The Advanced Decoding Schemes for Low-density Parity-check Codes

Peng Kang

A thesis submitted to the Graduate Research School of
The University of New South Wales
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

School of Electrical Engineering and Telecommunications
Faculty of Engineering
The University of New South Wales

September 2019

Thesis/Dissertation Sheet

Surname/Family Name	:	**Kang**
Given Name/s	:	**Peng**
Abbreviation for degree as give in the University calendar	:	**Ph.D.**
Faculty	:	**Faculty of Engineering**
School	:	**School of Electrical Engineering and Telecommunications**
Thesis Title	:	**The Advanced Decoding Schemes for Low-density Parity-check Codes**

Abstract

In this thesis, a new code construction method and two enhanced decoding schemes for low-density parity-check (LDPC) codes were proposed to improve their error performance.

In the first part of the thesis, we develop a new construction method of binary spatially-coupled (SC) LDPC codes for storage applications based on Euclidean geometry (EG) and a two-dimensional edge-spreading process. We show that the error performance of the constructed EG-SC LDPC codes is superior to that of their EG LDPC code counterparts, and there is no error floor compared to the protograph SC LDPC codes and regular LDPC codes. To achieve a high error correction capability, we further propose a reliability-based windowed decoding (RBWD) scheme for the SC LDPC codes based on a partial message reservation method and a partial syndrome check stopping rule. It is shown that the RBWD scheme significantly improves the error floor performance compared to the sliding window decoder with the conventional WBF algorithm.

We investigate the decoding of the LDPC codes adopted in the fifth generation (5G) mobile networks. We propose a two-dimensional scale-corrected min-sum algorithm based on partial self-correction and message amplification, which results in the error performance near the sum-product algorithm (SPA). Then we present an enhanced quasi-maximum likelihood (EQML) decoding method to further improve the error performance for 5G short LDPC codes. The proposed decoding method performs node selection with multiple rounds of decoding tests. We also present a partial pruning stopping rule to reduce the decoding complexity and derive a lower bound on the error performance. We show that the EQML decoding method outperforms the SPA with the same decoding complexity and approaches the Polyanskiy-Poor-Verdú bound within 0.4 dB.

We also design a decoding scheme based on the approximate message passing (AMP) algorithm for the 5G LDPC codes. We propose a decoding model, which aims to recover the error vector from the output of the decoder. When the initial decoding attempt fails, the AMP detector estimates the reliability of the channel outputs and flips the signs of unreliable channel outputs to create the new input sequence for one more decoding test. We show that the proposed AMP-aided decoding scheme achieves a 0.1 dB gain over the counterpart of one-time decoding for various block lengths and low code rates and the obtain about 0.05 dB gain for the EQML decoding.

ORIGINALITY STATEMENT

‘I hereby declare that this submission is my own work and to the best of my knowledge it contains no materials previously published or written by another person, or substantial proportions of material which have been accepted for the award of any other degree or diploma at UNSW or any other educational institution, except where due acknowledgement is made in the thesis. Any contribution made to the research by others, with whom I have worked at UNSW or elsewhere, is explicitly acknowledged in the thesis. I also declare that the intellectual content of this thesis is the product of my own work, except to the extent that assistance from others in the project's design and conception or in style, presentation and linguistic expression is acknowledged.’

Signed ……………………………………………..............

Date ……………………………………………..............

INCLUSION OF PUBLICATIONS STATEMENT

UNSW is supportive of candidates publishing their research results during their candidature as detailed in the UNSW Thesis Examination Procedure.

Publications can be used in their thesis in lieu of a Chapter if:

- The candidate contributed greater than 50% of the content in the publication and is the "primary author", ie. the candidate was responsible primarily for the planning, execution and preparation of the work for publication
- The candidate has approval to include the publication in their thesis in lieu of a Chapter from their supervisor and Postgraduate Coordinator.
- The publication is not subject to any obligations or contractual agreements with a third party that would constrain its inclusion in the thesis

Please indicate whether this thesis contains published material or not:

☐ This thesis contains no publications, either published or submitted for publication

☒ Some of the work described in this thesis has been published and it has been documented in the relevant Chapters with acknowledgement

☐ This thesis has publications (either published or submitted for publication) incorporated into it in lieu of a chapter and the details are presented below

CANDIDATE'S DECLARATION

I declare that:

- I have complied with the UNSW Thesis Examination Procedure
- where I have used a publication in lieu of a Chapter, the listed publication(s) below meet(s) the requirements to be included in the thesis.

Candidate's Name	Signature	Date (dd/mm/yy)

COPYRIGHT STATEMENT

Signed ……………………………………………..........................

Date ……………………………………………..............................

AUTHENTICITY STATEMENT

'I certify that the Library deposit digital copy is a direct equivalent of the final officially approved version of my thesis.'

Signed ……………………………………………..........................

Date ……………………………………………..............................

Dedicated to my parents and wife.

Abstract

In this thesis, we mainly investigate the construction method and the advanced decoding schemes for low-density parity-check (LDPC) codes to improve their error performance. Various construction methods and decoding schemes for LDPC codes have been well studied and investigated in the literature. However, with the increase of the new applications, such as the low-cost NAND flash memories, Internet of things, and ultra-high-speed communications, these classical coding techniques still face new challenges in term of the complexity and robustness. We address these problems by providing a new construction method and several improved decoding schemes for the LDPC codes. The key idea is to evaluate the reliability of the received messages properly and utilize these reliable messages to improve the error performance. The research work described in this thesis consists of three parts.

In the first part of the thesis, we develop a new construction method of binary spatially-coupled (SC) LDPC codes based on Euclidean geometry (EG) for storage applications. In the construction method, we propose a two-dimensional edge-spreading process to generate a base matrix for the SC LDPC codes. Then the parity-check matrix of the constructed LDPC code is obtained by unwrapping the base matrix and the lifting operation. We evaluate the error performance of the constructed EG-SC LDPC codes by using a weighted bit-flipping (WBF) decoding algorithm for its low decoding complexity. We show that the error

performance of the constructed EG-SC LDPC codes is superior to that of their EG LDPC code counterparts, and there is no error floor compared to the constructed protograph SC LDPC codes and regular LDPC codes. To achieve a high error correction capability and low decoding complexity, we further propose a reliability-based windowed decoding (RBWD) scheme for the SC LDPC codes. A partial message reservation method is adopted in the RBWD scheme to mitigate the error propagation. And a partial syndrome check stopping rule is also introduced for each decoding window to further reduce the error floor. It is shown that our proposed scheme significantly improves the error floor performance compared to the sliding window decoder with the conventional WBF algorithm.

Next, we investigate the decoding of the LDPC codes adopted in the fifth generation (5G) networks standard. To be more specific, we propose a two-dimensional scale-corrected min-sum algorithm for 5G LDPC codes to approach the error performance of the sum-product algorithm (SPA) with low decoding complexity. In the proposed decoding algorithm, we adopt a partial self-correction method followed by message amplification to improve the reliability of the variable-to-check (V2C) messages. Simulation results show that the proposed decoding algorithm can approach the frame error performance of the SPA by using a pair of universal scaling factors. After that, we present an enhanced quasi-maximum likelihood (EQML) decoding method based on the proposed decoding algorithm to further approach the error performance of the maximum likelihood decoding for 5G short LDPC codes. The proposed decoding method performs multiple rounds of decoding tests once the first decoding attempt fails, where the decoder inputs of the selected unreliable variable nodes are modified in each decoding test. A novel node selection method based on the sign fluctuation of V2C messages is proposed for the EQML decoding method. We also introduce a partial pruning stopping rule to reduce the decoding complexity by deactivating part of the decoding tests once a valid codeword is found. A lower bound on the error performance is derived by using the semi-analytical method to predict the error performance of the EQML decoding method. We show that the proposed EQML decoding method also outperforms the SPA with the same decoding complexity and approaches the Polyanskiy-Poor-Verdú bound within 0.4 dB.

In the last part of the thesis, we also design a detector aided decoding scheme based on the approximate message passing (AMP) algorithm for the 5G LDPC codes with code rate less than 1/2. In the proposed decoding scheme, we propose a system model, which aims to recover the error vector from the output of the decoder. The AMP detector estimates the reliability of the channel output for each node when the initial decoding attempt fails. The signs of the channel outputs on the unreliable VNs are flipped according to the preset threshold to generate the updated decoder input sequence and one decoding test is conducted afterwards. We show that the proposed AMP-aided decoding scheme achieves a 0.1 dB gain over the counterpart of one-time decoding for the 5G LDPC codes with various block lengths and low code rates. In addition, attributing to the AMP detector, the error performance under the conventional MSA can approach that of SPA within 0.3 dB for all simulated LDPC codes. Moreover, we also propose an AMP-Enhanced Quasi-Maximum Likelihood (EQML) decoding scheme for the decoding of 5G LDPC codes. The AMP detector is conducted for the unsuccessful decoding tests in the reprocessing of the EQML decoding to further improve the error performance. Some properties of the proposed AMP-aided decoding scheme are also analyzed and we also show that the proposed AMP-EQML decoding scheme outperforms EQML decoding for information bit length of $K = 320$ and 752.

Acknowledgments

First of all, I am grateful to my supervisor, Professor Jinhong Yuan, who has provided me constant support and exemplary guidance from the beginning of my Ph.D. study. His deep understanding of the coding theory has always allowed the pursuit of different ideas to some meaningful research results. His enthusiasm for the research and constructive suggestions on my research work have always inspired me to explore research problems and become an independent researcher. His dedication to me is one of my most fortunate during my Ph.D. study.

Secondly, I would like to thank my co-supervisor, Dr. Yixuan Xie, for his guidance, numerous discussions and technical supports on programming and materials on channel coding techniques. Dr. Xie has always been supportive and available for discussion. I would never forget the countless times of discussion when I was struggling with my research topics in my first year. I would like to thank Dr. Lei Yang for his useful guidance on my paper writing. I would also like to thank Dr. Derrick Wing Kwan Ng, for his suggestions on my research planning.

"We are family, say it proudly": Lots of thanks to my colleagues in the wireless communication group of the University of New South Wales. Especially, I want to thank Lou Zhao, Min Qiu, Zhuo Sun, Zhiqiang Wei, Bryan Liu, Xiaowei Wu, Ruide Li, Yihuan Liao, Yuanxin Cai, and Shuangyang Li. We studied together, played together, shared our life experience, and grew up together, which brought me unforgettable memories.

Finally, I would like to express sincere gratitude to my family for their unselfish love and unconditional support. My parents have always been there to help me and encourage me when I had a hard time. My grandparents have always given me strong support in mentally. Especially, I would like to thank my wife, Lingjie Xu, for sacrificing her life for the whole family, and enabled me to remain motivated throughout my Ph.D. study.

Without you all, this thesis would not have been accomplished.

List of Publications

Journal Articles:

1. Y. Xie, L. Yang, **P. Kang**, and J. Yuan, "Euclidean geometry-based spatially coupled LDPC codes for storage," *IEEE J. Select. Areas Commun.*, vol. 34, pp. 2498–2509, Sep. 2016.

2. **P. Kang**, Y. Xie, L. Yang, and J. Yuan, "Reliability-based windowed decoding for spatially coupled ldpc codes," *IEEE Commun. Lett.*, vol. 22, pp. 1322–1325, Jul. 2018.

3. **P. Kang**, Y. Xie, L. Yang, C. Zheng, J. Yuan, and Y. Wei, "Enhanced quasi-maximum likelihood decoding for 5G LDPC codes," submitted to *IEEE Trans. Commun.*.

Conference Articles:

1. **P. Kang**, Y. Xie, L. Yang, C. Zheng, J. Yuan, and Y. Wei, "Enhanced quasi-maximum likelihood decoding of short LDPC codes based on saturation," in *Proc. IEEE Inf. Theory Workshop*, pp. 1–6, Aug. 2019.

Patents:

1. **P. Kang**, Y. Xie, L. Yang, C. Zheng, J. Yuan, and Y. Wei, "A method for improved decoding of low-density parity-check codes," Ref. No. 201811279697.5.

2. **P. Kang**, Y. Xie, L. Yang, C. Zheng, J. Yuan, and Y. Wei, "A method for enhanced decoding of low-density parity-check codes," Ref. No. 201811279838.3.

Industrial Reports:

1. **P. Kang**, Y. Xie, and J. Yuan, "Improved Min-Sum Decoding of LDPC Codes," Project: Enhanced Decoding Algorithm for 5G LDPC Codes - Stage 1, Huawei Technology CO., LTD, Shanghai, China.

2. **P. Kang**, Y. Xie, and J. Yuan, “Improved Quasi-Maximum Likelihood Decoder of LDPC Codes,” Project: Enhanced Decoding Algorithm for 5G LDPC Codes - Stage 2, Huawei Technology CO., LTD, Shanghai, China.

3. **P. Kang**, Y. Xie, Z. Sun, and J. Yuan, “The AMP-aided Decoding Scheme of 5G LDPC Codes,” Project: Enhanced Decoding Algorithm for 5G LDPC Codes - Stage 3, Huawei Technology CO., LTD, Shanghai, China.

Abbreviations

2D	Two dimensional
3GPP	3rd Generation Partnership Project
4G	The Fourth-generation
5G	The Fifth-generation
ABP	Augmented belief propagation
ACE	Approximate cycle extrinsic
AMP	Approximate message passing
APP	A priori probability
AWGN	Additive white Gaussian noise
BCH	Bose-Chaudhuri-Hocquengham
BER	Bit error rate
BI-AWGN	Binary-input additive white Gaussian noise
BSC	Binary symmetric channel
BP	Belief propagation
BPSK	Binary phase shift keying
C2V	Check-to-variable
CN	Check node
CPM	Circulant permutation matrix
dB	Decibel
DE	Density Evolution
ECC	Error correcting code
EG	Euclidean geometry
eMBB	Enhanced mobile broadband

EQML	Enhanced quasi-maximum likelihood
EXIT	Extrinsic information transfer
EWS	Edge-wise Selection
FBD	Full block decoding
FER	Frame error rate
GF	Galois field
HDD	Hard-disk drives
i.i.d.	Independent and identically distributed
IoT	Internet-of-things
LDS	List stopping rule
LDPC	Low-density parity-check
LLR	Log-likelihood ratio
LTE	Long-term evolution
MAP	Maximum a posterior
MBF	Multi-bit flipping
ML	Maximum-likelihood
MLC	Multi-level cell
mMIMO	Massive multiple-input multiple-output
mMTC	Massive machine type communications
mmWave	Millimeter wave
MIMO	Multiple-input multiple-output
MMSE	Minimum mean square error
MSA	Min-sum algorithm
MSE	Mean squared error
NMSA	Normalized min-sum algorithm
NOMA	Non-orthogonal multiple access
NSW	Node-wise Selection
OMSA	Offset min-sum algorithm
OSD	Ordered statistic decoding

PBRL	Protograph-based raptor-like
PDF	Probability density function
PEG	Progressive edge-growth
PMR	Partial message reservation
PPS	Partial pruning stopping
QAM	Quadrature amplitude modulation
QML	Quasi-maximum likelihood
QPSK	Quadrature phase shift keying
RBER	Raw bit error rate
RBWD	Reliability-based windowed decoding
RS	Reed-Solomon
SBF	Single-bit flipping
SC	Spatially-coupled
SMS	Saturated min-sum
SNR	Signal-to-noise ratio
SPA	Sum-product algorithm
uRLLC	Ultra reliable and low latency communications
UBER	Uncorrectable bit error rate
V2C	Variable-to-check
VN	Variable node
WBF	Weighted bit-flipping

List of Notations

Scalars, vectors and matrices are written in italic, boldface lower-case and upper-case letters, respectively, e.g., x, $\mathbf{x}$ and $\mathbf{X}$. Random variables are written in uppercase Sans Serif font e.g., X.

$\mathbf{X}^{\mathrm{T}}$	Transpose of $\mathbf{X}$
$\mathbf{X}^{H}$	Conjugate transpose of $\mathbf{X}$
$\mathbf{X}^{-1}$	Inverse of $\mathbf{X}$
$\mathbf{X}_{i,j}$	The element in the row i and the column j of $\mathbf{X}$
$\mathrm{rank}\,(\mathbf{X})$	Rank of a matrix $\mathbf{X}$
$\|x\|$	Absolute value (modulus) of the complex scalar x
$\\|\mathbf{x}\\|$	The Euclidean norm of a vector $\mathbf{x}$
$\mathbb{R}$	The field of real number
$\mathbb{C}$	The field of complex number
$\mathbb{R}^n$	The Euclidean space
$\mathbb{F}_q$	The finite field of size q
$\mathbb{Z}$	The ring of integers
$\mathbb{Z}^+$	The ring of positive integers
$\mathbb{Z}[i]$	The ring of Gaussian integers
$\mathbb{Z}[\omega]$	The ring of Eisenstein integers
$\mathrm{Pr}(E), \mathrm{P}(E)$	The probability of event E occurs
$\mathrm{Pr}(E_1\|E_2), \mathrm{P}(E_1\|E_2)$	The conditional probability of event E_1 occurs given event E_2 occurs
$p_{\mathsf{X}}(x), p(x)$	Probability density function of the random variable X
$p_{\mathsf{X}\|\mathsf{Y}}(x\|y), p(x\|y)$	Conditional distribution of X given Y
$L_{\mathsf{X}}(x), L(x)$	Log-likelihood ratio of the random variable X

$L_{\mathsf{X}\mid\mathsf{Y}}(x\|y), L(x\|y)$	Conditional log-likelihood ratio of X given Y
$\Re(z)$	The real part of a complex number z
$\Im(z)$	The imaginary part of a complex number z
$\|\mathcal{S}\|$	The cardinality of a set $\mathcal{S}$
$\mathbf{0}$	The all-zero vector
$\mathbf{I}_{\mathrm{N}}$	N dimension identity matrix
$\mathbb{E}\{\cdot\}$	Statistical expectation
$\mathrm{Var}\{\cdot\}$	Statistical expectation
$\mathcal{N}(\mu, \sigma^2)$	Real Gaussian random variable with mean μ and variance σ^2
$\ln(\cdot)$	Natural logarithm
$\log_a(\cdot)$	Logarithm in base a
$\mathrm{diag}\{\boldsymbol{a}\}$	A diagonal matrix with the entries of $\boldsymbol{a}$ on its diagonal
$\lim$	Limit
$\max\{\cdot\}$	Maximization
$\min\{\cdot\}$	Minimization
$\max(a, b)$	Take the larger number in a and b
$e^x, \exp(x)$	Natural exponential function
$\tanh$	Hyperbolic tangent function
$\mathrm{sign}(x)$	Take the sign of x
$\mathrm{mod}(a, b)$	a modulo b
$d_H(\mathbf{x}, \mathbf{y})$	The Hamming distance between vector $\mathbf{x}$ and $\mathbf{y}$
$d_E(\mathbf{x}, \mathbf{y})$	The Euclidean distance between vector $\mathbf{x}$ and $\mathbf{y}$
$\otimes$	Convolution

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

At the end of the 20th century, mobile communication technology is one of the most important technologies to promote the rapid development of human society, which have a significant impact on people's lifestyles, working and economics. In fact, mobile communications have grown rapidly over the past 30 years after the invention of the cellular concept, which is a big breakthrough in solving the problem of capacity and coverage of mobile communication systems [2]. Since the successful implementation of the first analog cellular mobile phone system in Chicago in 1979, mobile communications have experienced four generations [2–5] and is moving toward the 5G mobile networks [1, 6–8].

In the coming 5G era, it is notable that the mobile Internet and the Internet of Things (IoT) have become the two driving forces for the development of mobile communications, which profoundly affects various aspects of people's daily life. In particular, the application scenarios in 5G mobile networks can be divided into three categories:

- Enhanced Mobile BroadBand (eMBB) for mobile Internet, which can provide the subscribers with ultra-high data rate and traffic density requirements across a wide coverage area. [9]

- Massive Machine Type Communications (mMTC) for IoT, which can support over 1 million/km^2 connection density for a huge number of low-cost IoT devices [10, 11].

- Ultra-Reliable and Low Latency Communications (uRLLC), which provide extremely reliable communications for the applications and services with strictly low end-to-end delay requirement less than 1ms [12].

To satisfy the increasing demands of mobile Internet and IoT services, the 5G communication systems face enormous challenges. Many development works of 5G technologies have drawn significant efforts and been investigated by a number of researchers [13–15]. Some advanced transmission technologies such as non-orthogonal multiple access (NOMA) [13, 16–31], millimeter wave (mmWave) [13, 32–37], and massive multiple-input multiple-output (mMIMO) [38–43] are determined to be the key technologies for wireless transmission in 5G frameworks. Behind these technologies, channel coding always plays an important role in communication systems.

The establishment of information and channel coding theory originates from Shannon's landmark paper proposed in 1948 [44]. In this famous paper, Shannon shows that as long as the transmission rate is less than the channel capacity, for any type of channel, there exists a coding scheme that can achieve an arbitrary small error probability for the communication system with a sufficiently large code block length. This theory, so-called the noisy channel coding theorem, states that the capacity of channels can be reached by channel coding technologies, which can be regarded as a guideline for the design of channel coding scheme. After that, constructing channel codes that can approach the channel capacity under effective decoding algorithms with desirable complexity has been a long-term goal pursued by many researchers [45–50].

Since the purpose of adopting channel codes is to correct the errors when the data is transmitted over unreliable or noisy communication channels, they are also called error correction codes. In the past few decades, block codes [51, 52] and convolutional codes [53, 54], which are two types of error correction codes, have been thoroughly studied and widely used in communication and storage systems. In 1950, the mathematician Hamming proposed the first practical linear block code, i.e., the (7,4) Hamming code [55]. This method divides the input data into 4 groups of bits, and then linearly combines the information bits to obtain 3 parity-check bits to form a 7-bit codeword. The use of parity-check bits can not only detect transmission errors but also correct a single random error. Hereafter, the block codes started to attract the interest of many researchers and been quickly developed into a systematic coding theory. For example, Reed and Muller proposed a new block code, Reed-Muller code [56] in 1954, which has strong flexibility in terms of code block lengths and error correction capability. In 1957, Prange proposed an important class of block codes, namely cyclic codes [57], where the generated codewords have cyclic shift characteristic. This codeword structure greatly simplifies the encoding and decoding process and makes it friendly for hardware implementations. A famous member of the cyclic codes is the BCH codes proposed by Bose, Ray-Chaudhuri and Hocquenghem in 1959 $\sim$ 1960 [58, 59], which can correct multiple random errors. In the same year, Reed and Solomon constructed a class of q-ary BCH codes with strong error correction capability, namely the Reed-Solomon codes [60]. The most outstanding advantage of RS codes is that they are ideal for correcting burst errors and thus are widely used in CD/DVD players, digital video broadcasting (DVB) [61], and digital subscriber line (DSL) standards [62].

The convolutional code [63] is another type of error correction code developed along with the block code. The encoding of the convolutional code is not only related to the current input bits but also related to several input bits at the previous time. The most commonly used decoding algorithm is the Viterbi algorithm [64], which is a maximum likelihood decoding based on the trellis diagram of a convolutional code and can be performed continuously on a bit stream. Thus, the decoding delay for convolutional codes is relatively small compared to that of block codes. In addition, there are some channel coding schemes proposed by combining more than one error correction code into a single design. For instance, the product code proposed by Elias [65] uses two linear block codes as component codes to construct a channel code with long block lengths from short codes, which can correct multiple bursts of errors. Forney proposed the serial concatenated codes [66] to construct long codes from short component codes, where the inner codes are serially cascaded with the outer codes. As shown in [66], the concatenated codes can achieve exponentially decreasing error probabilities at all data rates less than capacity, with decoding complexity that increases only polynomially with the code block lengths.

Although the above channel coding technologies have been well-developed for different applications, they still operate far away from the channel capacity. With the increasing performance requirements of communication systems, modern error correction codes are developed with more powerful error correction capability, which have error performance close to the channel capacity. Among the modern error correction codes, Turbo codes [67–69], LDPC codes [46, 70, 71], and polar codes [72] are of great impacts on both academic and industry. As originally invented by Gallager in 1962, LDPC codes are a type of error correction code with long block lengths and can be decoded efficiently by using the low-complexity iterative belief-propagation (BP) decoding algorithms [70]. However, due to the limitations of hardware and software, LDPC codes had not drawn much attention until MacKay *et al.* found that LDPC codes have decoding performance close to channel capacity under iterative BP decoding algorithms [45, 71]. Later in 2001, Richardson *et al.* proposed the well-known density evolution (DE) algorithm to optimize the degree distribution of irregular LDPC codes [73]. Based on DE, the designed 1/2-rate irregular LDPC code in the AWGN channel has a performance gap within 0.0045 dB from the capacity, which can be regarded as another milestone after the rediscovery of LDPC codes. The most remarkable advantage of LDPC codes is the low decoding complexity and suitable for parallel decoding [74], which can achieve high throughput decoding.

In the fourth generation (4G) mobile networks, which is known as the Long Term Evolution (LTE), turbo codes [67, 68] and tail-biting convolutional codes [4, 53] are used as the main coding schemes for data channels and control channels, respectively. While in the fifth generation (5G) mobile works, LDPC codes are eventually designed and used for the eMBB scenario [1]. Compared to 4G mobile networks, the envisioned applications of 5G mobile networks impose more stringent requirements for communication systems in terms of throughput, delay, and reliability, which give rise to new challenges for the channel coding schemes. Therefore, how to design proper LDPC coding schemes in order to achieve the high performance requirements of communication systems in 5G and beyond has become an attractive research problem. As such, this thesis aims to investigate LDPC coding techniques from the perspective of code construction and decoding methods. In the following, we will discuss the motivation for each research work.

1.1 Motivation

1.1.1 Design of Low-Density Parity-Check Coding Schemes with High Error Correction Capability

Digital communication systems and digital storage systems widely use flash memories, as a type of memory device characterized by non-volatility for data retention in power-off situations to achieve the reliability requirements. In the non-volatile memory application fields, NAND flash memory plays an increasingly important role over the past three decades for their higher throughput and lower power consumption compared to conventional hard-disk drives (HDDs) [75]. However, the cost of NAND Flash for the same storage capacity is still much higher than that of HDDs these days. To solve this problem, many technologies such as multi-level cell (MLC), triple-level cell (TLC), and 3D stacking are developed for NAND flash memory [76, 77], where there is more information packed in one storage element or more storage elements packed together. However, as more information per storage element or more storage elements are packed in a small package, the error rate of the stored information and the endurable program/erasure cycles of the storage cells will deteriorate. To deal with this issue, many powerful error control codes (ECCs), such as Bose-Chaudhuri-Hocquenghem (BCH) codes [78], concatenated codes [79], product codes [80, 81] and low-density parity-check (LDPC) codes [82] [83], were proposed and studied in the literature and in industry. Among these ECCs, LDPC codes are in favor by many researchers for their capacity-approaching property when soft information is available [70] [84].

Recently, spatially-coupled (SC) LDPC codes [85], which are a kind of convolutional LDPC codes [86], attracted a lot of researchers' attention [87–92]. SC LDPC codes combine the capacity-achieving property of irregular LDPC codes and the linear minimum distance growth property of regular LDPC codes [89]. More importantly, as shown in [85], SC LDPC codes show a significant convolutional gain compared to the associated block LDPC codes for the same circulant size. Even for the same structured decoding latency or the same decoder complexity, SC LDPC codes also have a considerable coding gain compared to their uncoupled counterparts, especially at high signal-to-noise-ratio (SNR) region [85,88]. Since the next-generation NAND Flash memories require both high error correction capability and very low uncorrectable error rate, it is of great interest to consider SC LDPC codes to achieve superior error performance.

It is known that SC LDPC codes can be constructed from LDPC block codes [85]. For the construction of block LDPC codes, algebraic construction methods, especially those based on finite geometries [46], provide a systematic way to construct a large family of girth-6 LDPC codes with various code lengths and code rates. These codes have near-capacity performance and low error floor. Instead of above merits, the LDPC codes constructed based on finite geometries can be decoded with various decoding algorithms, such as majority logic decoding algorithms, iterative message-passing decoding algorithms, which provide flexible options for different applications to practical systems [74]. Among the LDPC codes constructed based on finite geometry, Euclidean geometry (EG) LDPC codes [74] have relative good error performance under low complexity decoding algorithms, such as the conventional weighted bit-flipping algorithm [46], which can benefit to the high-speed low-latency decoding for NAND Flash memories. Hence, constructing binary SC LDPC codes based on EG LDPC block codes has great potential to satisfy the requirements of both error performance and decoding latency for NAND Flash memories.

