselt saavutatava

usaldusväärse kommunikatsiooni piiridele, on vaja ühendada kodeeritud modulatsioon ja signaalikujundamine (*shaping*), ning nendega kommunikatsioonisüsteemis kombineeritud veaparanduskoodide ühine optimeerimine. On teada, et signaalikujundamise abil on teoreetiliselt võimalik saavutada 1.53dB suurune võit kommunikatsioonisüsteemi üsaldusväärseks toiminguks vajalikus signaali- ja müratugevuse suhtarvus (SNR).

Keskendudes peamiselt mittebinaarsetele koodidele, on teada, et koodisümbolite mitteoptimaalne vastavusse sidumine füüsiliste kanalisignaalidega põhjustab sidesüsteemides suure hulga dekodeerimisvigu, mis toob omakorda kaasa märkimisväärse languse koodi veaparandusjõudluses ning seega sidesüsteemi usaldusväärsuses. Analüüsides koodi Hamming- ja Eukleidese kauguste spektreid ning nendevahelisi seoseid, on lõputöö raames loodud lähenemise abil võimalik erinevate kujundatud (*shaped*) kodeeritud modulatsiooni kasutavate sidesüsteemide puhul dekodeerimisel tekkivate vigade tõenäosust hinnata. Lisaks piirmäärade leidmisele kasutatakse antud lähenemisest tuletatud algoritme mitmet tüüpi kujundatud kodeeritud modulatsiooniskeemi optimeerimiseks. Selle tarbeks on lõputöö raames loodud kaheetapiline protsess, liites geneetilise algoritmi ning kujundatud sidesüsteemi Eukleidilise ruumi hinnangulisel miinimumdistantsil põhineva optimeerimisega. Optimeeritud skeemide abil on võimalik saavutada teoreetilistele piirmääradele lähenev võit kommunikatsioonisüsteemide vajatavas signaali-müratugevuse suhtarvus.

Lõputöö käigus väljapakutud piirmäärad ning disainitud koodide veaparandusvõimekuse simulatsioonitulemused on visualiseeritud ja võrreldud olemasolevate standardite ning muude valdkonnas populaarsete koodide ja modulatsiooniskeemidega. Ilma kujundatud modulatsioonita simuleeritud veaparanduse suhtarvudest võib järeldada, et antud töö käigus loodud koodid pakuvad suuremat jõudlust võrreldes Wi-Fi ja 5G standardites kasutusel olevate LDPC koodidega. Vahe jõudluses kasvab veel enam märgatavalt juhul kui müratase tõuseb. Võrreldes sidesüsteemi koos ja ilma töös loodud optimeeritud kujundatud modulatsiooniga, on näha teoreetilisele piirmäärale vähenevat veaparandusjõudluse tõusu.

CURRICULUM VITAE

Personal data

Name:	Sander Mikelsaar
Date of birth:	09.10.1992
Nationality:	Estonian
E-mail:	sandermikelsaar@gmail.com

Education

2021–2025	University of Tartu, PhD in Computer Science
2017–2019	Tallinn University of Technology and University of Tartu, MSc in Cybersecurity
2012–2016	University of Tartu, BSc in Computer Science
2000–2012	Miina Härma Gymnasium

Employment

2021–	Junior Research Fellow in Coding Theory, University of Tartu
2020–2021	Scientific Programmer, University of Tartu

Scientific work

Main fields of interest:

- Coding theory
- Cryptography

ELULOOKIRJELDUS

Isikuandmed

Nimi:	Sander Mikelsaar
Sünniaeg:	09.10.1992
Kodakondsus:	Eesti
E-post:	sandermikelsaar@gmail.com

Haridus

2021–2025	Tartu Ülikool, PhD, Informaatika
2017–2019	Tallinna Tehnikaülikool ja Tartu Ülikool, MSc, Küberkaitse
2012–2016	Tartu Ülikool, BSc, Informaatika
2000–2012	Miina Härma Gümnaasium

Teenistuskäik

2021–	Kodeerimisteooria nooremteadur, Tartu Ülikool
2020–2021	Teaduslik programmeerija, Tartu Ülikool

Teadustegevus

Peamised uurimisvaldkonnad:

- Kodeerimisteooria
- Krüptograafia

DISSERTATIONES INFORMATICAE PREVIOUSLY PUBLISHED IN DISSERTATIONES MATHEMATICAE UNIVERSITATIS TARTUENSIS

19. **Helger Lipmaa.** Secure and efficient time-stamping systems. Tartu, 1999, 56 p.
22. **Kaili Müürisep.** Eesti keele arvutigrammatika: süntaks. Tartu, 2000, 107 lk.
23. **Varmo Vene.** Categorical programming with inductive and coinductive types. Tartu, 2000, 116 p.
24. **Olga Sokratova.** Ω-rings, their flat and projective acts with some applications. Tartu, 2000, 120 p.
27. **Tiina Puolakainen.** Eesti keele arvutigrammatika: morfoloogiline ühestamine. Tartu, 2001, 138 lk.
29. **Jan Villemson.** Size-efficient interval time stamps. Tartu, 2002, 82 p.
45. **Kristo Heero.** Path planning and learning strategies for mobile robots in dynamic partially unknown environments. Tartu 2006, 123 p.
49. **Härmel Nestra.** Iteratively defined transfinite trace semantics and program slicing with respect to them. Tartu 2006, 116 p.
53. **Marina Issakova.** Solving of linear equations, linear inequalities and systems of linear equations in interactive learning environment. Tartu 2007, 170 p.
55. **Kaarel Kaljurand.** Attempto controlled English as a Semantic Web language. Tartu 2007, 162 p.
56. **Mart Anton.** Mechanical modeling of IPMC actuators at large deformations. Tartu 2008, 123 p.
59. **Reimo Palm.** Numerical Comparison of Regularization Algorithms for Solving Ill-Posed Problems. Tartu 2010, 105 p.
61. **Jüri Reimand.** Functional analysis of gene lists, networks and regulatory systems. Tartu 2010, 153 p.
62. **Ahti Peder.** Superpositional Graphs and Finding the Description of Structure by Counting Method. Tartu 2010, 87 p.
64. **Vesal Vojdani.** Static Data Race Analysis of Heap-Manipulating C Programs. Tartu 2010, 137 p.
66. **Mark Fišel.** Optimizing Statistical Machine Translation via Input Modification. Tartu 2011, 104 p.
67. **Margus Niitsoo**. Black-box Oracle Separation Techniques with Applications in Time-stamping. Tartu 2011, 174 p.
71. **Siim Karus.** Maintainability of XML Transformations. Tartu 2011, 142 p.
72. **Margus Treumuth.** A Framework for Asynchronous Dialogue Systems: Concepts, Issues and Design Aspects. Tartu 2011, 95 p.
73. **Dmitri Lepp.** Solving simplification problems in the domain of exponents, monomials and polynomials in interactive learning environment T-algebra. Tartu 2011, 202 p.