In addition, it is shown in [93] that a sliding windowed decoder is an efficient decoding architecture for SC LDPC codes in the sense that only a portion of one codeword is decoded at a time. Compared to the full block decoding (FBD) [91, 94] which takes the entire codeword for decoding, a sliding windowed decoder has a lower decoding latency and memory requirement. However, most of the previous work, such as [95–97] considered improving the error performance of the sliding windowed decoder by using soft-decision decoding algorithms such as sum-product algorithm (SPA) [45]. This causes a high decoding complexity and more power consumption for NAND Flash memories as soft information is used to perform message update [98,99]. Therefore, the reliability-based decoding algorithms were proposed in [74, 100, 101], which only requires hard information when performing message update and can obtain a lower decoding complexity compared to the SPA at the expense of performance loss. It is worth noting that there is a significantly high error floor when the conventional WBF algorithm is used for windowed decoding of SC LDPC codes. This is because there exist variable nodes (VNs) that have neighboring check nodes (CNs) outside the decoding window due to the windowed decoding architecture. Consequently, the messages sent out from these VNs may not be reliable. More severely, these unreliable messages are propagated along with the decoding window and deteriorate the error performance. As such, it is of great necessity to design a windowed decoding architecture for the reliability-based decoding algorithms to mitigate or eliminate the error propagation of the unreliable messages.

1.1.2 Design of Advanced Decoding Methods for 5G LDPC Codes

In the 5G frameworks, LDPC codes have been selected to be one of the channel coding schemes for the data channels in the eMBB scenario [1]. As known from [1], the 5G LDPC codes have the maximum length of 8448 information bits with the code rates covering from $1/3 \sim 8/9$, and also support the code rate 1/5 when the length of information bits is not greater than 3840 bits. However, the decoding complexity will substantially increase for long information bits and low code rates. Thus, the error performance and decoding complexity should be both considered in the design of decoding methods to make a good balance between the reliability and implementation complexity.

It is well-known that the SPA [45] has good error performance by iteratively performing messages update between variable nodes (VNs) and check nodes (CNs). However, its decoding complexity is fairly high for hardware implementations as the nonlinear operations are performed for messages update [98]. Therefore, simplified versions of the SPA, such as the conventional min-sum algorithm (MSA) [102], and MSA-based decoding algorithms [47–50], such as the normalized MSA (NMSA) [47] and the offset MSA (OMSA) [48] are proposed and widely used in practical communication systems. More specifically, the conventional MSA [102] reduces the decoding complexity on CN update by only taking the magnitude and sign of the messages, which causes a distinct performance degradation relative to the SPA. By improving the estimation accuracy for the messages computed at the CN or VN update, the MSA-based decoding algorithms were proposed in [47–50] to obtain the error performance near that of the SPA for the moderate to long LDPC codes with high rates. Nonetheless, the error performance of these MSA-based decoding algorithms is highly affected by the VNs with low degrees [103]. As shown in [1], there are a number of VNs with degree-1 in their associated Tanner graphs [104] of the 5G LDPC codes. This results in a poor error performance of the MSA-based decoding algorithms compared to that of the SPA since the messages sent from the CNs connected to these degree-1 VNs are bounded in magnitude [105] at each iteration and become unreliable for these degree-1 VNs. Moreover, the performance degradation becomes more notable for the MSA-based decoding algorithms when using long information block length and low code rate (less than 1/2). This is because the portion of the degree-1 VNs is even higher compared to the VNs of other degrees. As a result, it is necessary to develop an improved decoding algorithm, particularly for the 5G LDPC codes with degree-1 VNs.

To achieve ultra-reliability and low decoding complexity under the constraint of limited end-to-end delay, the LDPC codes with short block lengths have been considered in the 5G frameworks [1]. Compared to the LDPC codes with long block lengths, there exist a number of unavoidable small cycles in the Tanner graphs of short LDPC codes. This allows the error messages to propagate within the cycles and severely deteriorate the error performance of short LDPC codes [106]. In this case, the SPA, which is considered to have asymptotically optimal performance, has a considerable performance gap from the maximum likelihood (ML) decoding. Therefore, several improved decoding methods, called the quasi-ML (QML) decoding methods, were investigated in [107–110], where the reprocessing architecture is introduced after the failure of the initial decoding attempt. In the reprocessing, part of the codeword is forced to be corrected by using the idea of list decoding, i.e., a list of all possible combinations for the selected VNs is generated and multiple rounds of the decoding or re-encoding tests are conducted thereafter. The 'best' codeword is chosen from the reprocessing output according to a certain decision metric. One of the famous QML decoding methods, so-called the ordered statistic decoding (OSD) [107], performs sorting and Gaussian elimination to generate the list of bit combinations for a predetermined number of VNs with the most reliability. Each combination is re-encoded into a codeword, and one of them is chosen as the decoding output according to the Euclidean distance with the received signal. However, there is a high decoding complexity caused by the sorting and matrix transformation operations, which makes the OSD unsuitable for hardware implementations. Alternatively, the QML decoding methods in [108–110] conducts multiple rounds of the decoding tests based on the modified decoder input sequences derived from the channel output sequence. As shown in [108], although the error performance of all these above QML decoding methods can be close to that of ML decoding with a sufficient large number of conducted decoding tests (more than 1000 decoding tests for one received signal), the performance gap to the ML decoding is still considerable when a small list size is used (less than 100 decoding tests for one received signal). Consequently, developing the advanced decoding methods for the 5G short LDPC codes becomes important to the 5G mobile networks.

1.2 Literature review

In this section, the related works of this thesis about the construction and decoding of the LDPC codes are discussed and reviewed.

1.2.1 The Construction of LDPC Codes

In general, LDPC codes consist of two categories: random/pseudo-random LDPC codes [51], and the structured LDPC codes [46]. Due to the cyclic or quasi-cyclic structure of their parity-check matrices, the structured LDPC codes have been widely used in many practical applications as they are much easier for hardware implementation. The construction methods of structured LDPC codes are mainly divided into:

- Computer-based (or graph-based) construction: using computer-aided algorithm such as progressive edge-growth (PEG) [111], and approximate cycle extrinsic (ACE) [112] to construct LDPC codes with good graph structures. A typical example of such LDPC codes is the protograph-based LDPC codes [113].

- Algebra-based (or matrix theory-based) construction: using algebraic tools, including finite fields, finite geometry, to construct LDPC codes.

It is known that the PEG algorithm is one of the most efficient computer-based methods to design the LDPC codes with large girth property [111]. For a given degree distribution of VNs, and the number of VNs and CNs, the PEG algorithm adds one edge on the code Tanner graph at a time and aims to maximize the local cycle length. Therefore, it is a greedy algorithm for constructing the Tanner graph of an LDPC code with large lengths of cycles. Nevertheless, the parity-check matrices of LDPC codes constructed by the PEG algorithm still have randomness, which increases the encoding and decoding complexity.

A much easier way to construct LDPC codes is generating a large Tanner graph from a small protograph [113], which is called the protograph-based construction method. For a given design code rate and the degree distribution of the LDPC codes, the protograph can be obtained based on computer searching and optimized through the mathematical tools such as density evolution [52] and extrinsic information transfer chart [114]. Then the Tanner graph of an LDPC code can be derived by the graph lifting operation [113], where the protograph is copied for U times and the edges of each independent graph are permuted between different copies. The major characteristic of protograph-based construction method is that the properties of the protograph will be inherited to its lifted Tanner graph such as good iterative decoding thresholds and linear minimum distance growth. As a consequence, for properly designed protographs, we can also have these excellent properties on their derived LDPC codes [115].

In addition, the algebra-based construction method constructs the cyclic shift matrix based on the given parameters of finite geometry [46]. Each element in the cyclic shit matrix is then replaced by a circulant permutation matrix with the same size according to the rule of matrix dispersion and masking [74] rules. As shown in [46], the LDPC codes constructed from finite geometry have good error performance for various low-complexity decoding algorithms, which is in favor of simple hardware implementation.

Apart from the construction methods mentioned above, spatially-coupled (SC) LDPC codes have been paid widespread attention by a lot of researchers [85, 89–91, 116, 117]. It is known that SC LDPC codes are a type of LDPC convolutional codes [86] which can be constructed from LDPC block codes by matrix unwrapping [85] and termination [89]. The graph structure of the associated LDPC block code is preserved for the unwrapped graph, which means all of the nodes remain their degrees and local connectivity compared to the underlying LDPC block code. The termination further introduces lower CN degrees at both sides of the unwrapped graph and causes slight structured irregularity. This results in an effect, called threshold saturation [118, 119], which demonstrates a dramatic improvement for the BP threshold of a regular SC LDPC code. More specifically, it shows that the BP threshold of a regular SC LDPC code can converge to the maximum a posteriori (MAP) threshold of its block code counterpart [120]. Note that the threshold refers to the worst channel parameter (e.g., SNR for additive white Gaussian noise channel) that leads the error probability after decoding converges to zero. Since the MAP decoding has the same performance as the ML decoding if all codewords are transmitted with equal probability, it is possible to achieve the ML threshold of the underlying LDPC block codes through spatial-coupling. Notice that ML decoding is the optimal decoding algorithm in terms of minimizing error probability. Moreover, as shown in [85], SC LDPC codes have good features of both regular and irregular LDPC codes that the BP threshold is close to the capacity and the minimum distance of the code grows linearly with the increase of block lengths.

1.2.2 The Decoding Algorithms of LDPC Codes

In addition to introducing LDPC codes, Gallager also presented a near-optimal decoding algorithm in [70], which is now called the sum-product algorithm (SPA) or BP algorithm [45]. The SPA is initially performed in probability domain, where the messages sent along an edge between VNs and CNs represent the posterior conditional distribution on the bit associated with the VN connected to that edge. Each VN or CN acts under the assumption that each incoming message at current iteration is conditionally independent of the others, and represents a conditional distribution on the associated received bit. Furthermore, only extrinsic information is used for computing the messages. The probability domain SPA uses a continuous message alphabet, which increases the implementation complexity for hardware. Therefore, the SPA in the logarithmic domain was proposed in [45], where the messages are represented by log-likelihood ratios (LLRs). However, due to the nonlinear operations performed during the update of messages, the SPA has a high decoding complexity. For practical considerations, some simplified versions of the SPA, such as the conventional min-sum algorithm (MSA) [102], the normalized MSA (NMSA) [47], and the offset MSA (OMSA) [48] are proposed and widely used in practical communication systems. To be more specific, the conventional MSA [102] reduces the decoding complexity by simplifying the message update rule at each iteration, which only considers the magnitudes and signs of the messages. However, it is shown in [102] that the magnitude of the messages computed by MSA is always greater than that computed by the SPA. This phenomenon is called the over-estimation and causes the performance degradation of the conventional MSA algorithm compared to the SPA. To reduce the performance gap between the conventional MSA and the SPA, the MSA-based algorithms are proposed in [47–49], where one or two correction terms are introduced to restrict the magnitude of the computed messages. Instead, the self-correct MSA [50] exploits the sign flipping phenomenon of VNs' extrinsic messages, where these messages can be sent if they have consistent signs in two consecutive iterations. Other than the decoding algorithms based on the soft decision as discussed above, Gallager also proposed a bit-flipping (BF) algorithm in [70], where the messages computed at each iteration are all quantized based on hard decision according to their signs. Compared to the SPA, the BF algorithm has a much lower decoding complexity and can be easily implemented. However, the error performance of the BF algorithm is far away from that of the SPA as shown in [46]. In [74], the conventional weighted BF (WBF) algorithm was proposed, which is a kind of reliability-based decoding algorithm. In the WBF algorithm, the fixed weights obtained from received signals are assigned to the checksums. A flipping metric is then computed for each VN according to the majority-logic. The least reliable VNs are flipped in priority at each iteration. Compared to the BF algorithm, the conventional WBF algorithm aims to provide better error performance closer to the SPA with slight increased computational complexity.

1.2.3 The Decoding Architectures of LDPC Codes

For the decoding of short LDPC codes, there is a significant performance gap between the SPA and ML decoding due to the existence of the small cycles [106]. Hence, several quasi-ML (QML) decoding methods [107–110] were proposed to approach the ML-decoding performance for the LDPC codes with short block lengths by introducing the reprocessing architecture. The general flow of the reprocessing can be summarized as follows:

- Perform the conventional BP decoding.
- If the BP decoding successes, output the decoded codeword. Otherwise, choose the candidate VNs according to a certain node selection methods [107–110].
- Create a list of all possible combinations of the selected VNs.
- Reconstruct the input sequences from the previous decoder output [107] or the channel outputs [108–110].
- Conduct multiple rounds of re-encoding [107] or decoding [108–110] tests.
- Determine the 'best' codeword according to certain decision metric.

Note that the idea of these QML decoding methods is intentionally correcting partial codeword for a received signal by enumerating all possible combinations of the selected VNs. For example, the OSD proposed in [107] only corrects the selected VNs with the most reliability according to the magnitude of their channel outputs. By performing sorting and matrix transformation, a list of codewords is obtained by encoding all error combinations into corresponding codewords. In the end, one of them is chosen as the decoding output, which has the minimum

Euclidean distance with the received signal. Apparently, there is an extremely high decoding complexity caused by the sorting, matrix transformation, and multiple rounds of the encoding process. To reduce the decoding complexity, the augmented BP (ABP) decoding and saturated min-sum (SMS) decoding were proposed in [108] and [109], respectively. Instead of re-encoding all error combinations, these QML decoding methods select the unreliable VNs based on the magnitude of their channel outputs. The conventional BP decoding tests are conducted for several rounds thereafter to generate a list of codewords, where the decoder input is modified from the initial channel output sequence at each decoding test. In addition, the QML decoder in [110] is different in the sense of selecting unreliable VNs, where the reliability of each unsatisfied CN is first calculated based on the LLRs. Starting from the unsatisfied CN with the largest LLR value, two least reliable VNs connected to the unsatisfied CNs with the smallest magnitude of their channel outputs are forced to flip their hard decision with a high priority. Nevertheless, for all these QML decoding methods with a small number of unreliable VNs being selected, the performance gap to the ML decoding is still considerable.

For the decoding of SC LDPC codes, an efficient decoding architecture was proposed in [93], so-called the sliding windowed decoder. Instead of performing the decoding on the entire codeword as that in FBD [94], a sliding windowed decoder uses a window with much smaller size and decoding on a portion of the codeword to achieve much less latency and memory demands. In the decoding window, an iterative message-passing decoding algorithm can be performed in an either FBD or non-uniform [96] way. The decoding window shifts to the next position if the stopping rule is satisfied or the preset number of iterations is reached. The symbols that are shifted out of the decoding window is called the target symbols. It is shown in [97] that the windowed decoding architecture introduces inherent performance degradation compared to the FBD. Most of the work in the literature [95–97] focused on improving the performance of the sliding windowed decoder with soft-decision decoding algorithms such as the SPA. In [88], the authors present the performance comparison of both binary and non-binary SC LDPC codes in terms of the computational complexity and decoding latency. The density evolution of SC LDPC codes under the sliding windowed decoder was proposed in [90], which is used for the construction of SC LDPC codes over GF(q) with good windowed decoding threshold.

1.3 Thesis Outline and Main Contributions

1.3.1 Thesis Organization

In this subsection, we present the outline of each chapter in this thesis. There are seven chapters in total, which covers the following aspects

- An overview of 5G frameworks and the development of channel coding technologies;
- Motivation of the research works conducted in this thesis;
- Background information on digital communication and channel coding;
- Basics and related works on LDPC codes.
- Details of the conducted research works.
- Conclusions of this thesis and future prospects.

Chapter 1

We provide an overview of 5G mobile networks and the development of channel coding technologies. Then, we state the motivation of this thesis and the related works in the literature. We also present the outline and the main contributions of this thesis.

Chapter 2

We provide background knowledge of digital communication and channel coding techniques. Definitions and basic concepts are also given. The materials presented in this chapter will be used throughout the rest of this thesis.

Chapter 3

The basics and related works on LDPC codes, including various decoding methods, commonly used analysis method, some classical construction methods of LDPC codes, are discussed in this chapter. These materials provide the fundamental concepts for the conducted research works in the following chapters.

Chapter 4

In Chapter 4, we first address the problem of how to construct a binary spatially-coupled (SC) LDPC code for storage applications with high error correction capability and very low uncorrectable error rate. Then we focus on how to employ a sliding windowed decoder with the conventional weighted bit-flipping algorithm to achieve a lower decoding complexity with slightly performance degradation compared to the full block decoding. Details of the construction methods, full descriptions of the proposed windowed decoding scheme, and relevant simulation results are presented.

Chapter 5

In Chapter 5, we introduce an enhanced quasi-maximum likelihood (EQML) decoding method for 5G LDPC codes. Detailed design of the decoding algorithm and the decoding architecture, the theoretical analysis of the error performance and decoding complexity are also presented in this chapter.

Chapter 6

In Chapter 6, we aim to achieve a desirable error performance for 5G LDPC codes by using the reprocessing architecture with a much lower decoding complexity. A novel approximate message passing (AMP)-aided decoding scheme is presented. Detailed explanation on how to formulate the decoding model with the AMP detector, the decoding procedures, as well as the analysis for some properties, and error performance of the proposed scheme are presented in this chapter.

Chapter 7

This chapter concludes the thesis by summarizing the main ideas of each chapter and the contributions of all the works conducted during my Ph.D. career.

1.3.2 Research Contributions

In the following, we present the detailed research contributions in chapters 4-6, respectively.

Chapter 4 presents a systematic design method of binary SC LDPC codes for storage applications. Most importantly, the proposed construction method is based on Euclidean geometry (EG) with high error correction capability and low uncorrectable bit error rate. In the proposed construction method, we adopt a two-dimensional edge-spreading process to generate a base matrix for SC LDPC codes, where a circulant decomposition method is used for the two-dimensional edge-spreading on a protograph of the constructed SC LDPC codes. Then the base matrix of the protograph is unwrapped and repeated periodically to construct the base matrix for a terminated SC LDPC code. The resulting base matrix is then lifted by the circulant permutation matrices to obtain the SC LDPC codes for various code lengths and code rates. In addition, we derive a lower bound on the rank of the parity-check matrix of our proposed EG-SC LDPC codes which is determined by the rank of the unwrapped parity-check matrix of the underlying EG LDPC code. We show that the error performance of the constructed EG-SC LDPC codes outperform their EG LDPC code counterparts, and show no error floor compared to the constructed protograph SC LDPC codes and regular LDPC codes.

We further propose a reliability-based windowed decoding scheme for SC LDPC codes to significantly reduce the error floor. In the proposed scheme, we propose a partial message reservation method which mitigates the error propagation by only reserving the reliable messages between two adjacent windows. We also introduce a partial syndrome check stopping rule to reduce the error floor under windowed decoding. The error performance of the proposed RBWD scheme is evaluated by simulation and it shows that the bit error rate of the RBWD scheme can approach that of full block decoding within 0.1 dB by using the low-complexity weighted bit-flipping decoding algorithm.

The results in Chapter 4 have been presented in the following publications:

- Y. Xie, L. Yang, **P. Kang**, and J. Yuan, "Euclidean geometry-based spatially coupled LDPC codes for storage," *IEEE J. Select. Areas Commun.*, vol. 34, pp. 2498–2509, Sep. 2016.

- **P. Kang**, Y. Xie, L. Yang, and J. Yuan, "Reliability-based windowed decoding for spatially coupled LDPC codes," *IEEE Commun. Lett.*, vol. 22, pp. 1322–1325, Jul. 2018.

In Chapter 5, a novel enhanced quasi-maximum likelihood decoding method for 5G LDPC Codes is presented. We first propose an improved MSA-based decoding algorithm with a low decoding complexity compared to the SPA but has the frame error performance near the SPA. A self-correction method is employed in the decoding algorithm, which reserves the reliable variable-to-check (V2C) messages and reduces the sign flips. Moreover, we apply one pair of universal scaling factors to the update of check node (CN) and variable node (VN) messages for all information bit lengths $K \in [40, 8448]$ and code rates $R \in [1/5, 8/9]$ in the 5G standard. In particular, the scaling factor used for VN update amplifies the magnitude of the extrinsic messages sending to their neighboring CNs which are connected to degree-1 VNs, and further improves the reliability of the reserved messages with the self-correction method.

We further propose an enhanced reprocessing architecture, so-called the enhanced quasi-maximum likelihood (EQML) decoding method, for 5G short LDPC codes to approach the frame error performance of the ML decoding with a desirable decoding complexity. By utilizing the sign flips for VNs' extrinsic messages, a novel node selection method from edge perspective is proposed in the reprocessing architecture to improve the accuracy of selecting unreliable VNs. We also introduce a stopping rule based on partial pruning to reduce the decoding complexity caused by multiple rounds of decoding tests in the reprocessing. The lower bounds on frame error rate (FER) of the proposed EQML decoding method by using a semi-analytical method. We investigate the FER performance of the proposed decoding algorithm and the EQML decoding method. Simulation results show that the proposed decoding algorithm can approach the error performance of the SPA for the 5G LDPC codes within 0.3 dB for all information bit lengths. Moreover, the EQML decoding method outperforms the SPA with the same decoding complexity for the 5G short LDPC codes and can approach the Polyanskiy-Poor-Verdú (PPV) bound within 0.4 dB in terms of FER performance.

And the derived lower bound on the FER is close to the simulation results in the low signal-to-noise ratio (SNR) region.

The results in Chapter 5 have been presented in the following publications:

- **P. Kang**, Y. Xie, L. Yang, C. Zheng, J. Yuan, and Y. Wei, "Enhanced quasi-maximum likelihood decoding for 5G LDPC codes," submitted to *IEEE Trans. Commun.* on 19th Jul. 2019.

- **P. Kang**, Y. Xie, L. Yang, C. Zheng, J. Yuan, and Y. Wei, "Enhanced quasi-maximum likelihood decoding of short LDPC codes based on saturation," in *Proc. IEEE Inf. Theory Workshop*, pp. 1–6, Aug. 2019.

- **P. Kang**, Y. Xie, L. Yang, C. Zheng, J. Yuan, and Y. Wei, "A method for improved decoding of low-density parity-check codes," Ref. No. 201811279697.5.

- **P. Kang**, Y. Xie, L. Yang, C. Zheng, J. Yuan, and Y. Wei, "A method for enhanced decoding of low-density parity-check codes," Ref. No. 201811279838.3.

- **P. Kang**, Y. Xie, and J. Yuan, "Improved Min-Sum Decoding of LDPC Codes," Project: Enhanced Decoding Algorithm for 5G LDPC Codes - Stage 1, Huawei Technology CO., LTD, Shanghai, China.

- **P. Kang**, Y. Xie, and J. Yuan, "Improved Quasi-Maximum Likelihood Decoder of LDPC Codes," Project: Enhanced Decoding Algorithm for 5G LDPC Codes - Stage 2, Huawei Technology CO., LTD, Shanghai, China.

In Chapter 6, a reprocessing scheme based on approximate message passing (AMP) [121] is presented to improve the performance of the 5G short LDPC codes with code rates $R \leq 0.5$. The decoding of the LDPC codes is formulated as a compressed sensing (CS) problem, where we use a sparse error vector to indicate the reliability of each VN in the codeword and reconstruct the error vector by the AMP algorithm. In the proposed decoding scheme, the AMP detector estimates the reliability of the channel output for each VN from the residue signal, where the residue signal is constructed by removing the decoded signal from the associated channel output sequence. Then the signs of the channel outputs on the unreliable VNs are flipped according to the preset threshold to generate the updated decoder input sequence. The decoder conducts a new decoding test with the modified input sequence and outputs the valid codeword if the decoding test successes. Moreover, an AMP-EQML decoding scheme is also presented for the 5G LDPC codes, where the AMP detector is used as a post-process operation for the unsuccessful decoding tests in the reprocessing of the EQML decoding. In this way, the probability of obtaining a valid codeword can be increased, which results in further improvement of the error performance of the EQML decoding. In addition, some properties of the proposed AMP-aided decoding scheme such as false flip rate (FFR), the denoiser success rate over the total number of decoding failure (DSRF) and the denoiser success rate over total transmissions (DSRT) are also analyzed and discussed. We show that the proposed AMP-aided decoding scheme achieves a 0.1 dB gain over the counterpart of one-time decoding for the 5G LDPC codes with various block lengths and low code rates. Furthermore, attributing to the AMP detector, the error performance of the AMP-aided decoding scheme with the conventional MSA can approach that of SPA within 0.3 dB for all simulated LDPC codes. It is also shown that the proposed AMP-EQML decoding scheme outperforms EQML decoding for information bit length of $K = 320$ and 752.

The results in Chapter 6 have been presented in the following publications:

- **P. Kang**, Y. Xie, Z. Sun, and J. Yuan, "The AMP-aided Decoding Scheme of 5G LDPC Codes," Project: Enhanced Decoding Algorithm for 5G LDPC Codes - Stage 3, Huawei Technology CO., LTD, Shanghai, China.

Chapter 2

Backgrounds and Preliminaries

It is well-known that channel coding is one of the key technologies in modern digital communication systems. In this chapter, we firstly introduce the fundamental mathematics and information theories of channel coding. This includes the communication system model, linear block code, decoding method, performance metric and channel capacities of coded systems. Note that we only do a preliminary introduction here without proofs. More details about the classic channel coding techniques can be found in [51, 54, 74, 122]. The introduction of modern channel coding techniques can also be found in the literature [52,69], and the knowledge of wireless digital communication can be referred to [123–126].

2.1 The System Model of Digital Communication

In 1948, Shannon proposed a general model of the digital communication system in his landmark paper [44], where he pointed out that the essential issue in the design of a communication system is *how* to transmit messages efficiently and

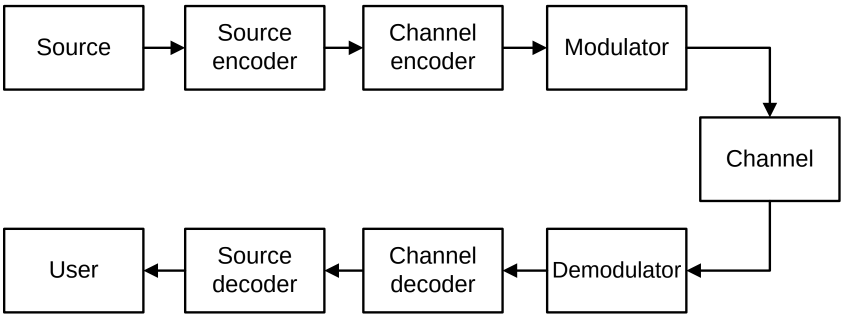


Figure 2.1: The block diagram of digital communication system with single carrier.

reliably, given the presence of noise. He further demonstrated that this goal can be achieved using coding techniques.

The details of Shannon's proposed model for a digital communication system are depicted in Fig. 2.1. This model is the basis behind various modern communication systems such as digital satellite TV, digital broadcasting, cellular network and Wi-Fi. The components of the digital communication system are described as follows.

- *Source* and *user* (or sink). The information source occurs in digital form (e.g., computer files), or digitized from an analog source (e.g., speech). We consider the output as a sequence of bits with a particular probability distribution function.

- *Source encoder* and *source decoder*. The encoder performs compression on the information source, wherein the bit sequence of the information source is converted into an alternative bit sequence with fewer bits. Hence, the efficiency of the information representation is enhanced. Depending on the source, the compression may be lossless (e.g., for computer data files) or lossy (e.g., for video, still images, and music). The source decoder performs the inverse function as the encoder, which recovers the bit sequence of the information source exactly (in the case of lossless compression), or approximately (in the case of lossy compression), from the output sequence of the source encoder.

- *Channel encoder* and *channel decoder*. To achieve reliable transmission, channel encoder introduces appropriately designed redundant bits to protect the information bits from various channel impairments, such as noise, distortion, and interference. Denoted by R $(0 < R < 1)$, the code rate is the ratio of the number of input bits at the channel encoder to the number of output bits. The function of the channel decoder is to recover the input of the channel encoder (i.e., the compressed sequence) from the channel output. Note that the designed goal of the channel encoder and decoder is to transmit the information bits at the highest possible rate while achieving a low probability of decoding errors under the channel with the existence of noise, distortion and interference.