74. **Meelis Kull.** Statistical enrichment analysis in algorithms for studying gene regulation. Tartu 2011, 151 p.
77. **Bingsheng Zhang.** Efficient cryptographic protocols for secure and private remote databases. Tartu 2011, 206 p.
78. **Reina Uba.** Merging business process models. Tartu 2011, 166 p.
79. **Uuno Puus.** Structural performance as a success factor in software development projects – Estonian experience. Tartu 2012, 106 p.
81. **Georg Singer.** Web search engines and complex information needs. Tartu 2012, 218 p.
83. **Dan Bogdanov.** Sharemind: programmable secure computations with practical applications. Tartu 2013, 191 p.
84. **Jevgeni Kabanov.** Towards a more productive Java EE ecosystem. Tartu 2013, 151 p.
87. **Margus Freudenthal.** Simpl: A toolkit for Domain-Specific Language development in enterprise information systems. Tartu, 2013, 151 p.
90. **Raivo Kolde.** Methods for re-using public gene expression data. Tartu, 2014, 121 p.
91. **Vladimir Šor.** Statistical Approach for Memory Leak Detection in Java Applications. Tartu, 2014, 155 p.
92. **Naved Ahmed.** Deriving Security Requirements from Business Process Models. Tartu, 2014, 171 p.
94. **Liina Kamm.** Privacy-preserving statistical analysis using secure multi-party computation. Tartu, 2015, 201 p.
100. **Abel Armas Cervantes.** Diagnosing Behavioral Differences between Business Process Models. Tartu, 2015, 193 p.
101. **Fredrik Milani.** On Sub-Processes, Process Variation and their Interplay: An Integrated Divide-and-Conquer Method for Modeling Business Processes with Variation. Tartu, 2015, 164 p.
102. **Huber Raul Flores Macario.** Service-Oriented and Evidence-aware Mobile Cloud Computing. Tartu, 2015, 163 p.
103. **Tauno Metsalu.** Statistical analysis of multivariate data in bioinformatics. Tartu, 2016, 197 p.
104. **Riivo Talviste.** Applying Secure Multi-party Computation in Practice. Tartu, 2016, 144 p.
108. **Siim Orasmaa.** Explorations of the Problem of Broad-coverage and General Domain Event Analysis: The Estonian Experience. Tartu, 2016, 186 p.
109. **Prastudy Mungkas Fauzi.** Efficient Non-interactive Zero-knowledge Protocols in the CRS Model. Tartu, 2017, 193 p.
110. **Pelle Jakovits.** Adapting Scientific Computing Algorithms to Distributed Computing Frameworks. Tartu, 2017, 168 p.
111. **Anna Leontjeva.** Using Generative Models to Combine Static and Sequential Features for Classification. Tartu, 2017, 167 p.
112. **Mozhgan Pourmoradnasseri.** Some Problems Related to Extensions of Polytopes. Tartu, 2017, 168 p.

113. **Jaak Randmets.** Programming Languages for Secure Multi-party Computation Application Development. Tartu, 2017, 172 p.
114. **Alisa Pankova.** Efficient Multiparty Computation Secure against Covert and Active Adversaries. Tartu, 2017, 316 p.
116. **Toomas Saarsen.** On the Structure and Use of Process Models and Their Interplay. Tartu, 2017, 123 p.
121. **Kristjan Korjus.** Analyzing EEG Data and Improving Data Partitioning for Machine Learning Algorithms. Tartu, 2017, 106 p.
122. **Eno Tõnisson.** Differences between Expected Answers and the Answers Offered by Computer Algebra Systems to School Mathematics Equations. Tartu, 2017, 195 p.

DISSERTATIONES INFORMATICAE UNIVERSITATIS TARTUENSIS

1. **Abdullah Makkeh**. Applications of Optimization in Some Complex Systems. Tartu 2018, 179 p.
2. **Riivo Kikas**. Analysis of Issue and Dependency Management in Open-Source Software Projects. Tartu 2018, 115 p.
3. **Ehsan Ebrahimi**. Post-Quantum Security in the Presence of Superposition Queries. Tartu 2018, 200 p.
4. **Ilya Verenich**. Explainable Predictive Monitoring of Temporal Measures of Business Processes. Tartu 2019, 151 p.
5. **Yauhen Yakimenka**. Failure Structures of Message-Passing Algorithms in Erasure Decoding and Compressed Sensing. Tartu 2019, 134 p.
6. **Irene Teinemaa**. Predictive and Prescriptive Monitoring of Business Process Outcomes. Tartu 2019, 196 p.
7. **Mohan Liyanage.** A Framework for Mobile Web of Things. Tartu 2019, 131 p.
8. **Toomas Krips.** Improving performance of secure real-number operations. Tartu 2019, 146 p.
9. **Vijayachitra Modhukur.** Profiling of DNA methylation patterns as biomarkers of human disease. Tartu 2019, 134 p.
10. **Elena Sügis.** Integration Methods for Heterogeneous Biological Data. Tartu 2019, 250 p.
11. **Tõnis Tasa.** Bioinformatics Approaches in Personalised Pharmacotherapy. Tartu 2019, 150 p.
12. **Sulev Reisberg.** Developing Computational Solutions for Personalized Medicine. Tartu 2019, 126 p.
13. **Huishi Yin.** Using a Kano-like Model to Facilitate Open Innovation in Requirements Engineering. Tartu 2019, 129 p.
14. **Faiz Ali Shah.** Extracting Information from App Reviews to Facilitate Software Development Activities. Tartu 2020, 149 p.
15. **Adriano Augusto**. Accurate and Efficient Discovery of Process Models from Event Logs. Tartu 2020, 194 p.
16. **Karim Baghery.** Reducing Trust and Improving Security in zk-SNARKs and Commitments. Tartu 2020, 245 p.
17. **Behzad Abdolmaleki.** On Succinct Non-Interactive Zero-Knowledge Protocols Under Weaker Trust Assumptions. Tartu 2020, 209 p.
18. **Janno Siim.** Non-Interactive Shuffle Arguments. Tartu 2020, 154 p.
19. **Ilya Kuzovkin.** Understanding Information Processing in Human Brain by Interpreting Machine Learning Models. Tartu 2020, 149 p.
20. **Orlenys López Pintado.** Collaborative Business Process Execution on the Blockchain: The Caterpillar System. Tartu 2020, 170 p.
21. **Ardi Tampuu.** Neural Networks for Analyzing Biological Data. Tartu 2020, 152 p.

22. **Madis Vasser.** Testing a Computational Theory of Brain Functioning with Virtual Reality. Tartu 2020, 106 p.
23. **Ljubov Jaanuska.** Haar Wavelet Method for Vibration Analysis of Beams and Parameter Quantification. Tartu 2021, 192 p.
24. **Arnis Parsovs.** Estonian Electronic Identity Card and its Security Challenges. Tartu 2021, 214 p.
25. **Kaido Lepik.** Inferring causality between transcriptome and complex traits. Tartu 2021, 224 p.
26. **Tauno Palts.** A Model for Assessing Computational Thinking Skills. Tartu 2021, 134 p.
27. **Liis Kolberg.** Developing and applying bioinformatics tools for gene expression data interpretation. Tartu 2021, 195 p.
28. **Dmytro Fishman.** Developing a data analysis pipeline for automated protein profiling in immunology. Tartu 2021, 155 p.
29. **Ivo Kubjas.** Algebraic Approaches to Problems Arising in Decentralized Systems. Tartu 2021, 120 p.
30. **Hina Anwar.** Towards Greener Software Engineering Using Software Analytics. Tartu 2021, 186 p.
31. **Veronika Plotnikova.** FIN-DM: A Data Mining Process for the Financial Services. Tartu 2021, 197 p.
32. **Manuel Camargo.** Automated Discovery of Business Process Simulation Models From Event Logs: A Hybrid Process Mining and Deep Learning Approach. Tartu 2021, 130 p.
33. **Volodymyr Leno.** Robotic Process Mining: Accelerating the Adoption of Robotic Process Automation. Tartu 2021, 119 p.
34. **Kristjan Krips.** Privacy and Coercion-Resistance in Voting. Tartu 2022, 173 p.
35. **Elizaveta Yankovskaya.** Quality Estimation through Attention. Tartu 2022, 115 p.
36. **Mubashar Iqbal.** Reference Framework for Managing Security Risks Using Blockchain. Tartu 2022, 203 p.
37. **Jakob Mass.** Process Management for Internet of Mobile Things. Tartu 2022, 151 p.
38. **Gamal Elkoumy.** Privacy-Enhancing Technologies for Business Process Mining. Tartu 2022, 135 p.
39. **Lidia Feklistova.** Learners of an Introductory Programming MOOC: Background Variables, Engagement Patterns and Performance. Tartu 2022, 151 p.
40. **Mohamed Ragab.** Bench-Ranking: A Prescriptive Analysis Approach for Large Knowledge Graphs Query Workloads. Tartu 2022, 158 p.
41. **Mohammad Anagreh.** Privacy-Preserving Parallel Computations for Graph Problems. Tartu 2023, 181 p.
42. **Rahul Goel.** Mining Social Well-being Using Mobile Data. Tartu 2023, 104 p.