- *Modulator* and *demodulator*. The modulator converts the output bit sequence from the channel encoder into a proper signal sequence $\mathbf{s}$ that is suitable for the channel. For example, for a wireless communication channel, the bit sequence must be represented by a signal with high-frequency to facilitate transmission with an antenna of reasonable size. In particular, the modulator can be divided into two types: the *baseband* modulator and the *passband* modulator. The baseband modulator converts the output bit sequence from the channel encoder into a baseband signal waveform by performing bit mapping followed by pulse shaping. As the inverse operator to the modulator, the demodulator recovers the modulator input sequence from the modulator output sequence. More specifically, the demodulator converts the received waveform into a discrete-time signal sequence (implemented by a matched filter). It then performs symbol-by-symbol or sequence detection and sends the decision result in the form of bit sequence or the calculated likelihood ratio information to the channel decoder.

- *Channel.* The channel is a physical medium that is capable of transmitting the output signal of a modulator. Physically, the channel includes transceiver antennas, amplifiers and filters that exist at both ends of the system. When the signal is transmitted over the channel, the channel may further introduce noise and interference in addition to the signal distortion. For research purpose, we adopt probabilistic models to evaluate the characteristics of the channel and obtain a variety of equivalent channel models.

Based on Shannon's model, a channel can be characterized by the channel capacity C, which is a measure of how much information the channel can transmit with an arbitrary small error probability. Shannon showed that we can achieve any reliable transmission (arbitrary small error probability) with appropriately designed codes as long as the code rate $R < C$. This thesis focuses on the low-density parity-check (LDPC) codes and their variations, particularly for the spatially-coupled LDPC codes and the LDPC codes used in 5G mobile communication systems.

2.1.1 Signal Constellation

The binary sequence from the output of the channel encoder is mapped to a signal constellation $\mathcal{A}$, which is defined in the following.

Definition 2.1. *(Signal Constellation): An N-dimensional signal constellation* *of size M is an N-dimensional vector set, such that*

$$\mathcal{A} = \left\{\mathbf{a}_m \in \mathbb{C}^N, 1 \leq m \leq M\right\}. \tag{2.1}$$

Note that each point in the signal constellation, which is also known as a signal point, corresponds to a different modulated waveform after pulse shaping, and all modulated waveforms have the same set of orthogonal basis. A set of N-dimensional signals of size M can transmit $\log_2 M$ bits for each N-dimension. For example, the signal constellation of quadrature phase-shift keying (QPSK) modulation is a set of 2-dimensional real vectors of size $M = 4$, and each 2-dimensional vector (complex signal) has $\log_2 M = 2$ bits. Therefore, the average energy of a signal constellation $\mathcal{A}$ (for each of the N dimensions) is given by

$$E_s = \mathbb{E}\{\|a_m\|^2\} = \sum_{m=1}^{M} \|a_m\|^2 \mathrm{P}(a_m), \tag{2.2}$$

where $\mathrm{P}(a_m)$ is the probability that the signal point a_m is transmitted. If the M signal points in $\mathcal{A}$ are transmitted with equal probability, we have

$$E_s = \frac{1}{M} \sum_{m=1}^{M} \|a_m\|^2. \tag{2.3}$$

2.1.2 Signal Mapping

The function of signal mapping is to convert the bit sequence $\mathbf{z}$ of length $\log_2 M$ into a certain signal point $s = \mathcal{C}(\mathbf{z})$ in the signal constellation $\mathcal{A}$. Note that $\mathcal{C}(\cdot)$ is the mapping function and the bit sequence $\mathbf{z}$ is called the label of the signal point. It is known that the signal mapping is not arbitrary, and usually needs to be optimized for desirable error performance. In general, the Gray mapping has a better error performance and is more commonly used in engineering practice (e.g., the 5G communication systems).

In this thesis, we consider binary phase-shift keying (BPSK) and QPSK modulations. For BPSK modulation, the bit z_m is mapped to a real signal s as follows:

$$s = 1 - 2z_m. \tag{2.4}$$

For QPSK modulation, the bits z_m and z_{m+1} are mapped to a complex signal s as

$$s = \frac{1}{\sqrt{2}}[(1 - 2z_m) + j(1 - 2z_{m+1})]. \tag{2.5}$$

2.2 Linear Block Codes

The most investigated and commonly used channel codes in practice are the codes defined on a finite field. Assume that the source can be represented by a sequence of consecutive binary symbols on the finite field GF(2), which is called the information sequence. The binary symbols in the information sequence is called the information bits. With respect to block codes, their information sequences are divided into multiple message sequences with fixed-lengths, where each message sequence contains K information bits. As a result, there are 2^K types of different message sequences in total. In the channel encoder, each input message sequence $\mathbf{u} = (u_1, u_2, \ldots, u_K)$ of K information bits is encoded as a length-N binary sequence $\mathbf{c} = (c_1, c_2, \ldots, c_N)$ according to certain coding rules, where $N > K$. The sequence $\mathbf{c}$ is called the codeword of the underlying message sequence $\mathbf{u}$ and the binary digits in the codeword are called code bits. Since there are 2^K different message sequences, resulting in 2^K corresponding codewords, the set of all codewords constitutes an $(N,\ K)$ block code. The parameters N and K are called the block length and the dimension of the code, respectively. The

$N - K$ bits are the redundant bits added by the encoder to each input message sequence. The ratio $R = K/N$ is called the code rate and can be interpreted as the average number of information bits carried by each code bit.

2.2.1 Definition

Definition 2.2. *(Binary Linear Block Codes): A binary (N, K) linear block code $\mathcal{C}$ is a K-dimensional subspace, containing 2^K codewords, of the vector space which is composed of all N-dimensional vectors over GF(2).*

In any linear code, the all-zero codeword is always one of its valid codewords since it represents the origin of the vector space.

2.2.2 The Generator Matrix and Parity-check Matrix

According to the definition of a binary $(N,\ K)$ linear block code $\mathcal{C}$, there are K linear independent codewords $\mathbf{g}_1, \mathbf{g}_1, \ldots, \mathbf{g}_K$, such that each codeword $\mathbf{c}$ in $\mathcal{C}$ is a linear combination of the K linear independent codewords, i.e.,

$$\mathbf{c} = u_1\mathbf{g}_1 + u_2\mathbf{g}_2 + \cdots + u_K\mathbf{g}_K, \tag{2.6}$$

where $u_i \in \mathrm{GF}(2)$.

We rearrange the K linearly independent codewords $\mathbf{g}_1, \mathbf{g}_1, \ldots, \mathbf{g}_K$ in $\mathcal{C}$ as the row vector of a $K \times N$ matrix over GF(2) as follows:

$$\mathbf{G} = \begin{bmatrix} \mathbf{g}_1 \\ \mathbf{g}_2 \\ \vdots \\ \mathbf{g}_K \end{bmatrix}. \tag{2.7}$$

Define $\mathbf{u} = (u_1, u_2, \ldots, u_K)$ as the message sequence to be encoded. Let $\mathbf{c} = (c_1, c_2, \ldots, c_N)$ be the corresponding codeword for $\mathbf{u}$, we have

$$\mathbf{c} = \mathbf{u} \cdot \mathbf{G}. \tag{2.8}$$

Therefore, the codeword $\mathbf{c}$ of the message sequence $\mathbf{u}$ is a linear combination of the row vectors of the matrix $\mathbf{G}$, where $\mathbf{G}$ is called the (N, K) generator matrix of the linear block code $\mathcal{C}$. Furthermore, the generator matrix $\mathbf{G}$ can be transformed into the systematic form:

$$\mathbf{G} = [\mathbf{I}_K|\mathbf{P}], \tag{2.9}$$

where $\mathbf{I}_K$ is a $K \times K$ identity matrix, and $\mathbf{P}$ is the matrix of size $K \times (N-K)$, $p_{i,j} \in$ GF(2). A codeword constructed from the generator matrix $\mathbf{G}$ in the systematic form is called the systematic code, where the most left K bits in the codeword $\mathbf{c}$ are identical with the message sequence $u_1, u_2, \ldots, u_K$ and the right most $N-K$ bits in the codewrod are the linear combination of the original information bits. These $N-K$ bits are called the parity bits. Additionally, an (N, K) linear block code $\mathcal{C}$ can also be defined by its parity-check matrix $\mathbf{H}$. The relationship between the generator matrix $\mathbf{G}$ and $\mathbf{H}$ is

$$\mathbf{G}\mathbf{H}^{\mathrm{T}} = \mathbf{0}. \tag{2.10}$$

Note that a binary N-dimensional vector $\mathbf{c} \in V$ is a codeword in $\mathcal{C}$ if and only if $\mathbf{c} \cdot \mathbf{H}^T$ is an $N-K$ dimension all-zero vector, such that

$$\mathcal{C} = \left\{\mathbf{c} \in V : \mathbf{c} \cdot \mathbf{H}^{\mathrm{T}} = \mathbf{0}\right\}, \tag{2.11}$$

where $\mathcal{C}$ is called the null space of $\mathbf{H}$. To consider the case of a systematic generator matrix $\mathbf{G}$ for an (N, K) linear block code, its corresponding parity-check matrix $\mathbf{H}$ in the systematic form can be represented as

$$\mathbf{H} = \left[\mathbf{P}^{\mathrm{T}}|\mathbf{I}_{N-K}\right]. \tag{2.12}$$

In summary, a linear block code can be uniquely determined by two matrices, namely the generator matrix and the parity-check matrix. In this thesis, LDPC codes are defined based on the parity-check matrix. Note that the parity-check matrix $\mathbf{H}$ of the (N, K) linear block code is said to be full rank if the number of rows in $\mathbf{H}$ is equal to the rank of $\mathbf{H}$, i.e., $\mathrm{rank}(\mathbf{H}) = N - K$. However, it is possible that $\mathbf{H}$ is not full rank, meaning that the number of rows in $\mathbf{H}$ is greater than $N - K$. In this case, some rows in the parity-check matrix $\mathbf{H}$ will be the linear combinations of the $N - K$ linearly independent rows in $\mathbf{H}$. These extra rows are called the redundant rows. We will introduce LDPC codes in the next chapter, of which the parity-check matrices are not necessarily full rank.

2.2.3 Shortening and Puncturing

Assuming $\mathcal{C}$ is a given (N, K) linear block code, we can simply modify it to get a new code. These methods include extending a code, puncturing, shortening, and expurgating, which are simple, but very practical and are often used in the design of coded systems. Here we only briefly introduce puncturing and shortening as these methods are most relevant to this thesis.

- Puncturing: This method reduces the code block lengths by deleting some parity bits of the codewords in $\mathcal{C}$, and thereby we can obtain a code with a higher code rate. This operation corresponds to deleting the associated columns of the generator matrix $\mathbf{G}$.

- Shortening: This method reduces the code length by removing some information bits of the codewords in $\mathcal{C}$, which corresponds to deleting the associated column of the parity-check matrix $\mathbf{H}$ or reducing the dimension of the generator matrix $\mathbf{G}$. Denote the number of shortened bits by ϕ, the code rate of the new code is $\frac{K-\phi}{N-\phi} < \frac{K}{N}$. Note that the information bits that are thrown away are generally with fixed positions in the codeword, such that the transmitter and the receiver both know the positions of the discarded data.

2.3 Decoding and Performance Measurements

In a communication system, the channel decoder determines the transmitted messages based on the received sequence, the encoding rules, and the noise characteristics of the channel. This operation is called decoding. Depending on the format of the messages from the demodulator, the decoding operation can be divided into hard-decision decoding and soft-decision decoding.

- *Hard decision*: Assume that a codeword $\mathbf{c}$ of a binary (N, K) linear block code is modulated by BPSK and transmitted over the additive white Gaussian noise (AWGN) channel. Note that the binary modulator has only two inputs ($M = 2$) and the channel output is a real number $y \in (-\infty, \infty)$. If the output of the demodulator adopts only two-level quantization, then the input values of the decoder will only have two values. In this case, we say that the demodulator adopts a hard decision, and the decoding based on the hard decision output of the demodulator is called hard-decision decoding.

- *Soft decision*: If the output of the demodulator uses more than two levels of quantization or does not perform quantization, the demodulator is said to adopt a soft decision. The decoding of the channel output based on this soft-decision is called soft-decision decoding.

Generally speaking, hard-decision has low complexity and is easier to implement than soft-decision, but soft-decision can provide better error performance. We show the channel capacity of hard-decision and soft-decision in Fig. 2.2. The Shannon limit of unconstrained-input AWGN channel is also demonstrated in the figure for comparison.

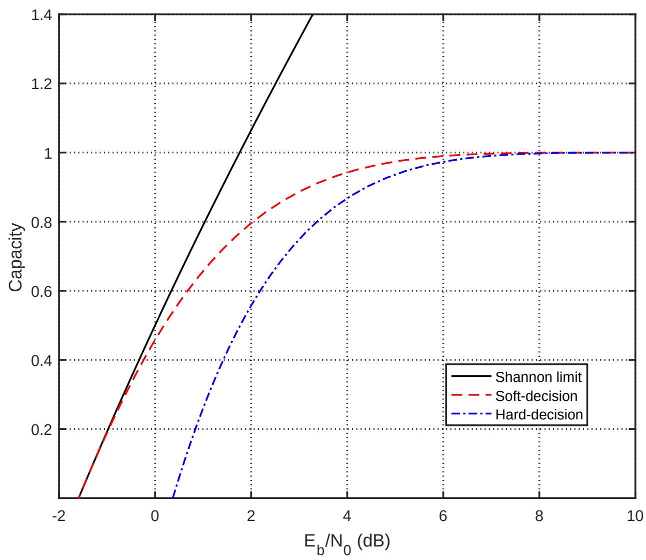


Figure 2.2: The BI-AWGN channel capacity for the soft-decision and hard-decision together with the channel capacity of unconstrained-input AWGN channel.

2.3.1 Error Detection of Linear Block Codes

Assume that an (N, K) linear block code $\mathcal{C}$ is with parity-check matrix $\mathbf{H}$ of size $(N-K) \times N$. Suppose $\mathbf{c}$ is the transmitted codeword and $\mathbf{z}$ is the hard decision sequence at the demodulator output. Note that $\mathbf{z}$ and $\mathbf{c}$ may differ due to the existence of channel noise and interference. Let $\mathbf{z} = \mathbf{c} + \mathbf{e}$, where $\mathbf{e}$ is called the error vector. To detect errors in the vector $\mathbf{z}$, we compute the $(N-K)$

dimensional vector over GF(2) as follows.

$$\mathbf{s} = \mathbf{z} \cdot \mathbf{H}^T = \mathbf{c} \cdot \mathbf{H}^T + \mathbf{e} \cdot \mathbf{H}^T = \mathbf{e} \cdot \mathbf{H}^T, \tag{2.13}$$

where $\mathbf{s} = (s_1, s_2, \ldots, s_{N-K})$ is called the syndrome of $\mathbf{z}$ and can be used to detect whether the received vector $\mathbf{z}$ contains transmission errors. More specifically, if $\mathbf{s} \neq 0$, this means that $\mathbf{z}$ is not a codeword in $\mathcal{C}$. Thus, we can detect the transmission error in $\mathbf{z}$. On the contrary, if $\mathbf{s} = 0$, then $\mathbf{z}$ can be determined as a codeword in $\mathcal{C}$ although $\mathbf{z}$ is not necessarily equal to the transmitted codeword $\mathbf{c}$. In this case, an undetectable error occurs.

2.3.2 The Optimal Decoding Rule

Let $\mathbf{c}$ and $\mathbf{y}$ be the transmitted codeword and the received vector, respectively. Define $\hat{\mathbf{c}}$ as the decoding output. The average decoding error probability is given by

$$P_e = \sum_{\mathbf{y}} \mathrm{P}(\hat{\mathbf{c}} \neq \mathbf{c}|\mathbf{y})\mathrm{P}(\mathbf{y}). \tag{2.14}$$

In digital communication, the decoding rule that minimizes the average decoding error probability is called the optimal decoding rule. Since minimizing $\mathrm{P}(\hat{\mathbf{c}} \neq \mathbf{c}|\mathbf{y})$ is equivalent to maximizing $\mathrm{P}(\hat{\mathbf{c}} = \mathbf{c}|\mathbf{y})$, then the *maximum a posteriori* (MAP) decoder performs as

$$\hat{\mathbf{c}} = \arg\max_{\mathbf{c} \in \mathcal{C}} \mathrm{P}(\mathbf{c}|\mathbf{y}). \tag{2.15}$$

For the MAP decoder, we select the codeword with the maximal conditional probability $\mathrm{P}(\mathbf{c}|\mathbf{y})$ among the codewords of $\mathcal{C}$ as the decoder output, where $\mathrm{P}(\mathbf{c}|\mathbf{y})$ is called the *a posteriori probability* (APP). According to the Bayes' rule, we have

$$\mathrm{P}(\mathbf{c}|\mathbf{y}) = \mathrm{P}(\mathbf{c})\mathrm{P}(\mathbf{y}|\mathbf{c})/\mathrm{P}(\mathbf{y}). \tag{2.16}$$

Note that $\mathrm{P}(\mathbf{c})$ is a constant if each codeword is transmitted with equal probability, in which case the MAP decoding is equivalent to the maximum-likelihood (ML) decoding, i.e.,

$$\hat{\mathbf{c}} = \arg\max_{\mathbf{c}\in\mathcal{C}} \mathrm{P}(\mathbf{y}|\mathbf{c}). \tag{2.17}$$

Note that the ML decoder always selects the codeword with the maximal conditional probability $\mathrm{P}(\mathbf{y}|\mathbf{c})$ for a given received vector $\mathbf{y}$. However, the MAP decoder can give the most likely codeword if the a priori information about $\mathbf{c}$ is available. It is also noticeable that both MAP and ML decoding are the decoding methods that can minimize the probability of codeword error.

In the following, we discuss how to apply the ML decoding for different channels. Suppose BPSK modulation is adopted, where the transmitted signal $\mathbf{x} = 1$, if $\mathbf{c} = 0$, and $\mathbf{x} = -1$, if $\mathbf{c} = 1$. We first consider the case of binary-input AWGN (BI-AWGN) channel with zero mean and noise variance σ^2. The conditional probability is given by

$$\mathrm{P}(\mathbf{y}|\mathbf{x}) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{\|\mathbf{y}-\mathbf{x}\|^2}{2\sigma^2}\right). \tag{2.18}$$

Define $d_E(\mathbf{y},\mathbf{x})$ as the Euclidean distance between the vector $\mathbf{y}$ and $\mathbf{x}$, such that

$$d_E(\mathbf{y},\mathbf{x}) = \|\mathbf{y}-\mathbf{x}\| = \sqrt{\sum_n (y_n - x_n)^2}. \tag{2.19}$$

The ML decoding is equivalent to minimize the Euclidean distance $d_E(\mathbf{y},\mathbf{x})$ as

$$\hat{\mathbf{c}} = \arg\min_{\mathbf{c}\in\mathcal{C}} \|\mathbf{y}-\mathbf{x}\|^2. \tag{2.20}$$

Next, we consider the decoding under binary symmetric channel (BSC), where only hard-decision can be adopted. Denote the hard-decision sequence of the received vector $\mathbf{y}$ by $\mathbf{z}$. With BPSK modulation, we have $d_E(\mathbf{y}, \mathbf{x}) = 2d_H(\mathbf{z}, \mathbf{c})$. Note that $d_H(\mathbf{z}, \mathbf{c})$ refers to the Hamming distance between the vector $\mathbf{z}$ and $\mathbf{c}$, and indicates the number of corresponding elements in vectors $\mathbf{z}$ that differs from each other. Therefore, the ML decoding here can be simplified as the minimum-distance decoding as

$$\hat{\mathbf{c}} = \arg\min_{\mathbf{c}\in\mathcal{C}} d_H(\mathbf{z}, \mathbf{c}). \tag{2.21}$$

To summarize, for the BSC, the ML decoding is to obtain the codeword $\mathbf{c}$ with the minimum Hamming distance from the received vector $\mathbf{y}$. While for the BI-AWGN channel, the ML decoding outputs the codeword with the minimum Euclidean distance from the received vector $\mathbf{y}$. This gives us guidance towards designing the codes applied for different channels. For example, the minimum Hamming distance between two codewords should be maximized when optimizing the code for the BSC. However, for BI-AWGN channel, we need to maximize the minimum Euclidean distance between two codewords.

2.3.3 Decoding in Log-likelihood Ratio Domain

As discussed previously, both the MAP and ML decoding are derived in the probability domain. However, in a practical communication system, the input of the soft decision decoder is usually represented by log-likelihood ratio (LLR) information for both numerical stability and simplified operation. Let $x \in \{+1, -1\}$

be a binary random variable, then its log-likelihood ratio (LLR) is defined as

$$L(x) \triangleq \ln\left(\frac{\mathrm{P}(x=+1)}{\mathrm{P}(x=-1)}\right), \tag{2.22}$$

where $\mathrm{P}(x = i)$ is the probability that the random variable x takes the value of i. Fig. 2.3 shows the relationship of the $L(x)$ value with different probability of $x = +1$ and $x = -1$, respectively. It can be seen that the sign of $L(x)$ gives a hard decision about x, and the magnitude of the LLR indicates the reliability of this decision.

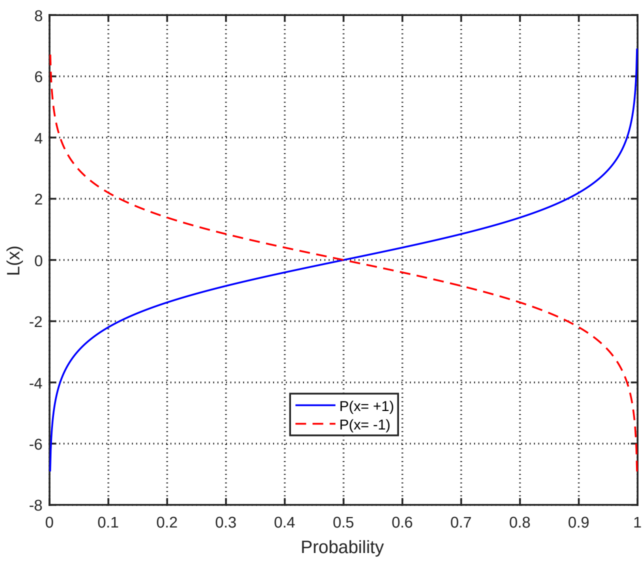


Figure 2.3: The log-likelihood ratio curve.

In terms of decoding, we also use the conditional LLRs. Assume that x_n is transmitted over the AWGN channel with BPSK signaling and the output of the demodulator is y_n. Then the conditional LLR is defined as

$$L(y_n|x_n) \triangleq \ln\left(\frac{P(y_n|x_n=+1)}{P(y_n|x_n=-1)}\right). \tag{2.23}$$

Since $\mathrm{P}(y_n|x_n)$ satisfies Gaussian distribution with variance σ^2, we have

$$L(y_n|x_n) = \frac{2y_n}{\sigma^2}, \tag{2.24}$$

This conditional LLR is also called the soft output of the channel.

2.3.4 Performance Evaluation of Coded Systems

Common Performance Metrics

For digital communication systems, the most commonly used performance metric is the bit error probability P_b. This is defined as the average probability that the decoder output bit $\hat{u}_i$ is not equal to the encoder input bit u_i [74]:

$$P_b \triangleq \frac{1}{K} \sum_{1 \leq i \leq K} \Pr(\hat{u}_i \neq u_i). \tag{2.25}$$

Note that P_b is called the bit error rate (BER) and we will continue to use this terminology in the remainder of this thesis.

An alternative performance metric for coded systems is the codeword error probability. It is defined as the probability that the decoder output decision $\hat{\mathbf{c}}$ is not equal to the output codeword $\mathbf{c}$ from the encoder. Denoted by P_e, the codeword error probability can be computed as

$$P_e \triangleq \Pr(\hat{\mathbf{c}} \neq \mathbf{c}). \tag{2.26}$$

Note that there are also different terminologies that refer to P_e, such as the word error rate and frame error rate (FER). For consistency, we use the terminology FER in the remainder of this thesis.

In addition to the decoding error probability (BER and FER), the coding gain can be also used to evaluate the performance of the decoder. Furthermore, coding gain is an important parameter to measure the power efficiency of a coded communication system. Let E_b/N_0 be the ratio of the average energy per information bit to the channel noise power spectral density. The coding gain is defined as the amount of reduction in E_b/N_0 by coded systems compared to uncoded systems to achieve a given target error probability (generally under the same modulation). Denoted by G_c, the coding gain (in dB) can be formulated as

$$G_c = \left[\frac{E_b}{N_0}\right]_{\text{uncoded}} - \left[\frac{E_b}{N_0}\right]_{\text{coded}}. \tag{2.27}$$

The Capacity of Coded Systems

Apart from the error probability (BER or FER) and the coding gain, the performance gap between the actual performance and the capacity is also an important metric of modern coded systems. Define the signal-to-noise ratio (SNR) as $\frac{E_s}{N_0} = \log_2 M \frac{R_c E_b}{N_0}$, where M is the size of the signal constellation and R_c is the code rate. The channel capacity of a discrete-time AWGN channel [74] is

$$C = \log_2(1 + \text{SNR}) = \log_2(1 + \log_2 M \frac{R_c E_b}{N_0}) \quad \text{b/2D}. \tag{2.28}$$

Note that the right side of Eq. (2.28) is called the Shannon limit, representing by the minimal SNR to ensure the error-free transmission for a given code rate. From the perspective of the performance analysis, we compare the computer simulation results to the Shannon limits of the corresponding code rate, and evaluate the performance gap.

However, in practice, the transmitted signal is often limited by a certain signal constellation $\mathcal{A}$, and follows a certain probability distribution such as a uniform distribution. In this case, the capacity will change according to a cer-

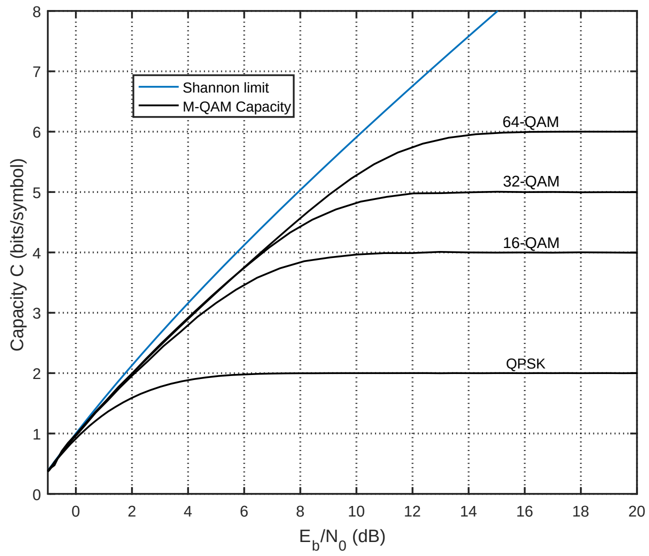


Figure 2.4: The capacity for different modulation over AWGN channels.

tain modulation in the coded systems, which is called the constrained-capacity. Correspondingly, the capacity without considering modulation is known as the unconstrained-capacity. In this thesis, only the capacity for discrete-time AWGN channels with two-dimensional signal constellation are considered. Assume that the AWGN channel model in discrete time follows $\mathbf{y} = \mathbf{x} + \mathbf{n}$, where $\mathbf{n}$ is a Gaussian white noise sequence with a mean of zero and variance σ^2 per dimension. Suppose that the set of channel input includes M complex number : $\mathcal{A} = \left\{\mathbf{a}_m \in \mathbb{C}^N, 1 \le m \le M\right\}$. Therefore, the probability density function of the channel output is

$$p(y|x) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{|y-x|^2}{2\sigma^2}\right). \tag{2.29}$$

Since $y = a_m + n$, we have

$$C^* = \log_2 M - \frac{1}{M}\sum_{m=1}^{M}\left\{\log_2 \sum_{a_{m'} \in \mathcal{A}} \exp\left(-\frac{|a_m + n - a_{m'}|^2 - |n|^2}{2\sigma^2}\right)\right\}. \tag{2.30}$$

In this way, we can calculate the capacity as a function of SNR by Eq. (2.30) from Monte-Carlo simulation. We demonstrate the constrained-capacity for the AWGN channel with different modulation in Fig. 2.4, where we assume that each symbol is transmitted with equal probability.