43. **Anti Ingel.** Algorithms using information theory: classification in brain-computer interfaces and characterising reinforcement-learning agents. Tartu 2023, 142 p.
44. **Shakshi Sharma.** Fighting Misinformation in the Digital Age: A Comprehensive Strategy for Characterizing, Identifying, and Mitigating Misinformation on Online Social Media Platforms. Tartu 2023, 158 p.
45. **Kristiina Rahkema.** Quality Analysis of iOS Applications with Focus on Maintainability and Security Aspects. Tartu 2023, 182 p.
46. **Ivan Slobozhan.** Studying Online Social Media Engagement in CIS Countries during Protests, Mass Demonstrations and War. Tartu 2023, 81 p.
47. **Nurlan Kerimov.** Building a catalogue of molecular quantitative trait loci to interpret complex trait associations. Tartu 2023, 248 p.
48. **Pavlo Tertychnyi.** Machine Learning Methods for Anti-Money Laundering Monitoring. Tartu 2023, 117 p.
49. **Abasi-amefon Obot Affia.** A Framework and Teaching Approach for IoT Security Risk Management. Tartu 2023, 180 p.
50. **Raimond-Hendrik Tunnel.** Video Game Design and Development Bachelor's Curriculum for Estonia. Tartu 2024, 137 p.
51. **Ahto Salumets.** Bioinformatics analysis of various aspects in immunology. Tartu 2024, 198 p.
52. **Mohammed Abdulhameed Shaif Ali.** Deep Learning Methods for Cell Microscopy Image Analysis. Tartu 2024, 143 p.
53. **Pille Pullonen-Raudvere**. Foundations of Efficient and Secure Algorithm Development for Secure Multiparty Computation. Tartu 2024, 265 p.
54. **Marili Rõõm.** Multiple approaches to learners' success and factors affecting it in computer programming MOOCs. Tartu 2024, 170 p.
55. **Shivananda Rangappa Poojara.** Design and Orchestration of Scalable, Event-Driven Serverless Data Pipelines for Internet of Things (IoT) Applications. Tartu 2024, 172 p.
56. **Hassan Abdulgaleel Hassan Salim Eldeeb.** Empowering Machine Learning Pipelines with Automated Feature Engineering. Tartu 2024, 121 p.
57. **Muhammad Uzair.** Soft decision making for agri-food 4.0. Tartu 2024, 158 p.
58. **Kirill Milintsevich.** Estimation of Depression Level from Text: Symptom-Based Approach, External Knowledge, Dataset Validity. Tartu 2024, 130 p.
59. **Maksym Del.** Multilingual and Multi-Domain Representational Patterns Across Trpansformer-Based Models. Tartu 2024, 131 p.
60. **Kristo Raun.** Adaptive Out-of-order Handling in Streaming Conformance Checking. Tartu 2024, 118 p.
61. **Toivo Vajakas.** Towards integration of mobile network data into analyzing human mobility. Tartu 2024, 103 p.
62. **Katsiaryna Lashkevich.** Data-Driven Analysis and Optimization of Waiting Times in Business Processes. Tartu 2024, 169 p.
63. **Alejandra Duque-Torres.** Classifying, Constraining and Ranking Metamorphic Relations. Tartu 2025, 159 p.

64. **Mariia Bakhtina.** A Method for Information Security and Privacy Management in Smart Solutions. Tartu 2025, 199 p.
65. **Andre Tättar.** Multilingual Machine Translation for Under-Resourced Languages. Tartu 2025, 170 p.
66. **Mahmoud Shoush.** Prescriptive Process Monitoring Under Uncertainty and Resource Constraints. Tartu 2025, 178 p.
67. **Alireza Akhavi Zadegan.** A Multimodal approach for refining Mapping and Localization by Integrating Generative AI and Pedestrian-Centric Data. Tartu 2025, 147 p.
68. **Eerik Muuli.** Automating the assessment and feedback processes in IT teaching – improving creation and maintenance from the teaching staff perspective. Tartu 2025, 196 p.
69. **Kateryna Kubrak.** Towards User-Centered Prescriptive Process Monitoring Systems. Tartu 2025, 151 p.
70. **Zhigang Yin.** Computing and Sensing in a Smart Ring. Tartu 2025, 251 p.
71. **Abdul-Rasheed Olatunji Ottun.** Practical Trustworthy Artificial Intelligence with Human Oversight. Tartu 2025, 239 p.