Performance in Finite Block Lengths

The capacity proposed by Shannon describes the performance of a coded systems with infinite code block length or the highest rate of the communication system to guarantee reliable transmission. However, in a practical communication system, there are finite code block lengths meaning that the Shannon capacity is not valid to evaluate the performance under the constraint of time delay. Therefore, the performance limits in finite block lengths is a more attractive benchmark for the coding design in practice. Gallager proposed an upper bound of the FER for random coding with fixed block lengths under ML decoding, so-called the random coding bound [127]. Moreover, the sphere packing bound is also given by Shannon, which describes a lower bound of the decoding error probability [128]. Nevertheless, these bounds become loose and inaccurate for finite code block lengths.

Recently, Polyanskyi, Poor and Verdú proposed the normal approximation formula [129] (PPV bound) for the error performance in finite-length, which provides a simple and efficient measurement for the channel codes with relatively long block lengths (code block lengths > 200). But the approximate result is still not accurate enough for the codes with very short block lengths. Later, Erseghe improved the work of Polyanskyi *et al.* by giving a compact integral expression in [130], which provides a better approximation for short codes. In the following, we introduce how to calculate the PPV bound for a short code under constrained AWGN channel by using the Monte Carlo method. It is known that the formula of the PPV bound is given by [129]

$$R^*\left(N^*, P_e\right) \approx C^* - \sqrt{\frac{V}{N^*}} Q^{-1}\left(P_e\right) + O(\log_2 N^*), \tag{2.31}$$

where the notations in the above equation are given as follows

- N^* is the length of the channel input sequence. For M-ary modulation, $N^* = N/\log_2 M$, where N is the code block lengths.

- C^* is the channel capacity.

- $R^* = R_c \cdot \log_2 M$ is the spectrum efficiency, where R_c is the code rate.

- V refers to channel dispersion, and can be computed by [74]

$$V = \mathrm{Var}\left\{\log \frac{p(Y|X)}{p(Y)}\right\}. \tag{2.32}$$

- $Q^{-1}(\cdot)$ is the inverse function of $Q(\cdot)$, i.e.,

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^{+\infty} \exp\left(-\frac{u^2}{2}\right) \mathrm{d}u. \tag{2.33}$$

- P_e refers to FER.

- $O(\log_2 N^*)$ is the correction term, which can be obtained by [129]

$$O(\log_2 N^*) = \frac{1}{2N} \log_2 N^*. \tag{2.34}$$

In particular, the channel dispersion can be estimated as follows

$$\mu \simeq -\frac{1}{L}\sum_{l=1}^{L}\log_2\frac{p(y_l|x_l)}{p(y_l)}, \tag{2.35}$$

$$V \simeq -\frac{1}{L}\sum_{l=1}^{L}\left[\log_2\frac{p(y_l|x_l)}{p(y_l)} - \mu\right]^2. \tag{2.36}$$

2.4 Summary

In this chapter, we present the background materials on coding theory, which provides an overview of the knowledge to describe the principles and construction methods for our research works in the following chapters. The main points presented in this chapter are summarized as follows.

- We introduce the basic concepts and the role of channel coding techniques in modern digital communication systems.
- We introduce the definitions and present some properties of the linear block codes, which are widely used in practical systems.
- We provide the basic knowledge of shortening, puncturing, and some definitions for the decoding of the linear block codes.
- We present some performance metrics and demonstrate various methods to evaluate the performance of linear block codes.

Chapter 3

Low-Density Parity-Check Codes

Low-Density Parity-Check (LDPC) codes are a class of linear block codes that can achieve near-capacity performance [52, 69, 74]. LDPC codes were firstly proposed by Gallager in his doctoral thesis in the early 1960s [70]. However, they have not been drawn much attention by researchers in the following 35 years until MacKay *et al.* rediscovered that LDPC codes have a near-capacity performance by using the sum-product algorithm (SPA) for decoding, and the decoding complexity grows linearly with the code block lengths [45]. Furthermore, LDPC codes can be represented by a sparse graph. Compared to Turbo codes [67, 68], which are a widely used in the third-generation (3G) and the fourth-generation (4G) communication systems, LDPC codes have been proven to have the following advantages [51]

- There is no need to adopt interleavers when applying LDPC codes in the communication systems, which reduces the complexity and latency of the communication systems.
- Better FER performance in the waterfall region, which satisfies the requirements of modern digital communications with high reliability.

In this chapter, we introduce the fundamentals of LDPC codes, including basic concepts, encoding/decoding, analytical tools, and several classic LDPC codes. This provides an overview and background knowledge of LDPC codes for the research works presented in the following chapters.

3.1 Definitions and Representation of LDPC Codes

An LDPC code defined over GF(q) with information bit length of K and code block length of N is a type of (N, K) linear block codes. In the following, we only consider binary LDPC codes, i.e., $q = 2$.

3.1.1 Matrix Representation

We can use the parity-check matrix $\mathbf{H}$ to represent an LDPC code. More specifically, the parity-check matrix $\mathbf{H}$ of an LDPC code is a sparse matrix of size $M \times N$, i.e., $\mathbf{H}$ contains majority of '0' elements and a relatively small number of '1' elements, where $M \geq N - K$. We call the number of '1' elements in each row and column of $\mathbf{H}$ as the row weight and the column weight, respectively. If the matrix $\mathbf{H}$ has a constant column weight d_v and a row weight d_c, the null space of the matrix $\mathbf{H}$ over GF(2) represents a (d_v, d_c) binary LDPC code. The code rate R_c is given by [74]

$$R_c = \frac{N - \mathrm{rank}\,(\mathbf{H})}{N}, \tag{3.1}$$

where $\mathrm{rank}\,(\mathbf{H})$ refers to the rank of the matrix $\mathbf{H}$. Since $\mathrm{rank}\,(\mathbf{H}) \leq M$, we have

$$\frac{N - M}{N} = 1 - \frac{d_v}{d_c} \leq R_c, \tag{3.2}$$

where $\frac{N-M}{N} = 1 - \frac{d_v}{d_c}$ is defined as the design rate of the LDPC code and the equality satisfies when $\mathbf{H}$ is full rank. Note that we call the LDPC codes with constant column and row weight as regular LDPC codes [74]. If the row weight or column weight is not fixed, then the LDPC code is an irregular LDPC code [74], which we would discuss later.

To obtain good performance by using iterative decoding, the parity-check matrix $\mathbf{H}$ of an LDPC code should have at most one non-zero element in common for any two rows (or two columns) [74]. This property is also called the row-column constraint (RC-constraint [74]).

In practice, people prefer structured LDPC codes, especially the LDPC codes with $\mathbf{H}$ in quasi-cyclic structure. Because this structure can significantly simplify the complexity of hardware implementation [74]. In the following, we introduce the definition of quasi-cyclic LDPC codes.

Definition 3.1. *If the parity-check matrix of an LDPC code consists of an array of circulants, then it is called a quasi-cyclic LDPC (QC-LDPC) code [74].*

The parity-check matrix $\mathbf{H}$ of a QC-LDPC code can be represented as:

$$\mathbf{H} = \begin{bmatrix} \mathbf{A}_{1,1} & \cdots & \mathbf{A}_{1,N} \\ \vdots & & \vdots \\ \mathbf{A}_{M,1} & \cdots & \mathbf{A}_{M,N} \end{bmatrix}, \tag{3.3}$$

where each matrix $\mathbf{A}_{i,j}$ is a $U \times U$ circulant. As an example, a 9×12 parity-check matrix of a QC-LDPC code is shown in Fig. 3.1, which is an array of 3×4 cyclic sub-matrices of size 3×3.

In addition, the structured LDPC codes also include LDPC codes with the parity-check matrix $\mathbf{H}$ in a cyclic, or a single/double diagonal structure. These structured codes can be constructed based on mathematical tools or computer

$$\boldsymbol{H} = \left[\begin{array}{ccc|ccc|ccc|ccc}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\
\hline
0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
\hline
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0
\end{array}\right]$$

Figure 3.1: A parity-check matrix of a QC-LDPC code.

search methods with the simplified design of the encoder and decoder, which is friendly for hardware implementation [74].

3.1.2 Graphical Representation

In 1981, Tanner proposed the Tanner graph in [104], which can be used to represent the parity-check matrix of LDPC codes. As a bipartite graph, the Tanner graph consists of one set of variable nodes (VNs) and one set of check nodes (CNs). Denote the Tanner graph of an LDPC code by $\mathcal{H}$ and assume the size of $\mathbf{H}$ is $M \times N$. A VN $v_n, 1 \leq n \leq N$, is connected to a CN $c_m, 1 \leq m \leq M$, by an edge in the Tanner graph if there is a nonzero element in the m-th row and n-th column of $\mathcal{H}$. If the Tanner graph is used to represent a (d_v, d_c)-regular LDPC code, then the total number of the edges is equal to $Md_c = Nd_v$.

An example of a length-12 (3,4)-regular LDPC code repesented by the Tanner graph is shown in Fig. 3.2, where the corresponding parity-check matrix $\mathbf{H}$ is

$$\mathbf{H} = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}. \tag{3.4}$$

Let $\mathbf{c}$ be a codeword of the above LDPC code. For each CN, the summation of

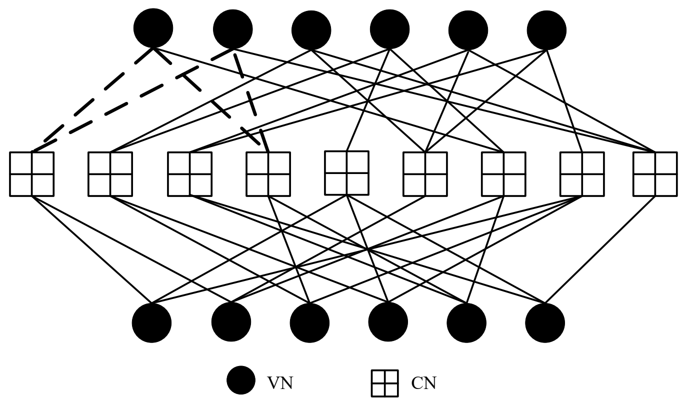


Figure 3.2: The Tanner graph of a length-12 (3,4)-regular LDPC code.

its neighboring VNs is equal to zero, according to Eq. (2.13). Note that there are four dash lines in the Tanner graph shown in Fig. 3.2, which results in a closed path.

Definition 3.2. *(The Cycle and Girth): A closed path in a Tanner graph $\mathcal{G}$ is called a cycle, which begins and ends at the same CN or VN. The number of edges on the cycle is called the length of the cycle. The length of the shortest cycle is* *defined as the girth of the Tanner graph. The number of the cycles with different lengths in $\mathcal{G}$ is called the distribution of cycles [74].*

As shown in [74], the girth and the number of short cycles in the Tanner graph can directly affect the performance of an LDPC code under iterative decoding algorithm since the errors can propagate between the nodes within a cycle.

3.1.3 Polynomial Representation

For irregular LDPC codes, the column weight and row weight of the parity-check matrix of an LDPC code vary with the columns and rows. Therefore, it is more convenient to represent an irregular LDPC code by specifying the degree distribution of the VNs and the CNs. Let $\rho(x)$ and $\lambda(x)$ be the degree distribution of the VNs and CNs for an irregular LDPC code, respectively. The polynomial representation can be given by [74]

$$\lambda(x) = \sum_{d=1}^{d_v} \lambda_d x^{d-1}, \tag{3.5}$$

$$\rho(x) = \sum_{d=1}^{d_c} \rho_d x^{d-1}, \tag{3.6}$$

where λ_d denotes the fraction of the edges connecting to degree-d VNs, and ρ_d edge denotes the fraction of the edges connected to degree-d CNs. If the irregular LDPC code has the parity-check matrix of size $M \times N$, then its code rate R_c is bounded by [74]

$$R_c \geq 1 - \frac{M}{N} = 1 - \frac{\int_0^1 \rho(x)dx}{\int_0^1 \lambda(x)dx}. \tag{3.7}$$

As shown in [131], irregular LDPC codes have a better waterfall performance closer to the capacity than regular LDPC codes under the same block lengths comparison.

3.2 LDPC Code Examples

Nowadays, LDPC codes can be generally divided into two categories: random or pseudo-random LDPC codes and structured LDPC codes. It is noticeable that the structured LDPC codes usually have their parity-check matrices with a cyclic or quasi-cyclic structure, which is very convenient for hardware implementation of the decoder. Therefore, we focus on structured LDPC codes in this section. In the following, we would introduce some classic LDPC codes in details.

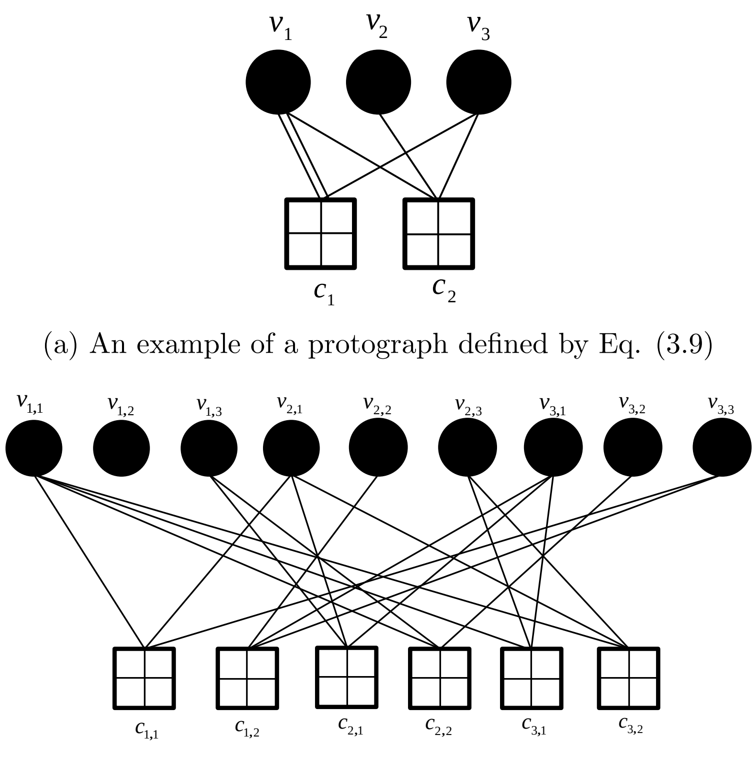
3.2.1 The Protograph-based LDPC Codes

An efficient technique to design LDPC codes is generating a large Tanner graph from a protograph, which is called the protograph-based construction [113]. It is known that the protograph is a small bipartite graph from which we can generate a larger Tanner graph by duplication and edge permutation [113]. To be more specific, we first select the code rate R and the number of VNs in the protograph to obtain the degree distribution of the LDPC code ensemble to be constructed. To construct a practical code from a protograph ensemble, the process of graph lifting is adopted to derive a large Tanner graph from the protograph of the ensemble. The constructed code from the derived Tanner graph is quasi-cyclic if the lifting matrix is a circulant permutation matrix (CPM). Denote by

$$\mathbf{P} := \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \ddots & \ddots & \ddots & 1 \\ 1 & 0 & \cdots & 0 & 0 \end{bmatrix} \tag{3.8}$$

a $U \times U$ CMP such that $\mathbf{P}^U = \mathbf{P}^0 = \mathbf{I}$, where $\mathbf{I}$ is the identity matrix. Note that $\mathbf{P}$ is of weight one since it has only one non-zero element in each row and column. Let $f(x) = \sum_{i=1}^{l} x^{r_i}$ be a univariate polynomial of l distinct terms such that $0 \leq r_1 < r_2 < \ldots < r_l < U$. We define a weight-$l$ CPM as $f(\mathbf{P}) := \sum_{i=1}^{l} \mathbf{P}^{r_i}$. We call $f(x)$ the generator polynomial of the weight-l circulant and it is represented as the index vector $\mathbf{f} = [r_1, r_2, \ldots r_l]$, which consists of only the exponents of $f(x)$. The order U of a CPM is known as the lifting factor, and we call $\mathbf{f}$ the generator vector of a circulant hereafter.

Fig. 3.3a demonstrates a protograph and its lifted Tanner graph is shown in Fig. 3.3b. It is worth mention that parallel edges are allowed in the protograph, while the Tanner graph is required to have an only single edge after the lifting operation.



(a) An example of a protograph defined by Eq. (3.9)

(b) The lifted Tanner graph

Figure 3.3: An example of lifting operation.

By considering the matrix representation of the protograpgh in 3.3a, we can use a base matrix, where

$$B = \begin{bmatrix} 2 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix}, \tag{3.9}$$

where the element "2" refers to the parallel edge between node c_1 and v_1, and "0" indicates that there exists no edge between node c_1 and v_2. When $U = 3$, the permutation matrices of the Tanner graph shown in Fig. 3.3b are given by

$$\mathbf{P}_{1,1} = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}, \mathbf{P}_{1,2} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \mathbf{P}_{1,3} = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix},$$

$$\mathbf{P}_{2,1} = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \mathbf{P}_{2,2} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}, \mathbf{P}_{2,3} = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}.$$

Based on the above permutation matrices, we can obtain the parity-check matrix as

$$\mathbf{H} = \begin{bmatrix} \mathbf{P}_{1,1} & \mathbf{P}_{1,2} & \mathbf{P}_{1,3} \\ \mathbf{P}_{2,1} & \mathbf{P}_{2,2} & \mathbf{P}_{2,3} \end{bmatrix} \tag{3.10}$$

We can see in the above example, the permutation matrices are selected as the CPMs, which results in a quasic-cyclic structure.

3.2.2 The Euclidean Geometry LDPC Codes

Let $\mathrm{EG}(m, 2^s)$ be an m-dimensional Euclidean geometry over the Galois Field $\mathrm{GF}(2^s)$ where $m, s \in \mathbb{Z}^+$ are positive integers. Denote by $\mathbf{H}_{(m,2^s)}$ an $N \times M$

parity-check matrix over GF(2) composed of $M = \left(2^{(m-1)s} - 1\right)(2^{ms} - 1) / (2^s - 1)$ lines in EG$(m, 2^s)$ that pass through $N = 2^{ms} - 1$ non-origin points. If $m > 2$, the M columns of the parity-check matrix $\mathbf{H}_{(m,2^s)}$ can be partitioned into $K = \left(2^{(m-1)s} - 1\right) / (2^s - 1)$ cycle classes each of which consists of $(2^{ms} - 1)$ lines [74] [46] [132], and each line is a generator vector of the cycle class. Alternatively, if $m = 2$, the special class of two-dimensional EG codes is characterized by a single cycle class, i.e., $K = 1$. Let $\mathbf{H}^{(j)}_{(m,2^s)}$ be a $(2^{ms}-1)\times(2^{ms}-1)$ square matrix with the first column being a generator vector of a cycle class. By cyclic shifting the generator vector downwards, each cycle class can be arranged into the form of weight-2^s circulant. Hence, the parity-check matrix $\mathbf{H}_{(m,2^s)} = \left[\mathbf{H}^{(j)}_{(m,2^s)}\right]_{0\leq j<K}$ is made of K weight-2^s circulants juxtaposed side-by-side.

3.2.3 The LDPC Codes in the 5G Standard

Recently, protograph-based raptor-like (PBRL) LDPC codes [133] are designed and used for the eMBB scenario in the fifth generation (5G) mobile networks due to their low implementation complexity and satisfactory error rate performance [1, 134]. Let $\mathbf{H}$ be a parity-check matrix of size $M \times N$ for a 5G LDPC code, where M and N are the number of rows and columns in $\mathbf{H}$, respectively. Define $\mathbf{H}_{\text{core}}$ as the core matrix with the highest rate that the parity-check matrix $\mathbf{H}$ can have. Denoted by $\mathbf{H}_{\text{ex}}$ the sub-matrix indicating the connection in the single parity check (SPC) extension of the 5G LDPC code from the high rate to low rate. The general structure of the parity-check matrix H of the 5G LDPC codes can be represented as

$$\mathbf{H} = \begin{bmatrix} \mathbf{H}_{\text{core}} & \mathbf{0} \\ \mathbf{H}_{\text{ex}} & \mathbf{I} \end{bmatrix}, \tag{3.11}$$

where $\mathbf{0}$ and $\mathbf{I}$ refer to the zero and the identity matrices. Note that the parity-check matrix $\mathbf{H}$ here is lifted from an underlying base matrix by corresponding CPMs, which also results in a quasi-cyclic structure. Fig. 3.4 shows the parity-check matrix $\mathbf{H}$ of 5G LDPC codes, where the submatrices $\mathbf{C}$ and $\mathbf{E}$ consists of CPMs and all-zero matrices, and the submatrix $\mathbf{D}$ corresponds to the parity-check bits with dual-diagonal structure. This structure of the parity-check matrix $\mathbf{H}$ is

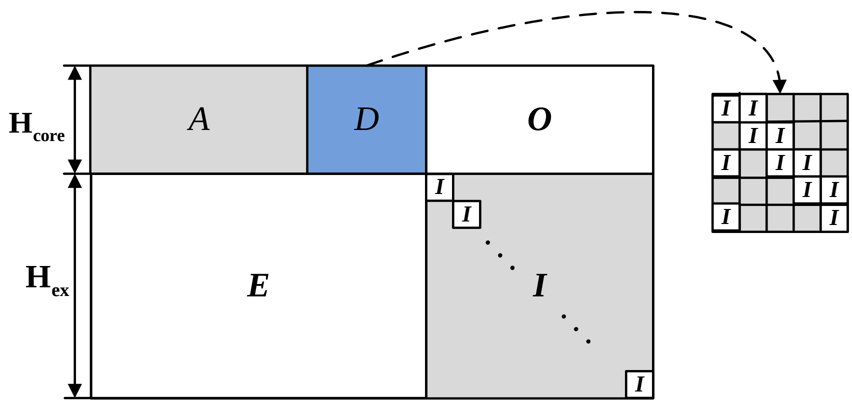


Figure 3.4: The parity-check matrix structure of 5G LDPC code.

equivalent to a high-rate LDPC code serially concatenated with multiple SPC codes, and a parity-check matrix of an LDPC code with an arbitrarily low code rate can be obtained as the number of rows and columns in $\mathbf{H}_{\text{ex}}$ increases.

To obtain the Tanner graph of a 5G LDPC code, we follow a similar procedure as described in Section 3.2.1. It is known that the 5G LDPC codes obtain the different size of the parity-check matrices by extending the associated base graphs (BGs) with different lifting factor U to achieve the flexibility of various information bit lengths. Therefore, the lifting factor U is the first key parameter to choose for the construction of the parity-check matrix. To select a proper value of U, we need to determine the frame structure according to [1]. Define K as the number of information bits and K_b is the number of columns in the base matrix used by the information bits. There are two different BGs in the 5G standard, namely BG1 and BG2. According to [1], BG2 is used if the $K \leq 3840$, and the code rate $R \leq 2/3$. Otherwise, BG1 is used. The value of K_b is selected based on the following rules [1]:

For BG1:

$K_b = 22.$

For BG2:

If $K > 640$

$K_b = 10,$

else if $K > 560$

$K_b = 9,$

else if $K > 192$

$K_b = 8,$

else

$K_b = 6.$

end if

For a given K_b, the minimum value of lifting factor, denoted by U_c, can be selected from the sets of lifting factors shown in Table 3.1, such that $K_b \cdot Z_c \geq K$, where a is known as the shift coefficient.

Table 3.1: Set of shift coefficients and lifting factors

Set No.	Shift coefficient a	Lifting sizes $U = a \times 2^j$, j=0,1,2,3,4,5,6,7
Set 1	2	{2, 4, 8, 16, 32, 64, 128, 256}
Set 2	3	{3, 6, 12, 24, 48, 96, 192, 384}
Set 3	5	{5, 10, 20, 40, 80, 160, 320}
Set 4	7	{7, 14, 28, 56, 112, 224}
Set 5	9	{9, 18, 36, 72, 144, 288}
Set 6	11	{11, 22, 44, 88, 176, 352}
Set 7	13	{13, 26, 52, 104, 208}
Set 8	15	{15, 30, 60, 120, 240}

With the chosen U_c, the cyclic shift value $\mathbf{A}_{ij}$ can be calculated by

$$\mathbf{A}_{i,j} = \begin{cases} -1, & \text{if } \mathbf{P}_{i,j} = -1 \\ \mathrm{mod}(\mathbf{P}_{i,j}, U_c), & \text{otherwise} \end{cases} \tag{3.12}$$

where $\mathbf{P}_{i,j}$ is the shift coefficient of the (i, j)-th element in the corresponding base matrix, and an all-zero CPM is used if $\mathbf{P}_{i,j} = -1$. Then the Tanner graph of the 5G LDPC code can be obtained by replacing each block in the base matrix by a $U \times U$ CPM with corresponding shift coefficient $\mathbf{P}_{i,j}$.

To obtain a better decoding threshold [1], the information bits in the first two circulant column blocks of the parity-check matrix are always punctured. Furthermore, to achieve rate adaptation with different information bit lengths K, the last $\Delta K = K_{b_{\max}} \cdot U_c - K$ information bits need to be shortened by zero padding, where $K_{b_{\max}} = 10$ for BG2, and $K_{b_{\max}} = 22$ for BG1. Additionally, the last $(n_{\mathcal{V}} - 2)U_c - K/R - \Delta K$ parity bits in $\mathbf{H}$ are also punctured for rate compatibility, where $n_{\mathcal{V}}$ is the number of columns in the base matrix. Fig. 3.5 illustrates the corresponding bit positions of the shortened and punctured bits in the parity-check matrix. Note that the shortened bits are considered as the known information, for which the channel ouput LLRs are initialized as the LLR

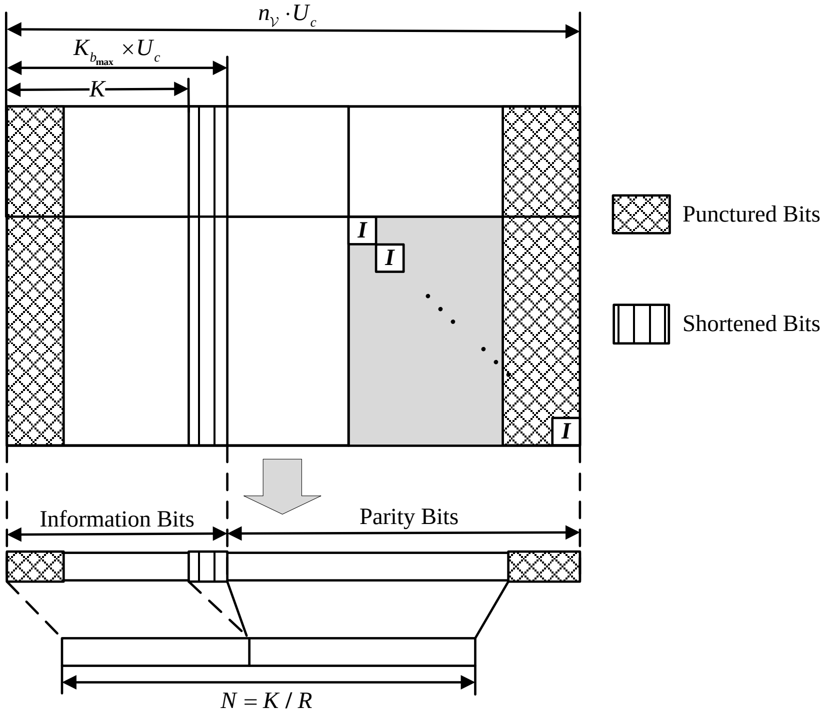


Figure 3.5: The bit positions of the shortened and punctured bits [1].

values of bit 0. However, the channel output LLRs on punctured bits are treated as unknown with initialized LLR=0, and these punctured bits are expected to be recovered at the end of the decoding process.

3.2.4 The Spatially-Coupled (SC) LDPC Codes

SC LDPC codes can be viewed as a type of LDPC convolutional codes (LDPC-CC) [86] that have the ability to combine good features of both regular and irregular LDPC codes in a single code design. To construct SC LDPC codes from LDPC block codes, the well-known approach called matrix unwrapping [86] is commonly adopted. Let $\mathbf{H}$ be the parity-check matrix of size $r \times c$ for a binary LDPC block code. Its code rate is given by $R_{BC} = 1 - M/N$. A practical SC LDPC code, commonly known as the terminated LDPC-CC, can be represented by a parity-check matrix [85]

$$\mathbf{H}^{SC} = \begin{bmatrix} \overbrace{\begin{matrix} \mathbf{H}_0 & & & \\ \mathbf{H}_1 & \mathbf{H}_0 & & \\ \vdots & \mathbf{H}_1 & & \\ \mathbf{H}_{m_s-1} & \vdots & \ddots & \\ \mathbf{H}_{m_s} & \mathbf{H}_{m_s-1} & \ddots & \mathbf{H}_0 \\ & \mathbf{H}_{m_s} & \ddots & \mathbf{H}_1 \\ & & & \vdots \\ & & & \mathbf{H}_{m_s-1} \\ & & & \mathbf{H}_{m_s} \end{matrix}}^{L} \end{bmatrix}, \tag{3.13}$$

where m_s is called the syndrome former memory. Each $\mathbf{H}_j$ of size $r \times c$, such that $\sum_{j=0}^{m_s} \mathbf{H}_j = \mathbf{H}$, is a descendent matrix of the parity-check matrix $\mathbf{H}$. The set of descendent matrices is then repeated L times as shown in (3.13) to construct the parity-check matrix $\mathbf{H}^{SC}$ of the terminated SC LDPC code, where L is called the termination length of the code. Note that the process of termination results in a parity-check matrix $\mathbf{H}^{SC}$ that contains irregular row weights. The code rate of an SC LDPC code is then a function of L, given by [85]

$$R_{SC} = 1 - \frac{(L + m_s)r}{Lc}. \tag{3.14}$$

Obviously, as the termination length $L \to \infty$, the SC LDPC code has the same code rate as the underlying LDPC block code defined by $\mathbf{H}$, that is $R_{SC} \xrightarrow{L\to\infty} R_{BC}$. Note that both R_{SC} and R_{BC} are known as the design code rate, while the true code rate R is not smaller than the design rate.

The Construction of SC LDPC Codes

An insightful way of designing terminated SC LDPC codes is to use a protograph representation of a code ensemble. Define a $\mathscr{P} = (\mathcal{V}, \mathcal{C}, \mathcal{E})$ protograph that connects a set of $n_{\mathcal{V}}$ VNs $\mathcal{V} = \{v_1, v_1, \dots, v_{n_{\mathcal{V}}}\}$ to a set of $n_{\mathcal{C}}$ CNs $\mathcal{C} = \{c_1, c_2, \dots, c_{n_{\mathcal{C}}}\}$ by a set of edges $\mathcal{E}$ [113]. Assume $n_{\mathcal{C}} > n_{\mathcal{V}}$, the positive design rate of this protograph ensemble block codes is $R_{BC} = 1 - n_{\mathcal{C}}/n_{\mathcal{V}}$, which is a lower bound on the code rate of each member of the ensemble. Let $\mathbf{B}$ be the equivalent base matrix of the protograph $\mathscr{P}$. Then the protograph $\mathscr{P}^{SC}$ of a $(n_{\mathcal{V}}, n_{\mathcal{C}}, L)$ ensemble of SC LDPC code is obtained by performing edge-spreading [89] to split the base matrix $\mathbf{B}$ into $m_s + 1$ descendent base matrices $\mathbf{B}^{(0)}, \mathbf{B}^{(1)}, \dots, \mathbf{B}^{(m_s)}$. Each of the descendent base matrices has size $b_r \times b_c$ and $\sum_{i=0}^{m_s} \mathbf{B}^{(i)} = \mathbf{B}$. By arranging the set of descendent base matrices into the similar form as shown in (3.15), the base matrix of an SC LDPC code can be represented as [89]

$$\mathbf{B}_L = \begin{array}{c} \left[\begin{array}{cccc} \multicolumn{4}{c}{\overbrace{\hspace{9em}}^{L}} \\ \mathbf{B}_0 & & & \\ \vdots & \mathbf{B}_0 & & \\ \mathbf{B}_{m_s-1} & \vdots & \ddots & \mathbf{B}_0 \\ \mathbf{B}_{m_s} & \mathbf{B}_{m_s-1} & \ddots & \vdots \\ & \mathbf{B}_{m_s} & \ddots & \mathbf{B}_{m_s-1} \\ & & & \mathbf{B}_{m_s} \end{array} \right] \\ \begin{array}{cccc} 1 & 2 & \dots & L \end{array} \end{array}, \tag{3.15}$$

Note that if all descendent base matrices $\mathbf{B}^{(i)}$ are identical, the resulting terminated base matrix of an SC LDPC code is time-invariant. Otherwise, it is time-varying, where each row of (3.15) could start with a different $\mathbf{B}^{(i)}$. The lifting operation can be then applied to the unwrapped base matrix $\mathbf{B}$ to obtain the Tanner graph of the SC LDPC code [89].

Since the derived Tanner graph of a block protograph LDPC code has $Un_{\mathcal{C}}$ check nodes and $Un_{\mathcal{V}}$ variable nodes, a protograph based SC LDPC code lifted from $\mathbf{B}$ using CMPs has $(m_s + L)\,Un_{\mathcal{C}}$ CNs and $LUn_{\mathcal{V}}$ VNs. Note that the design code rate of the protograph based SC LDPC codes is given by $R_{SC} = 1 - (L + m_s)b_r/Lb_c$.

The Sliding Window Decoder for SC LDPC Codes

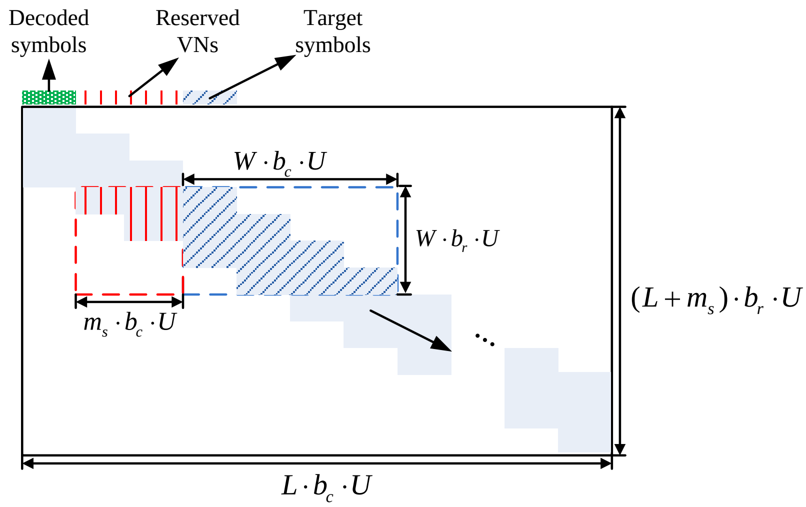


Figure 3.6: Illustration of sliding window decoder for an SC LDPC code. Note that the symbols marked in green over the parity-check matrix have all been decoded. The blue region over the parity-check matrix represents target symbols and the symbols in gray over the parity-check matrix are yet to be decoded.

In [93], a sliding window decoder was proposed for SC LDPC Codes. Instead of performing full block decoding (FBD) over the whole base matrix $\mathbf{B}_L$, the sliding window decoder uses a window of size W covering $W \cdot Ub_r$ CNs and $W \cdot Ub_c$ VNs. The decoding window slides from time index $t = 1$ to time index $t = L$ which associates with different window positions in $\mathcal{B}_L$. In a decoding window, an iterative message-passing decoding algorithm is performed between all VNs and CNs. The decoding process stops if a valid codeword is found or a predetermined maximum number of iterations is reached. Then the decoding window shifts by Ub_r CNs vertically and Ub_c VNs horizontally where the Ub_c VNs shifted out of the decoding window are called target symbols.

Fig. 3.6 shows the general decoding architecture of sliding window decoder operating on the parity-check matrix of an SC-LDPC code. It is important to mention that updating the CNs within current decoding window requires the extrinsic information based on the last update of VNs in the previous m_s protograph blocks (the red region in Fig. 3.6). Moreover, the sliding window decoder is assumed to be restricted by the boundaries of protograph if it moves beyond the protograph.

3.3 The Decoding of LDPC Codes

In terms of decoding the LDPC codes, Gallager originally proposed two decoding algorithms in [70]. One of them is based on the soft decision, called the sum-product algorithm, and another is based on the hard decision, which is called the bit-flipping algorithm. After the rediscovery of LDPC codes, many researchers put their efforts into designing the decoding algorithms and architectures with improved error performance and low complexity. In this section, we first review some of the decoding algorithms for LDPC codes. Then the improvement of the decoding architectures is also discussed.

3.3.1 The Iterative Decoding Algorithms

According to our research works in the following chapters, we mainly introduce the classic iterative decoding algorithms for binary LDPC codes, which are summarized as follows.

- Sum-product algorithm (SPA) [135].
- Min-sum algorithm (MSA) [47].
- The variations of MSA [47–50]
- Weighted-BF (WBF) algorithm [74].

Assume that the codeword $\mathbf{x} = (x_1, x_2, \ldots x_N)$ is transmitted through a BI-AWGN channel and the decoder obtains the received signal $\mathbf{y} = (y_1, y_2, \ldots y_N)$ at the output of the receiver match filter. Denoted by $L(v_n)$ the initial LLR for a node v_n, we have

$$L(v_n) \triangleq \log \left(\frac{\Pr(y_n|x_n = 0)}{\Pr(y_n|x_n = 1)} \right). \tag{3.16}$$

Let $\mathcal{H}(c_m)$ and $\mathcal{H}(v_n)$ denote the set of all VNs connected to c_m and the set of all CNs connected to v_n, respectively. We also define $\mathcal{H}(c_m)\backslash v_n$ as the set $\mathcal{H}(c_m)$ with v_n excluded, and $\mathcal{H}(v_n)\backslash c_m$ as the set $\mathcal{H}(v_n)$ with c_m excluded. We first introduce the SPA and then discuss its simplified decoding algorithms.

The Sum-Product Algorithm

The SPA is a message passing algorithm that can be visualized with a factor graph [135]. Since there always exist cycles in the Tannr graph of a practical LDPC code, the messages computed by the SPA becomes an approximation on the graph with cycles and is sub-optimal for decoding [52]. However, generally speaking, the SPA can still achieve a desirable error performance.

At the l-th iteration, the SPA computes the check-to-variable (C2V) message from c_m to v_n as [135]

$$Y_{mn}^{(l)} = 2\tanh^{-1} \left(\prod_{v_{n'} \in \mathcal{H}(c_m)\backslash v_n} \tanh \left(\frac{Z_{n'm}^{(l-1)}}{2} \right) \right), \tag{3.17}$$

where $1 \leq m \leq M, v_n \in \mathcal{H}(c_m)$, and $Z_{nm}^{(0)} = L(v_n)$. The variable-to-check (V2C) messages from v_n to c_m are computed by [135]

$$Z_{nm}^{(l)} = L(v_n) + \sum_{c_{m'} \in \mathcal{H}(v_n)\backslash c_m} Y_{m'n}^{(l)}, \tag{3.18}$$

where $1 \leq n \leq N$ and $c_m \in \mathcal{H}(v_n)$. Meanwhile, a posterior LLR of the node v_n at each iteration is given by [135]

$$L_n^{(l)} = L(v_n) + \sum_{c_m \in \mathcal{H}(v_n)} Y_{mn}^{(l)}. \tag{3.19}$$

Define $\hat{\mathbf{x}}^{(l)} = \left(\hat{x}_1^{(l)}, \hat{x}_2^{(l)}, \ldots \hat{x}_N^{(l)}\right)$ as the temporary codeword at the l-th iteration. The hard decision of $\hat{x}_n^{(l)}$ is made based on $L_n^{(l)}$ such that $\hat{x}_n^{(l)} = 1$ if $L_n^{(l)} < 0$, and $\hat{x}_n^{(l)} = 0$ if $L_n^{(l)} \geq 0$. The decoding process terminates until $\hat{\mathbf{x}}^{(l)}\mathbf{H}^T = 0$ or the preset maximum number of iterations is reached.

The Conventional Min-Sum Algorithm

In practice, the CN update in Eq. (3.17) contains non-linear operations $tanh(\cdot)$ which introduces a high computational complexity. To reduce the decoding complexity, one method is to use approximation function for computing the V2C messages. The simplified decoding algorithm, called the min-sum algorithm (MSA) computes the C2V message from c_m to v_n as [102]

$$Y_{mn}^{(l)} = \prod_{v_{n'} \in \mathcal{H}(c_m) \backslash v_n} \mathrm{sign}(Z_{n'm}^{(l-1)}) \cdot \min_{v_{n'} \in \mathcal{H}(c_m) \backslash v_n} \left| Z_{n'm}^{(l-1)} \right|. \tag{3.20}$$

The VN update and the calculation of the decision LLRs remains as the same as in Eqs. (3.18) and (3.19), respectively.

The Normalized MSA

It is shown in [102] that the C2V messages computed by Eq. (3.20) are inaccurate due to the approximation, which are always overestimated in magnitude compared to that of the SPA. This phenomenon causes performance degradation for the conventional MSA compared to the SPA, especially for the LDPC codes with a large number of low-degree VNs. To improve the error performance, the normalized MSA (NMSA) in [102] uses a scaling factor $\alpha^{(l)}$ to minimize the overestimation effect on the C2V messages, where Eq. (3.20) is modified into

$$Y_{mn}^{(l)} = \alpha^{(l)} \cdot \prod_{v_{n'} \in \mathcal{H}(c_m) \backslash v_n} \operatorname{sign}(Z_{n'm}^{(l-1)}) \cdot \min_{v_{n'} \in \mathcal{H}(c_m) \backslash v_n} \left| Z_{n'm}^{(l-1)} \right| . \tag{3.21}$$

Note that the scaling factor can either be a constant $\alpha^{(l)} = \alpha$ or an adaptive value, which is optimized based on iterations, SNR, or the node degrees by further adopting some mathematical tools such as the density evolution (DE). As shown in [48], the scaling factor can be optimized for most of LDPC codes to obtain the best decoding threshold or to make the magnitude of the C2V messages computed by both the NMSA and the SPA as close as possible. In practice, it is favorable to globally optimize the scaling factor based on the Tanner graph of an LDPC code which yields a fixed scaling factor.

The Offset MSA

An alternative method to improve the accuracy of the computed C2V messages was introduced in [102], yielding the offset MSA (OMSA). There is a correction term applied to the CN update by using a subtractive factor $\theta \in (0, 1)$. We modify Eq. (3.20) into

$$Y_{mn}^{(l)} = \prod_{v_{n'} \in \mathcal{H}(c_m) \backslash v_n} \operatorname{sign}(Z_{n'm}^{(l-1)}) \cdot \max_{v_{n'} \in \mathcal{H}(c_m) \backslash v_n} \left(\left| Z_{n'm}^{(l-1)} \right| - \theta^{(l)}, 0 \right) . \tag{3.22}$$

Similarly to the NMSA, the scaling factor $\theta^{(l)}$ can either be a constant or an adaptive value and need to be carefully designed. Although the NMSA and OMSA work well for regular LDPC codes as shown in [48], the degradation on error performance still remains for irregular LDPC codes.

The Two-dimensional Normalized MSA

There is one solution, so-called the two-dimensional normalized MSA (2D-NMSA), proposed in [49] to overcome the performance loss of the NMSA and the OMSA when decoding irregular LDPC codes. On top of the correction term for CN update as shown in Eq. (3.21), the V2C messages are also corrected by a scaling factor $\beta^{(l)} \in (0,1)$, i.e., the Eq. (3.18) is modified as

$$Z_{nm}^{(l)} = L(v_n) + \beta^{(l)} \cdot \sum_{c_{m'} \in \mathcal{H}(v_n) \backslash c_m} Y_{m'n}^{(l)} \tag{3.23}$$

To achieve the best possible performance for decoding irregular LDPC codes, both scaling factors $\alpha^{(l)}$ and $\beta^{(l)}$ need to be optimized [49].

The Self-Corrected MSA

In [50], V. Savin proposed the self-corrected MSA (SC-MSA) for the decoding of irregular LDPC codes. Different from the previous MS-based algorithms that make modifications on the magnitude of the extrinsic messages, SC-MSA can achieve an improved error performance for irregular LDPC codes by erasing the unreliable V2C messages which have different signs between two consecutive iterations. To be more specific, the C2V messages in the SC-MSA is computed in the same way as the conventional MSA, and the temporary V2C messages Z_{nm}^{tmp} at current iteration is first computed by Eq. (3.18). Then the self-correction operations are adopted for each VN as [50]

$$Z_{nm}^{(l)} \begin{cases} Z_{nm}^{tmp}, & \text{if sign}\left(Z_{nm}^{(l-1)}\right) = \text{sign}\left(Z_{nm}^{tmp}\right) \\ 0, & \text{otherwise} \end{cases} \tag{3.24}$$

As shown in [50], the SC-MSA can outperform the SPA in the error floor region for irregular LDPC codes with moderate block lengths. However, there is still noticeable performance degradation for irregular LDPC codes with short block lengths in the low SNR region.

The Conventional Weighted Bit-Flipping Algorithm

The WBF algorithm is a decoding algorithm based on flipping the hard-decision value of the least reliable VN messages according to the computed flipping metric at each iteration. Denoted by $\mathbf{s}^{(l)} = (s_1^{(l)}, s_2^{(l)}, \ldots s_M^{(l)})$ the syndrome vector computed at the l-th iteration. The conventional WBF algorithm computes the flipping metric for the node v_n at the l-th iteration according to [74]

$$E^{(l)}(v_n) = \sum\nolimits_{c_m \in \mathcal{H}(v_n)} (2s_m^{(l)} - 1) \cdot w_m, \tag{3.25}$$

where w_m is a weighted factor given by $w_m = \min\limits_{v_n \in \mathcal{H}(c_m)} |r(v_n)|$. Then the index of candidate bit(s) to be flipped can be determined by

$$\mathcal{F} = \{n | n = \arg \max_{1 \leq n \leq N} E^{(l)}(v_n)\}. \tag{3.26}$$

The process repeats until all the parity-check equations are satisfied or a preset maximum number of iterations is reached. As shown in [74], the conventional WBF algorithm may flip multiple bits selected from $\mathcal{F}$ in one iteration. Although the multi-bit flipping (MBF) rule leads to a fast convergence speed, carefully designed loop removal mechanisms are required to avoid the decoding process to be trapped in an infinite loop due to its greediness [101]. An alternative flipping rule for the conventional WBF algorithm is to randomly flip one bit in $\mathcal{F}$ at each iteration [74], which is also called the single-bit flipping WBF (SBF-WBF) decoding algorithm.

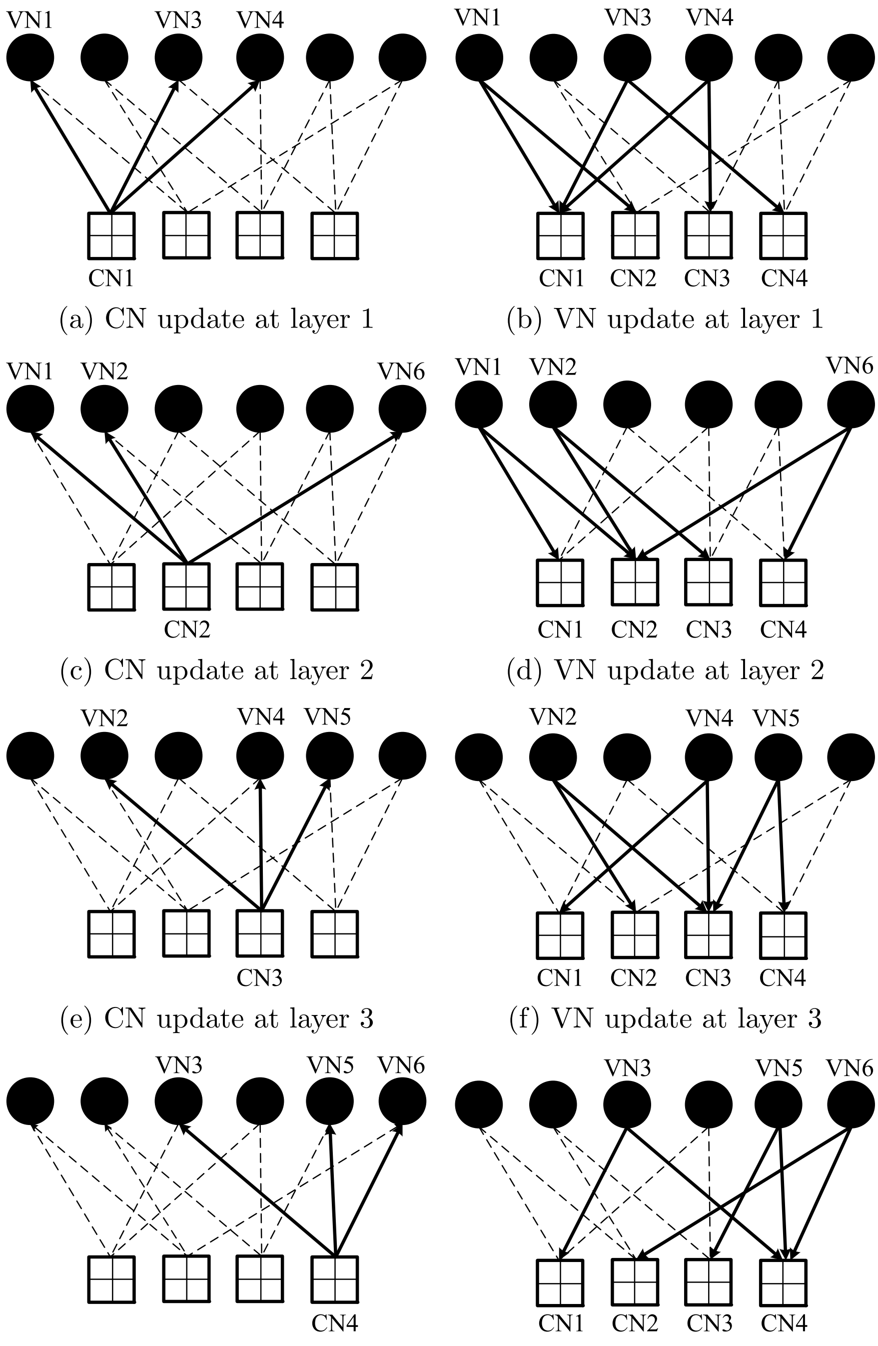
3.3.2 The Layered Decoding Schedule

Apart from the decoding algorithms, the decoding schedules can also affect the convergence speed of the decoder. As show in [94], the flooding schedule is the most straightforward decoding schedule, where all the C2V messages and all the V2C messages are updated simultaneously at each decoding iteration. On the contrast, the layered decoding schedules, such as layered BP (LBP) [136,137] and shuffled BP decoding [138], update the extrinsic messages in a fixed sequential order to accelerate the convergence speed. The basic idea of layered decoding is to divide the parity-check matrix $\mathbf{H}$ of an LDPC code into multiple groups based on its row or column, where each group can be regarded as a layer. A serial iteration strategy is adopted between each group, and the conventional parallel iteration strategy is adopted within the group by performing the decoding algorithms such as the SPA, MSA, and modified MSA.

The decoding process of one iteration for the typical LBP in [136] is illustrated in Fig. 3.7.

Note that in the layered decoding, a VN update is performed once all neighboring CNs are updated. Therefore, instead of using the node information of the last iteration, the update of the CNs in current iteration can utilize the messages from VNs that have been updated in the previous layer of the same iteration. This accelerates the convergence speed of the decoding. As shown in [139], these fixed serial decoding schedules increase the convergence speed by about twice than that of the flooding schedule.

Compared to sequential scheduling decodings that follow a fixed updating order, informed dynamic scheduling (IDS) in [140] considers the updating order



(a) CN update at layer 1 (b) VN update at layer 1

(c) CN update at layer 2 (d) VN update at layer 2

(e) CN update at layer 3 (f) VN update at layer 3

(g) CN update at layer 4 (h) VN update at layer 4

Figure 3.7: The decoding process of LBP

based on the residual of the messages and provides dynamic updating orders. Other variations of IDS strategies are also proposed in [141–143] to significantly reduce the number of iterations required for the decoding process. However, the challenge is how to implement the IDS in practical hardware due to the additional computational complexity. Thus, in the later chapter, we only consider the typical layered schedule as shown in [136].

3.3.3 The Improved Decoding Architectures

It is well-known that the smallest cycle in LDPC code graphs introduces a notable performance loss, particularly for the LDPC codes with short block length. To solve this problem, several QML decoding methods were proposed in [108–110], which are based on the reprocessing. More specifically, these conventional QML decoding methods perform the reprocessing with the maximum number of reprocessing stage $j_{\max}$ after the failure of the initial BP decoding test. At each stage of the reprocessing, they choose unreliable VNs according to certain node selection methods. A list of the decoder input sequences is then generated by modifying the LLRs of the unreliable VNs in the received signal. The conventional BP decoding is then performed with each decoder input sequence and only the valid codewords are collected from the reprocessing. In the end, the best codeword is selected as the decoding output based on a decision metric.

Note that the performance of a QML decoding is affected by the accuracy of selecting unreliable VNs for the reprocessing. Moreover, the decoding complexity depends on the number of decoding tests performed in the reprocessing. Here, we briefly introduce the node selection method and the reprocessing of the conventional QML decoding methods in [108] and [109].

Node-wise Selection (NWS) Method

For the decoding tests at the j-th stage, the ABP decoding in [108] selects j unreliable VNs for saturation. Here the saturation of a node v_n means that its LLR $\mathbf{r}(v_n)$ is set to the maximum $(+S)$ or minimum $(-S)$ value, which is determined by the precision or quantization of the decoder. Define T as the index of the decoding test in the reprocessing. Denote the set of VNs saturated for the T-th decoding test by $\mathcal{V}_S^{(T)}$. To determine $\mathcal{V}_S^{(T)}$ for every T-th decoding test at stage j reprocessing ($T = 2^j-1, 2^j, \ldots, 2^{j+1}-2$), the ABP decoder chooses a new unreliable VN, denoted by $v_s^{(T)}$, and $(j-1)$ unreliable VNs which have been selected in the $\lfloor \frac{T-1}{2} \rfloor$-th decoding test at stage $(j-1)$ reprocessing. Let $T' = \lfloor \frac{T-1}{2} \rfloor$, the newly saturated VN $v_s^{(T)}$ is selected based on the output of the T'-th decoding test at stage $(j-1)$ reprocessing. More specifically, for a given output of the T'-th decoding test, find all VNs connected to unsatisfied CNs, which is defined as $\mathcal{V}^{(T')}$. In particular, when $T' = 0$, the T'-th decoding test refers to the initial BP decoding. Note that an unsatisfied CN means its associated syndrome is nonzero after the decoding. From all VNs in $\mathcal{V}^{(T')}$, we only consider the VN with the maximum node degree to be the candidate node. If there are more than one VN with the same maximum node degree, the VN with the minimal magnitude of its channel output $|\mathbf{r}(v_n)|$ is selected with priority.

It is also shown in [110] that the proposed QML decoder computes the reliability of each unsatisfied CN based on the LLRs. Starting from the unsatisfied CN with the largest LLR value, two least reliable VNs with the smallest $|\mathbf{r}(v_n)|$ connected to the CN are forced to flip their hard decision with a high priority. Note that the SMS decoding in [109] chooses the unreliable VNs for saturation solely based on $|\mathbf{r}(v_n)|$ for hardware considerations. It is notable that the node selection methods in [108–110] determine the unreliable VNs based on the node reliability $|\mathbf{r}(v_n)|$. Therefore, we call these node selection methods as NWS methods.

The Reprocessing

Denote $\mathbb{M} = \{\mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_{2^j}\}$ as the list of length-j LLR vectors for the saturated VNs at stage j reprocessing, where $\mathbf{m}_t \in \{+S, -S\}^j, t = 1, 2, \ldots, 2^j$. The stage j reprocessing performs 2^j BP decoding tests with the modified input LLRs. For each BP decoding test at the j-th stage, the initial LLR values of $\mathcal{V}_S^{(T)}$ are replaced by one $\mathbf{m}_t$ from $\mathbb{M}$, i.e., $\mathbf{r}(\mathcal{V}_S^{(T)}) = \mathbf{m}_t$, and the rest of VNs' initial LLR values remain the same from the channel outputs. When all vectors $\mathbf{m}_t$ in $\mathbb{M}$ are tested, the reprocessing moves to stage $j+1$. It stops until the maximum number of reprocessing stage $j_{\max}$ is reached. Thus, there are total $\sum_{j=1}^{j_{\max}} 2^j = 2^{j_{\max}+1} - 2$ additional decoding tests need to be performed subsequently if the conventional BP decoding fails. In the end, the ABP decoding outputs the best codeword which has the minimum Euclidean distance to the received sequence [108].

It is shown that the error performance of the ABP decoding for the $(155, 64)$ Tanner code in [108] can approach that of the ML decoding when the number of saturated VNs is relatively large, e.g., $j_{\max} = 11$, which results in high decoding complexity. However, for a small or moderate $j_{\max}$, there is still a significant performance loss. In addition, the SMS decoding in [109] simplifies the reprocessing in the sense that all the $j_{\max}$ unreliable VNs are selected simultaneously. Therefore, only $2^{j_{\max}}$ decoder input sequences are tested, which reduces the decoding complexity. However, this degrades the error performance compared to the ABP decoding with the same $j_{\max}$ since the number of tested decoder input sequences reduces by almost half from $2^{j_{\max}+1} - 2$ to $2^{j_{\max}}$.

3.4 The Analysis and Design Tools of LDPC Codes

For practical considerations, there are usually multiple LDPC codes that share the same degree distribution for which we call these LDPC codes as an ensemble of LDPC codes [52, Definition 1.15]. It is known that a good error performance in the waterfall region is one of the essential factors that need to be considered in the design of LDPC codes. For an ensemble of binary LDPC codes, the density evolution (DE) [52], and the extrinsic information transfer (EXIT) chart [114, 144, 145] are the main mathematical tools to analyze and optimize its waterfall region performance under iterative decoding algorithms. In general, the analysis of DE methods are derived based on the following two assumptions [52]

1 **A cycle-free graph**: Under this assumption, the associated ensemble of LDPC codes always has a cycle-free graph under iterative decoding algorithm.

2 **Symmetry**: For binary inputs, we assume that $\mathrm{P}(y = q|x = +1) = \mathrm{P}(y = -q|x = -1)$, where x and y is the channel input and output, respectively. As such, the LLRs computed by the iterative decoding algorithms are symmetric.

Note that the first two assumptions guarantee that the input messages at each CN or VN are independent [52]. Then under the symmetric assumption, the performance of an LDPC code ensemble can be modeled by sending the all-zero codeword.

We further introduce the concept of decoding threshold to characterize the performance of an LDPC code ensemble in the waterfall region. The decoding threshold is defined as the point over which there is always a non-negligible error probability even after an infinite number of iterations [52]. For convenience, we introduce the DE [52] and the EXIT chart [114] in the following.

3.4.1 The Density Evolution Analysis

Density evolution (DE) is a technique used to obtain the decoding threshold for an LDPC code ensemble under iterative decoding. Assume that we adopt SPA for decoding, and the soft information passed between CNs and VNs are represented by LLRs. By considering an AWGN channel, we use Z to represent the V2C messages and Y to represent the C2V messages during iterative decoding, respectively. For a degree-d_v VN, the DE performs the update rule as

$$Z = \sum_{i=0}^{d_v-1} Y_i, \tag{3.27}$$

where Y_i $(i = 1, 2, \ldots, d_v - 1)$ is the messages from $(d_v - 1)$ its neighboring CNs, and Y_0 is the channel message.

Since these LLRs are of continuous values, we can use the probability density function to describe the probability that an LLR takes a particular value. By considering that the channel is memoryless and the factor graph of the LDPC code has no cycles, Eq. (3.27) indicates the summation of d_v random variables. Therefore, the probability density of Z can be expressed as a convolution of the density function of Y_i as

$$p_Z = \otimes_{i=0}^{d_v-1} p_{Y_i}, \tag{3.28}$$

where $\otimes$ refers to the convolution. For the case of irregular LDPC codes, we introduce the degree distribution $\lambda(x)$, i.e. [52],

$$p_Z = p_{Y_0} \otimes \left(\sum_d \lambda_d \otimes_{i=1}^{d} p_{Y_i} \right). \tag{3.29}$$

For a degree-d_c CN, the DE performs the operation as the MAP decoding for a $(d_c, d_c - 1)$ single parity-check code, which is given by

$$\tanh\left(\frac{Y}{2}\right) = \prod_{j=1}^{d_c-1} \tanh\left(\frac{Z_j}{2}\right), \tag{3.30}$$

where Y_j $(j = 1, 2, \ldots, d_c - 1)$ is the messages from $(d_c - 1)$ its neighboring VNs. For the efficient calculation by using Fourier transforms, we take the logarithm on both sides of Eq. (3.30) to convert the above product operation into a summation as [52]

$$\left(\mathrm{sgn}(Y), \log\left|\tanh\left(\frac{Y}{2}\right)\right|\right) = \sum_{j=1}^{d_c-1} \left(\mathrm{sgn}(Z_j), \log\left|\tanh\left(\frac{Z_j}{2}\right)\right|\right), \tag{3.31}$$

where $\mathrm{sgn}(\cdot)$ is defined as

$$\mathrm{sgn}(x) = \begin{cases} 0, & x \geq 0 \\ 1, & \text{otherwise} \end{cases} \tag{3.32}$$

Define $\gamma(x)$ as

$$\gamma(x) = \left(\mathrm{sgn}(x), \log\left|\tanh\left(\frac{x}{2}\right)\right|\right). \tag{3.33}$$

For the l-th iteration, Eq. (3.31) can be simplified to

$$\gamma(Y^{(l)}) = \sum_{j=1}^{d_c-1} \gamma(Z_j^{(l-1)}). \tag{3.34}$$

Further, the C2V messages can be expressed as

$$Y^{(l)} = \gamma^{-1}\left(\sum_{j=1}^{d_c-1} \gamma(Z_j^{(l-1)})\right). \tag{3.35}$$

This achieves the DE calculation of the CN on the Fourier domain. Note that there are some researchers using the distribution functions to describe DE (see the literature [146], [54]), and a discretized DE method was also proposed in [73] for efficient numerical calculation by computing C2V messages based on a lookup table.

3.4.2 The Extrinsic Information Transfer Chart

The DE analysis introduced in the previous section is based on the continuous space. It is necessary to track the evolution of the density function in the infinite dimensional space, which results in a high computational complexity. To simplify the analysis and calculation, many researchers have proposed a low-complexity calculation method for DE under the BI-AWGN channel in [114, 147–149]. One of the well-known methods is called extrinsic information transfer (EXIT) chart. Note that the EXIT chart is performed based on mutual information [114], which can give a relatively accurate decoding threshold prediction even for irregular LDPC codes. In the EXIT chart, it assumes that the messages transmitted on the factor graph has an approximate Gaussian distribution.

According to the Gaussian distribution, we have

$$\mu_Y = \mathbb{E}\left\{Y_{0,n}\right\} = \frac{2E_s}{\sigma_n^2}, \tag{3.36}$$

$$\sigma_Y = \mathrm{Var}\left\{Y_{0,n}\right\} = \frac{4E_s}{\sigma_n^2} = 2\mu_Y. \tag{3.37}$$

Thus, $Y_0 \sim \mathcal{N}(\mu_Y, 2\mu_Y)$ is called consistent Gaussian random variable which has a variance of twice of the mean. Suppose that all input messages for a CN or VN are independent, and all messages have a consistent Gaussian probability

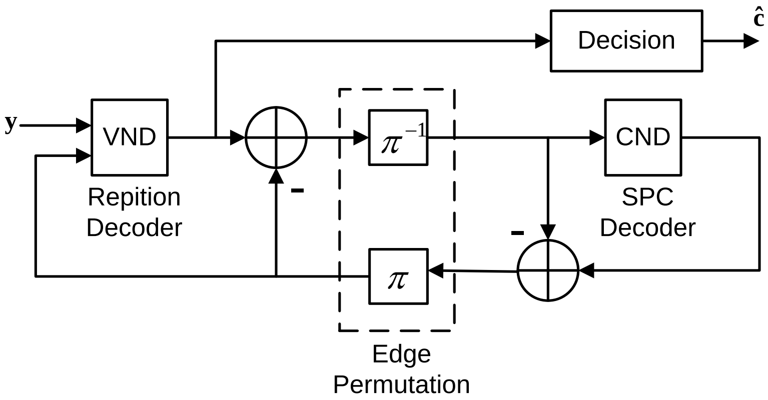


Figure 3.8: The iterative decoder structure of LDPC code.

density. The iterative decoding of an LDPC code is equivalent to decoding a serial concatenated coding system consisting of a repetition code and a SPC code as shown in Fig. 3.8, where a repetition code is used as an inner code and a SPC code is used as an outer code. For the l-th iteration, we take the expectation on both side of Eq. (3.27) and obtain the update rule for a degree-d_v VN as

$$\mu_Z^{(l)} = \mu_{Y_0} + (d_v - 1)\mu_Y^{(l-1)}, \tag{3.38}$$

where $\mu_Y^{(0)} = 0$. Therefore, the EXIT function for the VN can be computed by

$$I_{E,v} = J\left(\frac{2E_s}{\sigma_n^2} + (d_v - 1)J^{-1}(I_{A,v})\right), \tag{3.39}$$

where $I_{A,v}$ is the prior mutual information for VN and the function $J(\cdot)$ and $J^{-1}(\cdot)$ are given as [144]:

$$J(\sigma) \approx \begin{cases} A_{J,1}\sigma^3 + B_{J,1}\sigma^2 + C_{J,1}\sigma, & 0 \le \sigma \le 1.6363 \\ 1 - e^{A_{J,2}\sigma^3 + B_{J,2}\sigma^2 + C_{J,2} + D_{J,2}}, & 1.6363 < \sigma < 10 \\ 1, & \sigma \ge 10 \end{cases} \tag{3.40}$$

where the parameters in the equation are given in Table. 3.4.2 [144].

$A_{J,1}$	$B_{J,1}$	$C_{J,1}$	$A_{J,2}$	$B_{J,2}$	$C_{J,2}$	$D_{J,2}$
-0.0421061	0.209252	-0.00640081	0.00181491	-0.142675	-0.0822054	0.0549608

The inverse function $J^{-1}(\cdot)$ can be obtained from

$$\sigma = J^{-1}(I) \approx \begin{cases} A_{\sigma,1} I^3 + B_{\sigma,1} I^2 + C_{\sigma,1}\sqrt{I}, & 0 \leq I \leq 0.3646 \\ -A_{\sigma,2} \ln\left[B_{\sigma,2}(1-I)\right] - C_{\sigma,2} I, & 0.3646 < I < 1 \end{cases} \tag{3.41}$$

where the parameters in the equation are given in Table. 3.4.2 [144].

$A_{\sigma,1}$	$B_{\sigma,1}$	$C_{\sigma,1}$	$A_{\sigma,2}$	$B_{\sigma,2}$	$C_{\sigma,2}$
1.09542	0.214217	2.33727	0.706692	0.386013	-1.75017

For the update rule of $\mu_Y^{(l)}$ according to [149], we have

$$\mu_Y^{(l)} = J^{-1}\left(\frac{1}{\ln 2}\sum_{i=1}^{\infty}\frac{1}{(2i)(2i-1)}\left(\Phi_i(\mu_Z^{(l)})\right)^{d_c-1}\right), \tag{3.42}$$

where the function $\Phi_i(m)$ is computed by

$$\Phi_i(m) \approx \int_{-1}^{+1}\frac{2t^{2i}}{(1-t^2)\sqrt{4\pi m}}\exp\left(-\frac{\left(\ln\frac{1+t}{1-t}-m\right)^2}{4m}\right)dt. \tag{3.43}$$

Thus, the EXIT function for the CN can be represented as

$$I_{E,c} = \frac{1}{\ln 2}\sum_{i=1}^{\infty}\frac{1}{(2i)(2i-1)}\left(\Phi_i(J^{-1}(I_{A,c}))\right)^{d_c-1}, \tag{3.44}$$

where $I_{A,c}$ is the prior mutual information for CN. As shown in [144], $I_{E,c}$ can be approximately calculated as

$$I_{E,c} \approx 1 - J\left(\sqrt{d_c-1}\,J^{-1}(1-I_{A,c})\right). \tag{3.45}$$

In the iterative decoding process, the prior mutual information of the VN is the mutual information of the C2V message from its neighboring CN, and vice versa. Therefore, we have the following relationship [114]

$$I_{A,c} = I_{E,v}, \text{ and } I_{A,v} = I_{E,c}. \tag{3.46}$$

As a result, the EXIT chart can be obtained by simultaneously drawing the EXIT functions of the VN and CN on a graph. As shown in Fig. 3.9, the EXIT chart of (3, 6) regular LDPC codes under the BI-AWGN channel with SNR = 1.1 dB. The decoding threshold is defined as the SNR value at which the two EXIT function curves intersect with each other.

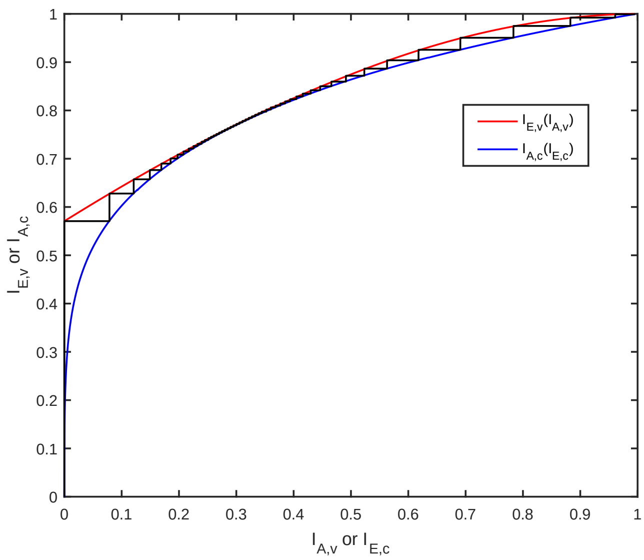


Figure 3.9: The EXIT chart of (3,6) regular LDPC code.

In addition, for irregular LDPC codes, the update formula of $\mu_Z^{(l)}$ and $\mu_Y^{(l)}$ become [144]

$$\mu_Z^{(l)} = J^{-1}\left(\sum_{d=1}^{d_v} \lambda_d J\left(\frac{2E_s}{\sigma_n^2} + (d-1)\mu_Y^{(l-1)}\right)\right). \tag{3.47}$$

$$\mu_Y^{(l)} = J^{-1}\left(\sum_{d=1}^{d_c-1}\rho_d\frac{1}{\ln 2}\sum_{i=1}^{\infty}\frac{1}{(2i)(2i-1)}\left(\Phi_i(\mu_Z^{(l)})\right)^{d-1}\right). \tag{3.48}$$

Therefore, the EXIT function for VN and CN can be given by [144]

$$I_{E,v} = \sum_{d=1}^{d_v}\lambda_d I_{E,v_d} = \sum_{d=1}^{d_v}\lambda_d J\left(\frac{2E_s}{\sigma_n^2} + (d-1)J(I_{A,v})\right). \tag{3.49}$$

$$I_{E,v} = \sum_{d=1}^{d_c}\rho_d I_{E,c_d} = \sum_{d=1}^{d_c}\frac{1}{\ln 2}\sum_{i=1}^{\infty}\frac{1}{(2i)(2i-1)}\left(\Phi_i(J^{-1}(I_{A,c}))\right)^{d-1}. \tag{3.50}$$

3.5 Summary

In this chapter, we present the basic knowledge and backgrounds for LDPC codes related to the research works in the remaining Chapters in the thesis. The main points presented in this chapter are summarized as follows.

- We briefly introduce the definitions and different representations of LDPC codes.

- We introduce various decoding algorithms, the layered scheduling strategy, and the decoding architecture for LDPC codes. The necessity of the improvement for current decoding methods is also discussed.

- We present two analytical tools to characterize the performance of LDPC codes, which can also be used for optimizing and constructing good LDPC codes in practice.

- We present some examples for classic LDPC codes, e.g., protograph-based LDPC codes and the 5G LDPC codes, which are widely used in practical applications.

Chapter 4

Euclidean Geometry Based Spatially-Coupled LDPC Codes and Windowed Decoding Scheme

4.1 Introduction

In the past three decades, NAND Flash memories took more and more places in the non-volatile memory application fields for their higher throughput and lower power consumption compared to conventional hard-disk drives (HDDs) [75]. To reduce the cost of NAND Flash memory, many technologies such as multi-level cell (MLC), triple-level cell (TLC), and 3D stacking are adopted [76, 77], where more information per storage element or more storage elements are packed together. However, one of the drawbacks by using these technologies is that the error rate of the stored information and the endurable program/erasure cycles of the storage cells will deteriorate when more information per storage element or more storage elements are packed in a small package. To deal with this issue, the powerful error correction codes (ECCs), such as Bose-Chaudhuri-Hocquenghem (BCH) codes [78], concatenated codes [79] and low-density parity-check (LDPC) codes [82] [83], were proposed in the literature. Among these ECCs, LDPC codes took the place for their near-capacity error correction capability when soft information is available [70] [84]. Recently, spatially-coupled (SC) LDPC codes [85] drawn significant attentions by many researchers from the perspective of both code construction and decoding methods [87–92] due to their universally achieving capacity under belief propagation [101]. Since most of the channels are typically binary in storage applications, in this work, we construct binary spatially-coupled (SC) low-density parity-check (LDPC) codes based on Euclidean geometry (EG) LDPC codes for storage applications, where both high error correction capability, extremely low uncorrectable bit error rate (UBER) and low decoding complexity are required. Then we propose a reliability-based windowed decoding (RBWD) scheme for SC LDPC codes, where a partial message reservation (PMR) method and a partial syndrome check (PSC) stopping rule are introduced for each decoding window to mitigate the error propagation.

4.2 Problem Statement

As the convolutional counterparts of LDPC codes [86], SC LDPC codes [85,87–92] have drawn attention of many researchers recently. It is shown in [89] that SC LDPC codes can combine the capacity-achieving property of the irregular LDPC codes and the linear minimum distance growth property of the regular LDPC codes. For the same systematic decoding latency or the same decoding complexity, SC LDPC codes have a considerable convolutional gain compared to the associated block LDPC codes, especially in the low raw bit error rate (RBER) or high signal-to-noise-ratio (SNR) regime [85]. Therefore, it is very interesting to consider SC LDPC codes for the next-generation NAND Flash memories, which require both high error correction capability and very low uncorrectable bit error rate (UBER). We know that the LDPC codes constructed based on finite geometries can have near-capacity performance and low error floor [46, 150, 151]. In addition, these LDPC codes have various options of the decoding algorithms, from majority logic decoding algorithms to iterative message-passing decoding algorithms, from hard-decision decoding algorithms to soft-decision decoding algorithms [74, 152–155], which provides a wide complexity-performance trade-off for different applications to practical systems. Among the finite geometry based LDPC codes, Euclidean geometry (EG) LDPC codes have superior error performance with low complexity decoding algorithms, such as the conventional weighted bit-flipping (WBF), as shown in [46]. Thus, it is of great potential to consider the construction of SC LDPC codes based on Euclidean geometry with further improved error correction capability and low UBER.

On the other hand, an applicable way to decode an SC LDPC code is to use a sliding windowed decoder [93, 156]. Compared to the full block decoding (FBD) which decodes the entire codeword of an SC LDPC code with full flooding schedule [94] [91], the sliding windowed decoder shifts along the Tanner graph and focuses on decoding only a portion of a codeword at a time, which results in a lower decoding latency and memory requirement. Since the windowed decoding architecture causes performance degradation compared to the FBD [97], most of the previous work, such as [95–97], focused on improving the performance of the sliding windowed decoder with soft-decision decoding algorithms such as sum-product algorithm (SPA). The SPA leads to a high decoding complexity as soft information is passed along the edges in the Tanner graph [98]. Therefore, the reliability-based decoding algorithms, such as the conventional WBF algorithm, are investigated by many researchers to obtain a lower decoding complexity with acceptable performance degradation [74, 100, 101], where only hard information is passed along the edges in the Tanner graph. However, we observe that there is a significantly high error floor when the conventional WBF algorithm is used for windowed decoding of SC LDPC codes. Since a sliding windowed decoder only covers a portion of the full Tanner graph, there exist variable nodes (VNs) that have neighbouring check nodes (CNs) outside the decoding window. Thus, the messages sent out from these VNs may not be reliable. Moreover, these unreliable messages are propagated to the next window and deteriorate the error rate performance of the code.

4.3 Main Contributions

Motivated by the advantages of the EG LDPC codes, we consider the problem of how to construct binary SC LDPC codes based on EG LDPC block codes for storage applications in this work, where high error correction capability, extremely low UBER and a low complexity decoding algorithm are required. In particular, we propose a construction method for SC LDPC codes based on EG LDPC codes. In the proposed construction method, a *two-dimensional edge-spreading* process is introduced to generate a base matrix for SC LDPC codes. More specifically, we employ a circulant decomposition method [157] to perform the two-dimensional edge-spreading on a protograph to obtain a base matrix. Then, instead of unwrapping a parity-check matrix directly, our method unwraps the base matrix of a protograph before the lifting operation. We show that the proposed method can be used to construct SC LDPC codes for various code lengths and code rates. The error rate performance of the constructed SC LDPC codes is evaluated by using a reliability-based decoding algorithm, which has much lower implementation complexity compared to the message passing based decoding algorithms. This is very important for future mobile data storage applications which require low cost and low power consumption in their decoders. In addition, motivated by the high error floor under the windowed decoding with the conventional WBF algorithm, we propose a new approach to perform windowed decoding, called the reliability-based windowed decoding (RBWD) scheme. We also consider an improved stopping rule for the windowed decoding scheme. The contributions of this work are summarized below:

- We present a systematic way to construct binary SC LDPC codes from m-dimensional EG with $m > 2$. We call the proposed codes EG-based SC (EG-SC) LDPC codes. Our proposed construction method is very flexible in selecting the underlying block codes, and has more degrees of freedom in optimizing the constructed SC LDPC codes. More specifically, the proposed construction starts from an EG with K cycle classes. By decomposing the generator vector of each cycle class into θ new generator vectors of smaller length, one can construct a $\theta \times \theta$ base matrix for each cycle class. By juxtaposing $K' \leq K$ different base matrices side-by-side and then performing matrix unwrapping, we obtain an unwrapped base matrix. This unwrapped base matrix is repeated periodically to construct the base matrix for a terminated SC LDPC code with design code rate $R_{SC} = \{1-(m_s+L)/LK'\}$, where m_s is the syndrome former memory and L is the termination length.
- We propose a two-dimensional edge-spreading process for EG-SC LDPC codes based on circulant decomposition [157]. The generator vector of each cycle class is decomposed into θ generator vectors, each of which is a generator vector of a smaller cycle class. A base matrix of a cycle class can be obtained by permuting the positions of each generator vector and its

nonzero elements. The process irregularly spreads a regular protomatrix in both column and row directions, such that the new base matrix satisfies a column-and-row summation constraint.

- We compute the exact rank of the parity-check matrix of an EG LDPC code, and use it to derive a lower bound on the rank of the parity-check matrix of our proposed EG-SC LDPC codes. We show that the derived lower bound is determined by the rank of the unwrapped parity-check matrix and it is a function of $Y \geq 2$, where Y is the number of times that an unwrapped matrix is periodically repeated.

- We propose a RBWD scheme for the decoding of SC LDPC codes to significantly reduce the error floor. In the scheme, we propose the PMR method which only reserves the reliable messages between two adjacent windows to avoid the error propagation. We also propose the PSC stopping rule to check the complete VNs for each decoding window. In this way, the error floor of the SC LDPC codes with windowed decoding can be significantly reduced.

- We evaluate the error performance of the constructed EG-SC LDPC codes using a WBF decoding algorithm based on FBD. We show that the proposed EG-SC LDPC codes achieve a considerable convolutional gain compared to their EG LDPC code counterparts. We demonstrate that the UBER performance of the proposed EG-SC LDPC codes have no error floor at the UBER of $10^{-9} \sim 10^{-10}$, whereas the protograph SC LDPC codes and the regular LDPC codes show an error floor around the UBER of 10^{-8} and 10^{-7}, respectively. More importantly, we further evaluate the error performance of the proposed RBWD scheme, where the bit error rate (BER) performance of the RBWD scheme can approach that of FBD within 0.1 dB, which is highly desirable for applications with a low decoding complexity requirement.

4.4 A General Construction of SC LDPC Codes from EG LDPC Codes

In this section, we present a general way to construct SC LDPC codes using the m-dimensional EG LDPC codes, where $m > 2$. The proposed construction method is performed based on the conventional matrix unwrapping technique on regular protographs. The key feature of the proposed construction is that the base matrix $\mathbf{B}$ of a protograph $\mathscr{P}$ undergoes a *two-dimensional* edge splitting process. To illustrate our point, let us consider the base matrix $\mathbf{B} = [b_0, b_1, \ldots, b_{v-1}]$ of size $1 \times v$, where $\{b_j\}^{1\times v} \in \mathbb{Z}^+$. We first obtain W descendent base matrices $\left[\mathbf{B}^{(0)}, \mathbf{B}^{(1)}, \ldots, \mathbf{B}^{(W-1)}\right]$ juxtaposed side-by-side, where each submatrix has size $1 \times v$ such that the summation constraint $\sum_{i=0}^{W-1} \mathbf{B}^{(i)} = \mathbf{B}$ is satisfied. Next, we obtain $W-1$ permutations of $\left\{\mathbf{B}^{(i)}\right\}_{0\leq i<W}$ by cyclicly shifting each $\mathbf{B}^{(i)}$ to the right. As a result, we obtain a $W \times W$ array of $1 \times v$ base matrix $\left\{\mathbf{B}^{(i,j)}\right\}_{0\leq i,j<W}$ such that it satisfies the column-and-row summation constraint

$$\sum\nolimits_{i=0}^{W-1} \mathbf{B}^{(i,j)} = \mathbf{B} \quad \text{for} \quad 0 \leq j < W. \tag{4.1}$$

and

$$\sum\nolimits_{j=0}^{W-1} \mathbf{B}^{(i,j)} = \mathbf{B} \quad \text{for} \quad 0 \leq i < W. \tag{4.2}$$

Due to the fact that W may be larger than the maximum $b_j, 0 \leq j < v$, and we allow each $\mathbf{B}^{(i)}$ to be different. Thus, the process of two-dimensional edge spreading results in a base matrix that contains zero, which will be lifted by an all-zero square matrix. To avoid any confusions, the important notations used in this paper are summarized in TABLE 4.1.

Notation	**Description**
$\mathbf{B}_{(m,2,s)}$	$\mathbf{B}_{(m,2,s)} = \left[\mathbf{B}^{(j)}_{(m,2,s)}\right]_{0\le j<K}$, the base matrix of an EG LDPC code with K circulants.
$\mathbb{B}_{(m,2^s)}$	Equivalent base matrix of $\mathbf{B}_{(m,2,s)}$ after circulant decomposition.
$\mathbb{B}^{uw}_{(m,2^s)}$	Unwrapped version of $\mathbb{B}_{(m,2^s)}$.
$\mathbb{B}^{SC}_{(m,2^s)}$	Base matrix of the terminated SC LDPC code.
$\mathbf{H}_{(m,2^s)}$, $\mathbb{H}_{(m,2^s)}$, $\mathbb{H}^{uw}_{(m,2^s)}$, $\mathbb{H}^{SC}_{(m,2^s)}$	Corresponding parity-check matrices lifted from $\mathbf{B}_{(m,2^s)}, \mathbb{B}_{(m,2^s)}, \mathbb{B}^{uw}_{(m,2^s)}, \mathbb{B}^{SC}_{(m,2^s)}$, respectively.

Table 4.1: Summary of notations

4.4.1 A general construction of EG-SC LDPC codes

From the above review of in Chapter 3, it is interesting to see that the EG LDPC codes are naturally cyclic, which, in practice, possess an efficient encoding and decoding implementation [158]. We now provide the general construction method of EG-SC LDPC codes.

Since each circulant $\mathbf{H}^{(j)}_{(m,2^s)}, 0 \le j < K$, has column and row weights of 2^s, we denote the regular base matrix of the associated protograph as

$$\mathbf{B}_{(m,2^s)} = \left[\mathbf{B}^{(0)}_{(m,2^s)}, \mathbf{B}^{(1)}_{(m,2^s)}, \ldots, \mathbf{B}^{(K-1)}_{(m,2^s)}\right], \tag{4.3}$$

where each $\mathbf{B}^{(j)}_{(m,2^s)} = 2^s$ for $0 \le j < K$. Next, choose a $\theta \in \mathbb{Z}^+$ such that θ divides $(2^{ms} - 1)$, we perform a two-dimensional edge spreading operation on each $\mathbf{B}^{(j)}_{(m,2^s)}$ to obtain a new base matrix

$$\mathbb{B}^{(j)}_{(m,2^s)} = \begin{bmatrix} b^{0,0}_{(m,2^s)} & b^{0,1}_{(m,2^s)} & \cdots & b^{0,\theta-1}_{(m,2^s)} \\ b^{1,0}_{(m,2^s)} & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ b^{\theta-1,0}_{(m,2^s)} & \cdots & \cdots & b^{\theta-1,\theta-1}_{(m,2^s)} \end{bmatrix}, \tag{4.4}$$

such that the column-and-row summation constraints given in (4.1) and (4.2) are satisfied. In this construction, each row summation is

$$\sum\nolimits_{j=0}^{\theta-1} b^{i,j}_{(m,2^s)} = 2^s \quad \text{for} \quad 0 \leq i < \theta,$$

and each column summation is

$$\sum\nolimits_{i=0}^{\theta-1} b^{i,j}_{(m,2^s)} = 2^s \quad \text{for} \quad 0 \leq j < \theta.$$

Note that $\mathbf{B}^{(j)}_{(m,2^s)} = 2^s$ and we split it into a $\theta \times \theta$ base matrix $\mathbb{B}^{(j)}_{(m,2^s)}$. Then, a base matrix $\mathbb{B}_{(m,2^s)}$ of size $\theta \times K\theta$ can be constructed by juxtapose all K base matrices $\mathbb{B}^{(j)}_{(m,2^s)}$ side-by-side with each row summation equals to $K2^s$ and each column summation equals to 2^s.

It is notable mention that we do not specify the choice of each $b^{i,j}_{(m,2^s)}$ at this stage. Instead, we shall see, from the next section that, both the choice of each $b^{i,j}_{(m,2^s)}$ and the exact cyclic shifts of circulant permutation matrices (CPMs) can be obtained after performing circulant decomposition. The process of the circulant decomposition breaks each one of the K generator vectors into θ small pieces of subvectors so that these subvectors are employed as a generator vector of a smaller circulant. The cardinalities of the K sets of θ subvectors are used as the first row of the base matrix $\mathbb{B}_{(m,2^s)}$, given in (4.4), whereas the rest rows of $\mathbb{B}_{(m,2^s)}$ are cyclic right shift of the first row in a particular way, which will be further discussed in the next section.

Since the set of factors of $2^{ms}-1$ contains odd primes only and the product of two odd numbers is still an odd number, we will set the syndrome former memory $m_s = \theta - 1$ to make sure that the periodicity of the final SC base matrix $\mathbb{B}^{SC}_{(m,2^s)}$ is θ. To this end, the base matrix $\mathbb{B}_{(m,2^s)}$ that contains K' base matrices $\mathbb{B}^{(j)}_{(m,2^s)}$ is then unwrapped using the conventional cut-and-paste operation [86] to construct a 'stair-like' diagonal base matrix $\mathbb{B}^{uw}_{(m,2^s)}$ with stair width K'. The unwrapped base matrix $\mathbb{B}^{uw}_{(m,2^s)}$ is of size $(m_s + \theta) \times K'\theta$. By coupling $\mathbb{B}^{uw}_{(m,2^s)}$ Y times, a base matrix $\mathbb{B}^{SC}_{(m,2^s)}$ of a terminated EG-SC LDPC code is obtained. Let γ be a positive integer such that $\gamma\theta = (2^{ms} - 1)$. Then each element $b^{i,j}_{(m,2^s)} \in \mathbb{B}^{(j)}_{(m,2^s)}$ is lifted by a weight-l CPM of order $U = \gamma$. The resulting parity-check matrix $\mathbb{H}^{SC}_{(m,2^s)}$ of size $\gamma\left(m_s + \theta Y\right) \times \gamma K'\theta Y$ corresponds to a terminated EG-SC LDPC code.

4.5 New Construction of EG-SC LDPC codes

In this section, we will introduce the process of circulant decomposition, which results in an edge-spread base matrix in both column and row directions. Notice that another feature of the resulting base matrix is that the descendent base matrices $\left\{\mathbb{B}^{(j)}_{(m,2^s)}\right\}_{0\leq j<K}$ of $\mathbf{B}_{(m,2^s)}$, in general, are different. This means that the proposed construction method possesses time-varying SC LDPC codes since each row of the base matrix of an EG-SC LDPC code could start with a different descendent base matrix.

4.5.1 Two-dimensional edge spreading

For an Euclidean geometry EG$(m, 2^s)$ and $m > 2$, there are K cycle classes that form K circulants, where each circulant can be generated by a generator vector.

Let $\mathcal{W} = \{w_1, w_2, \ldots, w_v\}$ be the set of factors of $(2^{ms} - 1)$[1] such that $1 \notin \mathcal{W}$. Let θ be the product of a subset of $\mathcal{W}$, and $\mathbf{h}^j_{(m,2^s)}$ be the first row of the j-th circulant. Note that $\mathbf{h}^j_{(m,2^s)}$ is also a generator vector of the j-th circulant. Then a set of θ new generator vectors can be obtained by having $\mathbf{h}^j_{(m,2^s)}$ permuted. The i-th portion of $\mathbf{h}^j_{(m,2^s)}$ is obtained from the following index sequence

$$\pi^j_i = [i, \gamma + i, 2\gamma + i, \ldots (\theta - 1)\gamma + i] \tag{4.5}$$

for $0 \leq i < \theta$ and $\gamma\theta = 2^{ms} - 1$. Thus, $\pi^j = \left[\pi^j_0, \pi^j_1, \ldots, \pi^j_{\theta-1}\right]$ gives a column permutation of $\mathbf{h}^j_{(m,2^s)}$, and it depends on the choices of θ and γ. Note that π^j is a sequence of binary γ-tuples. Denote by $\mathbf{f}_{\pi^j_i}$ the i-th generator vector of the j-th circulant. By cyclic shifting the elements inside π^j towards right, we obtain the base matrix of the j-th circulant that is given by

$$\mathbb{B}^{(j)}_{(m,2^s)} = \begin{bmatrix} |\mathbf{f}_{\pi^j_0}| & |\mathbf{f}_{\pi^j_1}| & \cdots & |\mathbf{f}_{\pi^j_{\theta-2}}| & |\mathbf{f}_{\pi^j_{\theta-1}}| \\ |\mathscr{C}_1(\mathbf{f}_{\pi^j_{\theta-1}})| & |\mathbf{f}_{\pi^j_0}| & \cdots & |\mathbf{f}_{\pi^j_{\theta-3}}| & |\mathbf{f}_{\pi^j_{\theta-2}}| \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ |\mathscr{C}_1(\mathbf{f}_{\pi^j_2})| & |\mathscr{C}_1(\mathbf{f}_{\pi^j_3})| & \cdots & |\mathbf{f}_{\pi^j_0}| & |\mathbf{f}_{\pi^j_1}| \\ |\mathscr{C}_1(\mathbf{f}_{\pi^j_1})| & |\mathscr{C}_1(\mathbf{f}_{\pi^j_2})| & \cdots & |\mathscr{C}_1(\mathbf{f}_{\pi^j_{\theta-1}})| & |\mathbf{f}_{\pi^j_0}| \end{bmatrix}, \tag{4.6}$$

where $|\mathbf{f}|$ denotes the cardinality of a vector $\mathbf{f}$ and $\mathscr{C}_1(\mathbf{f}_{\pi^j_i}) = (\mathbf{f}_{\pi^j_i} + 1) \bmod \gamma$ represents a cyclic right shift of all the elements in $\mathbf{f}_{\pi^j_i}$ by one position. Continuously performing the above method for all K circulants, we obtain the final base matrix

$$\mathbb{B}_{(m,2^s)} = \left[\mathbb{B}^{(0)}_{(m,2^s)}\ \mathbb{B}^{(1)}_{(m,2^s)}\ \ldots \mathbb{B}^{(K-1)}_{(m,2^s)}\right], \tag{4.7}$$

[1] $2^{ms} - 1$ can be factored since $2^{ms} - 1$ is divisible by $2^\theta - 1$ if and only if there exist a θ that divides ms. Hence, it is straightforward to see that ms is divisible by both $\theta = m$ or $\theta = s$.

where the size of the base matrix is $\theta \times K\theta$.

The above process of circulant decomposition [157] requires only the generator vector of each circulant. The result of the process irregularly spreads the edges of a regular protograph in both column and row directions. Although the elements of each $\mathbf{f}_{\pi_i^j}$ in the lower triangle of $\mathbb{B}^{(j)}_{(m,2^s)}$ are shifted, the column-and-row summation constraints in (4.1) and (4.2) are still satisfied since each base matrix $\mathbb{B}^{(j)}_{(m,2^s)}$ is cyclic and contains only the cardinality of each subvector $\mathbf{f}_{\pi_i^j}$. Moreover, each $\mathbf{f}_{\pi_i^j}$ contains the set of non-zero positions that can be used to generate the $\gamma \times \gamma$ circulant. Note that the generator vector $\mathbf{f}_{\pi_i^j}$ might be an empty vector, that is, $\mathbf{f}_{\pi_i^j} = \emptyset$. This is due to the fact that the corresponding π_i^j has only zero in it. However, this will not alter the performance of a code.

4.5.2 EG-SC LDPC codes from EG$(m, 2^s)$ codes with $m > 2$

In the following, we construct EG-SC LDPC codes from m-dimensional Euclidean geometry with $m > 2$. In this case, as described from Section 4.4-A, we have $K > 1$ circulants, each of size $2^{ms} - 1$. Choose a θ such that θ divides $2^{ms}-1$, and an integer K', where $1 < K' \leq K$. Let $\mathbb{B}_{(m,2^s)} = \left[\mathbb{B}^{(0)}_{(m,2^s)}, \mathbb{B}^{(1)}_{(m,2^s)}, \ldots, \mathbb{B}^{(K'-1)}_{(m,2^s)}\right]$ be the $\theta \times K'\theta$ base matrix given in (4.7). Denote each element of $\mathbb{B}_{(m,2^s)}$ as $e_{i,j}$ for $0 \leq i < \theta$ and $0 \leq j < K'\theta$. Recall that the syndrome former memory $m_s = \theta - 1$. By performing the matrix unwrapping technique on $\mathbb{B}_{(m,2^s)}$, we obtain the unwrapped $\mathbb{B}^{uw}_{(m,2^s)}$ of size $(m_s + \theta) \times K'\theta$ as

$$\mathbb{B}^{uw}_{(m,2^s)} = \left[\begin{array}{c} \mathcal{B}_0 \\ \mathcal{B}_1 \\ \vdots \\ \mathcal{B}_{\theta-1} \\ \hline \mathcal{B}_\theta \\ \vdots \\ \mathcal{B}_{\theta+m_s-1} \end{array}\right], \tag{4.8}$$

where

$$\mathcal{B}_i = [e_{i,0} \cdots e_{i,(i+1)K'-1} \mathbf{0}_{1\times K'(\theta-i-1)}] \text{for} 0 \leq i < \theta$$

and

$$\mathcal{B}_i = [\mathbf{0}_{1\times(i-\theta+1)K'} e_{(i-\theta),(i-\theta+1)K'} \cdots e_{(i-\theta),K'\theta-1}] \text{for} \theta \leq i < \theta + m_s.$$

Here, $\mathbf{0}_{1\times y}$ denotes an all-zero vector of length y. Hence, $\mathbb{B}^{uw}_{(m,2^s)}$ is a stair-like diagonal matrix with stair width K', which will be illustrated in Fig. 4.1 (part c). To obtain a base matrix for EG-SC LDPC codes, we repeat $\mathbb{B}^{uw}_{(m,2^s)}$ periodically Y times in a way presented below,

$$\mathbb{B}^{SC}_{(m,2^s)} = \begin{bmatrix} \multicolumn{4}{c}{\overbrace{\hphantom{\mathcal{B}_{\theta+m_s-1} \quad \mathcal{B}_{\theta+m_s-1} \quad \mathcal{B}_0 \quad \ddots \quad \mathcal{B}_0}}^{Y}} \\ \mathcal{B}_0 & & & & \\ \mathcal{B}_1 & & & & \\ \vdots & & & & \\ \mathcal{B}_\theta & \mathcal{B}_0 & & & \\ \vdots & \mathcal{B}_1 & & & \\ \mathcal{B}_{\theta+m_s-1} & \vdots & & & \\ & \mathcal{B}_\theta & \mathcal{B}_0 & & \\ & \vdots & \mathcal{B}_1 & \ddots & \\ & \mathcal{B}_{\theta+m_s-1} & \vdots & \ddots & \ddots \\ & & \mathcal{B}_\theta & \ddots & \mathcal{B}_0 \end{bmatrix}. \tag{4.9}$$

The resulting base matrix $\mathbb{B}^{SC}_{(m,2^s)}$ is a periodic stair-like diagonal matrix of size $(m_s + \theta Y) \times K'\theta Y$. Since $m_s = \theta - 1$ in our design, the period of the EG-SC LDPC codes is $m_s + 1 = \theta$, and $\mathbb{B}^{uw}_{(m,2^s)}$ given in (4.8) starts to repeat after every θ time instances. The final parity-check matrix of the EG-SC LDPC codes can be obtained by lifting each element of $\mathbb{B}^{SC}_{(m,2^s)}$ with CPM of order $U = \gamma$, where the exact cyclic shifts are given by the corresponding generator vector $\mathbf{f}_{\pi_i^j}$. The resulting parity-check matrix of the EG-SC LDPC codes, denoted as $\mathbb{H}^{SC}_{(m,2^s)}$, is of size $\gamma(m_s + \theta Y) \times \gamma K'\theta Y$. The above construction method is summarized in **Algorithm 4.1**, and graphically illustrated in Fig. 4.1.

It is obvious from (4.9) that the termination length is $L = \theta Y$ since every column of $\mathbb{B}^{SC}_{(m,2^s)}$ given in (4.9) contains θ columns of small base matrices, each of size $1 \times K'$. Thus, the design code rate of the EG-SC LDPC codes is given by

$$R_{SC} = \frac{LK' - (m_s + L)}{LK'} = \frac{K'\theta Y - (m_s + \theta Y)}{K'\theta Y}. \tag{4.10}$$

Similar to the conventional design of SC LDPC codes from protographs, we can see that as $Y \to \infty$, R_{SC} approaches the design code rate

$$R_{BC} = 1 - \frac{1}{K'} \tag{4.11}$$

of the corresponding EG LDPC block codes.

Algorithm 4.1 The Construction Method for EG-SC LDPC codes

1: For a m-dimensional Euclidean geometry, find K generator vectors for the K cycle classes.

2: Choose a θ such that θ divides $2^{ms} - 1$. Perform circulant decomposition on each one of the K generator vectors to obtain θ subvectors $\left\{\mathbf{f}_{\pi_0^j}, \mathbf{f}_{\pi_1^j}, \ldots, \mathbf{f}_{\pi_{\theta-1}^j}\right\}$.

3: Construct the base matrix $\mathbb{B}_{(m,2^s)}$ given in Equation (4.7).

4: For a given K' such that $1 < K' \leq K$, construct an unwrapped base matrix $\mathbb{B}^{uw}_{(m,2^s)}$ of size $(\theta + m_s) \times K'\theta$ according to (4.8).

5: Construct a $(m_s + \theta Y) \times K'\theta Y$ base matrix of the EG-SC LDPC code by repeating $\mathbb{B}^{uw}_{(m,2^s)}$ periodically Y times in the way shown in (4.9).

6: Lift each element of $\mathbb{B}^{SC}_{(m,2^s)}$ using a CPM of size $\gamma \times \gamma$ with exact cyclic shifts from $\mathbf{f}_{\pi_i^j}$ to obtain the $\gamma(m_s + \theta Y) \times \gamma K'\theta Y$ parity-check matrix $\mathbb{H}^{SC}_{(m,2^s)}$ of a EG-SC LDPC code, where $\theta\gamma = 2^{ms} - 1$.

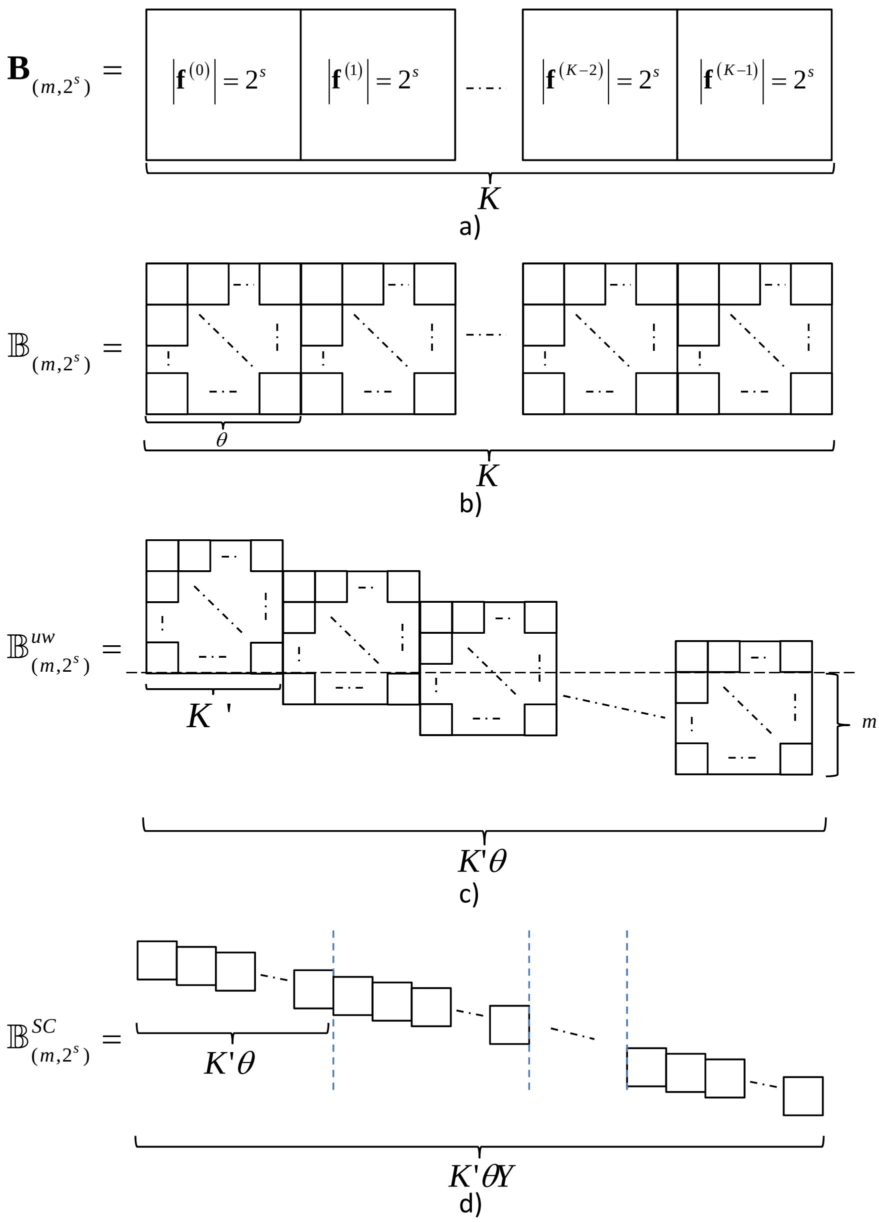


Figure 4.1: General construction of EG-SC LDPC codes. Part $a)$ gives a base matrix $\mathbf{B}_{(m,2^s)} = \left[|\mathbf{f}^{(0)}|, |\mathbf{f}^{(1)}|, \ldots, |\mathbf{f}^{(K-1)}|\right]$ of an m-dimensional EG code, where each $\mathbf{f}^{(j)}$, for $0 \leq j < K$, is the generator vector of a weight-2^s circulant. In part $b)$, the base matrix of a decomposed $\mathbf{B}_{(m,2^s)}$ is given. Each circulant is decomposed into a $\theta \times \theta$ array of $\gamma \times \gamma$ circulants, where θ is an integer that divides $2^{ms} - 1$ and $\gamma\theta = 2^{ms} - 1$. Part $c)$ shows the unwrapped stair-like diagonal base matrix $\mathbb{B}^{uw}_{(m,2^s)}$ with stair width K', where $1 < K' \leq K$ and the syndrome former memory $m_s = \theta - 1$. Part $d)$ shows the the periodic stair-like diagonal base matrix $\mathbb{B}^{SC}_{(m,2^s)}$ of an EG-SC code by coupling Y copies of $\mathbb{B}^{uw}_{(m,2^s)}$.

4.5.3 Rank analysis of EG-SC LDPC codes

In this section, we first determine the rank of the parity-check matrix of an EG LDPC block code, and then derive a lower bound on the rank of the proposed EG-SC LDPC codes. Denote by $\mathcal{R}\left(\mathbf{H}_{(m,2^s)}\right)$ the rank of matrix $\mathbf{H}_{(m,2^s)}$. The following result gives the exact $\mathcal{R}\left(\mathbf{H}_{(m,2^s)}\right)$ of an EG LDPC block code.

Theorem 1. *For an m-dimensional Euclidean geometry $EG(m,2^s)$ over $GF(2^s)$, the parity-check matrix $\mathbf{H}_{(m,2^s)}$ of $2^{ms}-1$ rows and $\left(2^{(m-1)s}-1\right)(2^{ms}-1)\,/\,(2^s-1)$ columns has a rank equal to [159]*

$$\mathcal{R}(\mathbf{H}_{(m,2^s)}) = \mathcal{F}(m,2^s) - \mathcal{F}(m-1,2^s) - 1, \tag{4.12}$$

where

$$\mathcal{F}(m,2^s) = \sum_{v_o}\sum_{v_1}\cdots\sum_{v_{s-1}}\prod_{j=0}^{s-1}\sum_{i=0}^{\Delta(v_{j+1},v_j)}(-1)^i\binom{m+1}{i}\binom{m+2v_{j+1}-v_j-2i}{m} \tag{4.13}$$

and $v_s = v_0$. The summations are taken over all possible integers v_j, $j = 0,1,\ldots,s-1$, such that

$$2 \le v_j \le m+1, \tag{4.14}$$

$$0 \le 2v_{j+1} - v_j \le m+1, \tag{4.15}$$

and $\Delta(v_{j+1},v_j)$ is the greatest integer not exceeding $(2v_{j+1}-v_j)/2$, i.e.,

$$\Delta(v_{j+1},v_j) = \left\lfloor \frac{2v_{j+1}-v_j}{2} \right\rfloor. \tag{4.16}$$

The proof of Theorem 1 can be found in [159, 160].

Note that, for a given EG code, the rank of the parity-check matrix $\mathbb{H}_{(m,2^s)} = \left[\mathbb{H}_{(m,2^s)}^{(i)}\right]_{0\leq i<K}$ is $\mathcal{R}\left(\mathbf{H}_{(m,2^s)}\right)$ since $\mathbb{H}_{(m,2^s)}$ and $\mathbf{H}_{(m,2^s)}$ are isomorphic up to column and row permutations. However, if $\mathbb{H}_{(m,2^s)}$ is a concatenation of $K' < K$ circulants, then we have $\mathcal{R}\left(\mathbb{H}_{(m,2^s)}\right) \leq \mathcal{R}\left(\mathbf{H}_{(m,2^s)}\right)$. In the following, we derive a lower bound on the rank of $\mathbb{H}_{(m,2^s)}^{SC}$, and show that the rank depends on Y, where Y is the number of times that an unwrapped matrix is periodically repeated.

Lemma 1. *For $m > 2$ and $Y \geq 2$, the rank of the parity-check matrix $\mathbb{H}_{(m,2^s)}^{SC}$ of a terminated EG-SC LDPC code is lower bounded by*

$$\mathcal{R}\left(\mathbb{H}_{(m,2^s)}^{SC}\right) \geq \mathcal{R}\left(\mathbb{H}_{(m,2^s)}^{uw}\right) + \mathcal{R}\left(\mathbf{Q}_{(m,2^s)}^{SC}\right)(Y-1), \tag{4.17}$$

where $\mathbb{H}_{(m,2^s)}^{uw}$ is the $(\theta + m_s)\gamma \times K'\theta\gamma$ unwrapped matrix, and $\mathbf{Q}_{(m,2^s)}^{SC}$ is the bottom-most $\theta\gamma$ rows of $\mathbb{H}_{(m,2^s)}^{uw}$.

The proof of Lemma 1 can be found in [91].

4.6 Constructed Codes and Numerical Results

An important advantage of the conventional EG codes is that their Tanner graphs contain no cycles of length four and their associated parity-check matrices have relatively high column and row weights. This allows conventional EG codes to be decoded in various ways, such as the hard-decision and soft-decision decodings, with a wide range of tradeoffs between decoding complexity, decoding speed and error performance. In this section, we evaluate our constructed EG-SC LDPC codes and the corresponding block codes using a WBF [46] decoding algorithm, which is a reliability based decoding algorithm with performance between hard and soft-decision decodings. We use the following example of the constructed codes to illustrate the construction of our proposed EG-SC LDPC codes, and show the performance of the constructed EG-SC LDPC codes using the flooding schedule decoding (FSD) [94].

Example 4.1. *Let* m=3, s=3 *and* p=2, *an EG LDPC code can be constructed from the* 3*-dimensional Euclidean geometry with parity-check matrix, denoted by* $\mathbf{H}_{(3,2^3)}$. *There are* $2^{ms}-1=511$ *non-origin points and the number of lines that pass through these non-origin points are* $\left(2^{(m-1)s}-1\right)(2^{ms}-1)/(2^s-1)=(2^6-1)\times 511/(2^3-1)=4599$ *[46]. Hence, the parity-check matrix* $\mathbf{H}_{(3,2^3)}$ *is of size* 511×4599. *Since* $K=\left(2^{(m-1)s}-1\right)/(2^s-1)=(2^6-1)/(2^3-1)=9$, *the* 4599 *lines can be grouped into nine cycle classes, each of which is a circulant matrix generated from its generator vector* $\mathbf{f}^{(j)}$. *The nine generator vectors of these circulants are*

$$\mathbf{f}^{(j)}=\begin{cases}[0,1,20,26,85,108,325,395] & j=0\\ [0,2,40,52,139,170,216,279] & j=1\\ [0,3,44,246,288,408,443,501] & j=2\\ [0,4,47,80,104,278,340,432] & j=3\\ [0,5,27,128,149,209,262,482] & j=4\\ [0,7,132,150,228,323,339,413] & j=5\\ [0,8,45,94,160,169,208,353] & j=6\\ [0,11,72,102,253,317,384,494] & j=7\\ [0,14,135,167,264,300,315,456] & j=8\end{cases}. \tag{4.18}$$

Since 511 *is the product of* 7 *and* 73, *the parity-check matrix* $\mathbf{H}_{(3,2^3)}$ *can be decomposed into either a* 7×63 *array of* 73×73 *circulants or a* 73×657 *array of* 7×7 *circulants. By choosing* $\theta=7$ *and following our construction method, we obtain a* 7×63 *array of* 73×73 *circulants* $\mathbb{H}_{(3,2^3)}=\left[\mathbb{H}^{(j)}_{(3,2^3)}\right]_{0\le j<9}$ *that can be used to construct a family of EG-SC LDPC codes with design rates* $R_{SC}=\{1-(m_s+\theta Y)/K'\theta Y\}_{2\le K'\le 9}$. *The rank of* $\mathbb{H}_{(3,2^3)}$ *is computed using Equation (4.12) with* $\mathcal{F}(3,2^3)=401$ *and* $\mathcal{F}(2,2^3)=28$. *Thus,* $\mathcal{R}\left(\mathbb{H}_{(3,2^3)}\right)=\mathcal{F}(3,2^3)-$

$\mathcal{F}(2,2^3)-1=372$, which is equal to $\mathcal{R}\left(\mathbf{H}_{(3,2^3)}\right)$. We find that $\mathcal{R}\left(\mathbb{H}^{uw}_{(3,2^3)}\right)=810=\mathcal{R}\left(\mathbb{H}_{(3,2^3)}\right)+m_s\gamma=372+6\times 73$, and this rank remains 810 for $2\leq K'\leq 9$. Moreover, let $K'=7$ and $Y=8$, we spatially-couple eight copies of $\mathbb{H}^{uw}_{(3,2^3)}$ to obtain the parity-check matrix $\mathbb{H}^{SC}_{(3,2^3)}$ of the EG-SC LDPC code with stair-like diagonal structure. The size of $\mathbb{H}^{SC}_{(3,2^3)}$ is 4526×28616 since there are $\gamma(m_s+\theta Y)=4526$ check nodes and $K'\theta\gamma Y=28616$ variable nodes. We find that $\mathcal{R}\left(\mathbb{H}^{SC}_{(3,2^3)}\right)=4387$, which achieves the equality of Equation (4.17), that is, $372+6\times 73+511\times(8-1)=4387$. Furthermore, $\mathcal{R}\left(\mathbb{H}^{SC}_{(3,2^3)}\right)$ remains 4387 for $2\leq K'\leq 9$. Hence, for $Y=8$, we can construct a family of EG$(3,2^3)$-SC LDPC codes with different code rates determined by K'. The exact code rates of the EG-SC LDPC codes are $R=1-4387/(4088K')$ for $2\leq K'\leq 9$.

Then we consider the EG$(3,2^3)$-SC LDPC codes shown in Example 4.1 with nine generator vectors given in (4.18). Let $\theta=7$, the 63 subvectors after circulant decomposition are

$$
\left[\mathbf{f}_{\pi_0^j},\mathbf{f}_{\pi_1^j},\ldots,\mathbf{f}_{\pi_{\theta-1}^j}\right]=
\begin{cases}
\{[0],[0,12],[\emptyset],[15,46,56],[\emptyset],[3],[2]\} & j=0\\
\{[0],[\emptyset],[0,24],[7],[\emptyset],[5],[19,30,39]\} & j=1\\
\{[0],[35,41],[6,58,63],[0],[71],[\emptyset],[\emptyset]\} & j=2\\
\{[0],[\emptyset],[\emptyset],[11],[0,48],[6,39,61],[14]\} & j=3\\
\{[0],[\emptyset],[18,21],[37],[\emptyset],[0],[3,29,68]\} & j=4\\
\{[0,1,60],[46],[\emptyset],[21,48],[32],[\emptyset],[18]\} & j=5\\
\{[0],[1,24],[\emptyset],[6,13,50],[\emptyset],[29],[22]\} & j=6\\
\{[0],[36],[10,45],[\emptyset],[1,14,70],[\emptyset],[54]\} & j=7\\
\{[0,2,45],[65],[19],[\emptyset],[\emptyset],[37],[23,42]\} & j=8
\end{cases},
\tag{4.19}
$$

where '$\emptyset$' denotes an empty set, which corresponds to an all-zero square matrix of size $U=\gamma$ when performing lifting. It can be seen that the number of non-empty elements of each row in (4.19) is $2^3=8$, while these 8 elements are irregularly distributed among 7 subvectors. For the j-th circulant, arrange the 7 subvectors into a base matrix as shown in Equation (4.6) by cyclicly shifting the order of the subvectors, and cyclicly shifting the elements of the subvectors in the lower triangle. By choosing $K' = 7$, a 7×49 base matrix $\mathbb{B}_{(3,2^3)} = \left[\mathbb{B}^{(0)}_{(3,2^3)}, \mathbb{B}^{(1)}_{(3,2^3)}, \ldots, \mathbb{B}^{(6)}_{(3,2^3)}\right]$ is obtained. By lifting each element of the base matrix with a CPM of size $U = 73$, we have constructed the 511×3577 parity-check matrix $\mathbb{H}_{(3,2^3)}$ of the corresponding EG LDPC block code. A parity-check matrix of the EG-SC code with termination length $L = \theta Y = 56$ is then obtained by lifting the corresponding stair-like diagonal base matrix $\mathbb{B}^{SC}_{(3,2^3)}$ given in Equation (4.9). The resulting parity-check matrix $\mathbb{H}^{SC}_{(3,2^3)}$ of a terminated EG-SC LDPC code is of size 4526×28616. We have $\mathcal{R}\left(\mathbb{H}_{(3,2^3)}\right) = 372$ and $\mathcal{R}\left(\mathbb{H}^{SC_{(3,2^3)}}\right) = 4387$. Thus, the null space of $\mathbb{H}^{SC}_{(3,2^3)}$ defines a $(28616, 24229)$ EG-SC LDPC code with code rate $R = 0.847$, which is only 0.01 away from the design rate of $R_{BC} = 1 - 1/7 = 0.857$ and 0.005 higher than the design code rate $R_{SC} = 0.842$.

We also constructed another EG-SC LDPC codes from the $\mathrm{EG}(3, 2^3)$ code with $K' = 9$. The resulting parity-check matrix $\mathbb{H}^{SC}_{(3,2^3)}$ is of size 4526×36792. Let $L = 7 \times 8 = 56$, then $\mathcal{R}\left(\mathbb{H}^{SC}_{(3,2^3)}\right) = 4387$. Thus, the null space of $\mathbb{H}^{SC}_{(3,2^3)}$ defines a $(36792, 32405)$ EG-SC LDPC code with code rate $R = 0.881$, and it is 0.008 away from the design rate $R_{BC} = 0.889$, and only 0.004 higher than $R_{SC} = 0.877$.

Another pair of EG-SC LDPC codes are constructed from $EG(3, 2^4)$ LDPC block codes. For this geometry, we have a total of $2^{ms} - 1 = 4095$ non-origin points and $M = \left(2^{(m-1)s} - 1\right)(2^{ms} - 1)/(2^s - 1) = 69615$ lines that do not go through the origin. The 69615 lines can be divided into $K = \left(2^{(m-1)s} - 1\right)/(2^s - 1) = 17$ cycle classes, each contains 4095 lines. Since each line has $2^s = 16$ nonzero positions, each cycle class is a weight-16 CPM. Let $\mathbf{H}_{(3,2^4)} = \left[\mathbf{H}^{(0)}_{(3,2^4)}, \mathbf{H}^{(1)}_{(3,2^4)}, \ldots, \mathbf{H}^{(16)}_{(3,2^4)}\right]$

be the parity-check matrix of the EG$(3, 2^4)$ LDPC block code, where each $\mathbf{H}^{(j)}_{(3,2^4)}$ is a weight-16 CPM. Note that the null space of $\mathbf{H}_{(3,2^4)}$ defines a $(69615, 69615 - \mathcal{R}\left(\mathbf{H}_{(3,2^4)}\right))$ LDPC block code with $\mathcal{R}\left(\mathbf{H}_{(3,2^4)}\right) = \mathcal{F}(3, 2^4) - \mathcal{F}(2, 2^4) - 1 = 2801 - 82 - 1 = 2718$ computed from Equation (4.12). Choosing $\theta = 9$, we have $\gamma = 4095/9 = 455$ and $m_s = 8$. Using the proposed construction method, we construct the terminated EG-SC LDPC codes for $K' = 7$ and $K' = 9$ and $Y = 4$. The results are a 20020×114660 parity-check matrix with $K' = 7$ and a 20020×147420 parity-check matrix with $K' = 9$. The rank of the associated parity-check matrix is 18643 for both $K' = 7$ and $K' = 9$. Thus, we obtain a $(114660, 96017)$ EG-SC LDPC code with rate 0.8374 and a $(147420, 128777)$ EG-SC LDPC code with rate 0.8735, respectively.

To evaluate the error performance of the constructed EG-SC LDPC codes and their block code counterparts in the context of storage systems, the binary symmetric channel (BSC) is used. Denote $p < 1$ as the probability for a bit to flip its sign, where p is commonly known as the raw BER (RBER). We assume the decoder knows this channel information and assign a constant magnitude of $\log(1 - p)/p$ to each bit as the decoder input. A maximum iteration number of 500 is used for all simulations. For comparison, the degree distributions of each constructed code and its decoding threshold, denoted by p_{th}, are given in TABLE 4.2. The decoding thresholds are obtained by using the Extrinsic Information Transfer Chart (EXIT) [144] based on their variable node and check node degree distributions. In the table, (d_c, d_r) represents, respectively, the column degree and row degree of a regular LDPC block code, and $(\rho(x), \lambda(x))$ represents, respectively, the distribution of column degrees and row degrees of an SC LDPC code.

	K'	Degree distributions $(\rho(x), \lambda(x))$	p_{th}
SC EG$(3, 2^3)$	7	$\rho(x) = 8$ $\lambda(x) = 0.0323(x^8 + x^{16} + x^{24} + x^{32} + x^{40} + x^{48}) + 0.8062x^{56}$	0.0114
	9	$\rho(x) = 8$ $\lambda(x) = 0.0162(x^9 + x^{10} + x^{20} + x^{22} + x^{30} + x^{31} + x^{41} + x^{42} + x^{50} + x^{52} + x^{62} + x^{63}) + 0.806x^{72}$	0.0083
SC EG$(3, 2^4)$	7	$\rho(x) = 16$ $\lambda(x) = 0.0455(x^{16} + x^{32} + x^{48} + x^{64} + x^{80} + x^{96}) + 0.727x^{112}$	0.0074
	9	$\rho(x) = 16$ $\lambda(x) = 0.0455(x^{16} + x^{32} + x^{48} + x^{64} + x^{80} + x^{96} + x^{112} + x^{128}) + 0.636x^{144}$	0.0059
Proto-SC	7	$\rho(x) = 7$ $\lambda(x) = 0.0364(x^7 + x^{14} + x^{21} + x^{28} + x^{35} + x^{42}) + 0.782x^{49}$	0.0126
	9	$\rho(x) = 7$ $\lambda(x) = 0.0364(x^9 + x^{18} + x^{27} + x^{36} + x^{45} + x^{54}) + 0.782x^{63}$	0.0092
(d_c, d_r)-reg	7	$(8, 56)$	0.0101
		$(16, 112)$	0.0057
	9	$(8, 72)$	0.0074
		$(16, 144)$	0.0043

Table 4.2: Degree distributions and decoding thresholds for EG-SC LDPC codes constructed from EG$(3, 2^3)$ and EG$(3, 2^4)$ with $K' = 7$ and 9, protograph SC LDPC codes with $K' = 7$ and 9 and (d_c, d_r) regular LDPC codes.

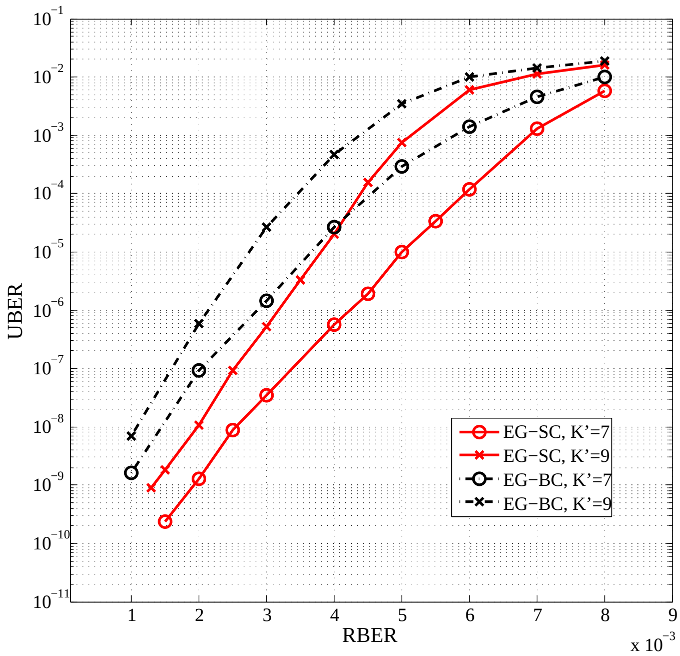


Figure 4.2: UBER of the proposed terminated EG-SC LDPC codes with $K' = 7$ and 9 compared to the 3-dimensional EG$(3, 2^3)$ LDPC block codes.

We first compare the constructed EG-SC LDPC codes with their EG block code counterparts to investigate the convolutional gain. Figs. 4.2 and 4.3 show the UBER of the four EG-SC LDPC codes constructed from Euclidean geometries EG$(3, 2^3)$ and EG$(3, 2^4)$, respectively. In addition, the UBER of each associated $(8, 7K')$-regular EG block code is also shown in the figures. It can be seen from

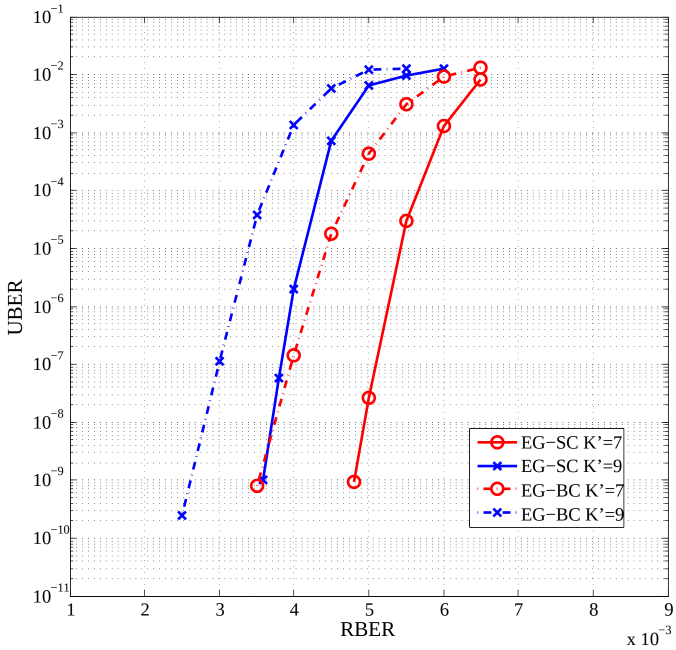


Figure 4.3: UBER of the proposed terminated EG-SC LDPC codes with $K' = 7$ and 9 compared to the 3-dimensional $\mathrm{EG}(3, 2^4)$ LDPC block codes.

Fig. 4.2 that the EG-SC LDPC codes constructed from $\mathrm{EG}(3, 2^3)$ achieve a 0.0012 and 0.0009 convolutional gain in RBER over their EG block code counterparts at the UBER level of 10^{-9} and 10^{-8} for $K' = 7$ and $K' = 9$, respectively. We can also see from Fig. 4.3 that the EG-SC LDPC codes constructed from $\mathrm{EG}(3, 2^4)$ achieve a 0.0012 and 0.0008 convolutional gain in RBER over their EG block code counterparts at the UBER level of 10^{-9} and 10^{-8} for $K' = 7$ and $K' = 9$, respectively.

We then construct protograph SC LDPC codes from $(7, 7K')$ regular protographs, and compare their UBER performance to that of the constructed EG-SC LDPC codes. We choose $K' = 7$ and 9 to make sure that the protograph SC LDPC codes have the same R_{SC} and m_s as the EG-SC LDPC codes constructed from $\mathrm{EG}(3, 2^3)$ with $K' = 7$ and $K' = 9$. The base matrices of the protograph SC LDPC codes are $\mathbf{B} = [7\,7\,7\,7\,7\,7\,7]$ for $K' = 7$ and $\mathbf{B} = [7\,7\,7\,7\,7\,7\,7\,7\,7]$ for $K' = 9$. Each base matrix performs edge spreading [89] such that $\mathbf{B} =$

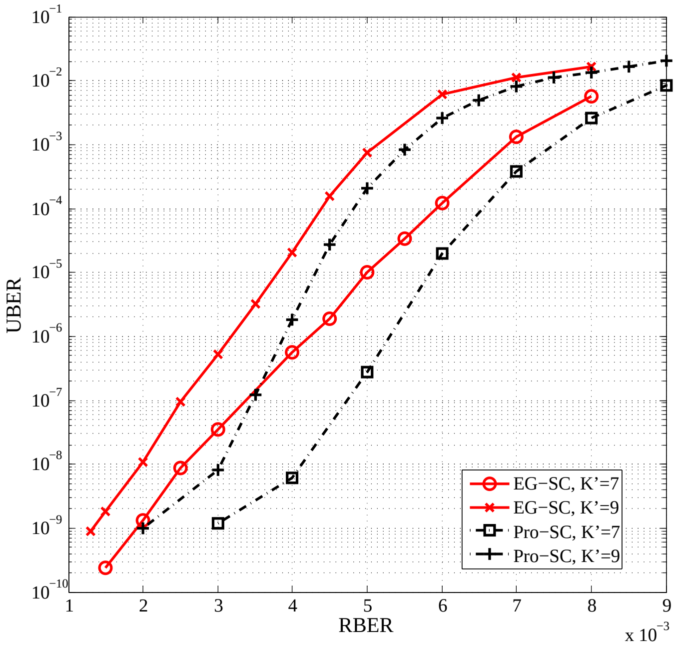


Figure 4.4: UBER of the proposed terminated EG-SC LDPC codes with $K' = 7$ and 9 compared to the UBER of protograph SC codes.

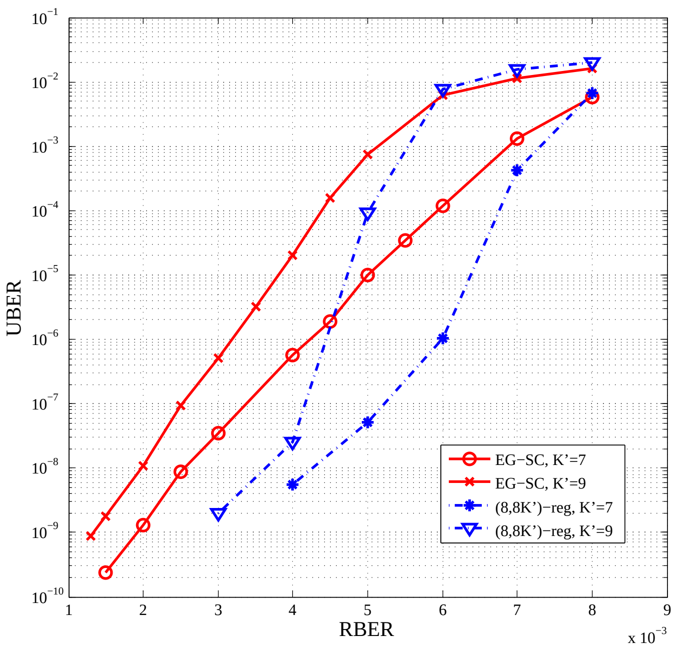


Figure 4.5: UBER of the proposed terminated EG-SC LDPC codes constructed from $\text{EG}(3, 2^3)$ with $K' = 7$ and 9 compared to the UBER of the $(8, 8K')$ regular LDPC codes constructed from $\text{EG}(3, 2^4)$ with $K' = 7$ and $K' = 9$.

$\mathbf{B}^{(0)} + \mathbf{B}^{(1)} + \ldots + \mathbf{B}^{(j)}$ for $j = 6$, where each $\mathbf{B}^{(j)} = [1, 1, \ldots, 1]$ is the $1 \times K'$ all-one vector. Let $L = 49$, by arranging the set of $\{\mathbf{B}^{(j)}\}$ in the form as shown in (3.13), we obtain a terminated base matrix for a protograph SC LDPC code. The base matrix of the terminated protograph SC code is then lifted with CPMs of size $\gamma = 79$, where the cyclic shift of each CPM is chosen such that the resulting $\gamma(L + m_s) \times \gamma LK'$ parity-check matrix contains no cycles of length four. For $K' = 7$ and $K' = 9$, we obtain two parity-check matrices of size 4345×27097 and 4345×34839 with rank 4339 for both matrices. Thus, they define a $(27097, 22758)$ protograph SC code with rate 0.84 and a $(34839, 30500)$ protograph SC code with rate 0.875, respectively. The decoding thresholds for the constructed protograph SC codes are given in TABLE 4.2. The UBER of the constructed protograph SC LDPC codes is shown in Fig. 4.4. We can see from the figure that our proposed EG-SC LDPC codes have better error floor performance than the constructed protograph SC LDPC codes, though they sacrifice some decoding threshold. In particular, the constructed EG-SC LDPC codes do not have an error floor around the UBER of 10^{-9}, while the constructed protograph SC LDPC codes show an error floor around the UBER of 10^{-8}.

Fig. 4.5 compares the UBER of the proposed EG-SC LDPC codes constructed from Euclidean geometry EG$(3, 2^3)$ with $(8, 8K')$ regular LDPC codes of similar code lengths and code rates, where $K' = 7$ and 9. In the figure, the codes denote by '$(8, 8K')$-reg' are LDPC codes constructed from Euclidean geometry EG$(3, 2^4)$. Since each line in $EG(3, 2^4)$ has $2^4 = 16$ nonzero positions, we evenly split each column of the parity-check matrix $\mathbf{H}_{(3,2^4)} = \left[\mathbf{H}^{(0)}_{(3,2^4)}, \mathbf{H}^{(1)}_{(3,2^4)}, \ldots, \mathbf{H}^{(16)}_{(3,2^4)}\right]$ into two different columns. Thus, we obtain 34 weight-8 CPMs. By concatenating 7 and 9 of weight-8 CPMs, a parity-check matrix $\mathbf{H}^{Ext}_{(3,2^4)}$ of size 4095×28665 and 4095×36855 can be constructed, respectively. Both matrices have a column weight of 8, which is identical to the column weight of the parity-check matrix for the EG-SC LDPC codes constructed from EGNote that the code lengths of $(8, 8K')$ regular LDPC codes are similar to that of the EG-SC LDPC codes shown in Fig. 4.5. Moreover, the code rates of $(8, 8K')$ regular LDPC codes are 0.8571 and 0.889 for $K' = 7$ and $K' = 9$, respectively, which are similar to the rates of the EG-SC LDPC codes. It can be seen from Fig. 4.5 that the proposed EG-SC LDPC codes show no error floor at the UBER level of $10^{-9} \sim 10^{-10}$, while the regular LDPC codes show an error floor around the UBER of $10^{-6} \sim 10^{-7}$.

4.7 Reliability-based Windowed Decoding for SC LDPC Codes

By employing the sliding windowed decoder with the conventional WBF algorithm for SC LDPC codes, we observe a considerable performance loss caused by error floor. In this section, we propose a partial message reservation (PMR) method and a partial syndrome check (PSC) stopping rule for the windowed decoder to solve this problem.

4.7.1 The PMR Method

Due to the structure of the sliding windowed decoder, we observe that some of the VNs in the decoding window have neighboring CNs outside the window. We call these VNs as *incomplete* VNs and the others as *complete* VNs for this decoding window. It was shown in [74] that the performance of the conventional WBF algorithm highly relies on a large column weight of the given parity-check matrix for an LDPC code. However, in the construction defined by Eq. (3.15), the incomplete VNs have a lower column weight than that of complete VNs.

Therefore, the messages passed along the edges connected to the incomplete VNs are less reliable than that associated with the complete VNs.

It is well-known that the good performance of SC LDPC codes with windowed decoding comes from reliable messages passed from one window to the next. To avoid the error propagation of unreliable messages from the incomplete VNs, we propose a PMR method for the sliding windowed decoder. Let $\mathbb{V}_C$ and $\mathbb{V}_I$ represent the sets of indices for complete VNs and incomplete VNs in a decoding window, respectively. Define $\mathbf{z}_t = (z_{t,0}, z_{t,1}, \ldots z_{t,n'-1})$ as the decoded codeword for the current window at time index t, where $n' = \tilde{W} \cdot Uc$. The outgoing message from the k-th VN in the current decoding window to the next window can be given by

$$u_k = \begin{cases} z_{t,k}, & k \in \mathbb{V}_C \\ v_k, & k \in \mathbb{V}_I \end{cases}, \tag{4.20}$$

where $0 \leq k \leq n' - 1$. This means that only the messages from complete VNs in the t-th window are reserved for the $t+1$-th window. Note that the window size is chosen to ensure that the number of complete VNs in a decoding window is larger than that of incomplete VNs so that more reliable messages can be reserved and propagated to the next window. We will show in Fig. 3 that this PMR method can significantly improve the error floor performance of the proposed RBWD scheme.

4.7.2 The PSC Stopping Rule

When decoding an LDPC code, all the parity-check equations need to be satisfied to get a valid codeword. An efficient stopping rule based on soft bit error indicators was introduced in [96] for sliding windowed decoder. However, this method can not be directly applied to an RBWD scheme since only hard information is

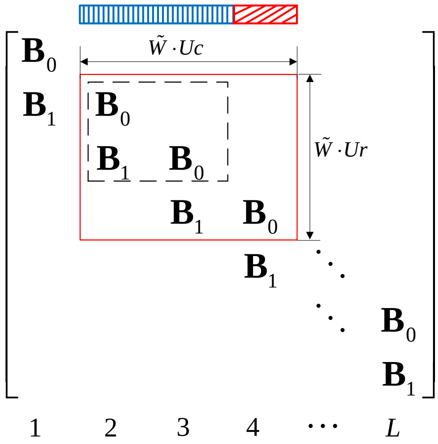


Figure 4.6: An example of sliding windowed decoder with window size $\tilde{W} = 3$ at time index $t = 2$ (solid region). The parity-check equations considered by the PSC stopping rule are shown in dashed region. The complete VNs are shown in blue (vertically hatched) and the incomplete VNs are shown in red (hatched) above the parity-check matrix.

passed along the edges in the Tanner graph.

By making use of the reliable messages, a PSC stopping rule is applied to the windowed decoding scheme. In particular, our stopping rule only focuses on the parity-check equations of complete VNs in a decoding window. To be specific, define $\tilde{W}'$ as the number of parity-check equations in one decoding window considered by the PSC stopping rule. The first $\tilde{W}' = (\tilde{W} - m_s) \cdot Ur$ parity-check rows are checked in each decoding window. Once these parity-check equations are satisfied or the preset maximum number of iterations is reached, the decoding window slides to the next position. Note that a PSC stopping rule is also proposed in [95]. However, it only checks a fixed number of syndromes that belong to the target symbols. In our proposed PSC stopping rule, all reliable VNs are considered. When $\tilde{W} > m_s + 1$, the number of complete VNs in one decoding window is larger than that of target symbols, which leads to a more strict stopping rule and ensures the messages from complete VNs to be more reliable.

An example of the proposed sliding windowed decoder with $\tilde{W} = 3$ and $m_s = 1$ at time index $t = 2$ is illustrated in Fig. 4.6. The first $2Uc$ VNs are complete VNs and the last Uc VNs are incomplete VNs. Note that only the first $2Ur$ CNs are considered for the parity-check equations since these CNs connect to the complete VNs. The last updated messages of the first $2Uc$ complete VNs are reserved for the decoding process at time index $t = 3$.

4.7.3 The Proposed RBWD Scheme

Denoted by $\hat{\mathbf{H}}$ an $m' \times n'$ parity-check matrix for one decoding window, where $m' = \tilde{W} \cdot Ur$ and $n' = \tilde{W} \cdot Uc$. Let $\mathbf{s}'^{(l)} = (s'^{(l)}_0, s'^{(l)}_1, \ldots s'^{(l)}_{\tilde{W}'-1})$ be the syndrome vector computed by the PSC stopping rule at the l-th iteration. Assume that vector $\mathbf{y}_t^{(l)} = (y_{t,0}^{(l)}, y_{t,1}^{(l)}, \ldots y_{t,n'-1}^{(l)})$ is the decoded codeword of the l-th iteration at time index t. Define $\mathcal{M}'(j')$ and $\mathcal{N}'(i')$ as the sets of indices of all the nonzero elements in the j'-th row and i'-th column of $\hat{\mathbf{H}}$, respectively. Set the maximum number of decoding iterations as $I_{\max}$. By combining the PMR and PSC with the SBF-WBF algorithm, the proposed RBWD scheme is summarized in **Algorithm 4.2**.

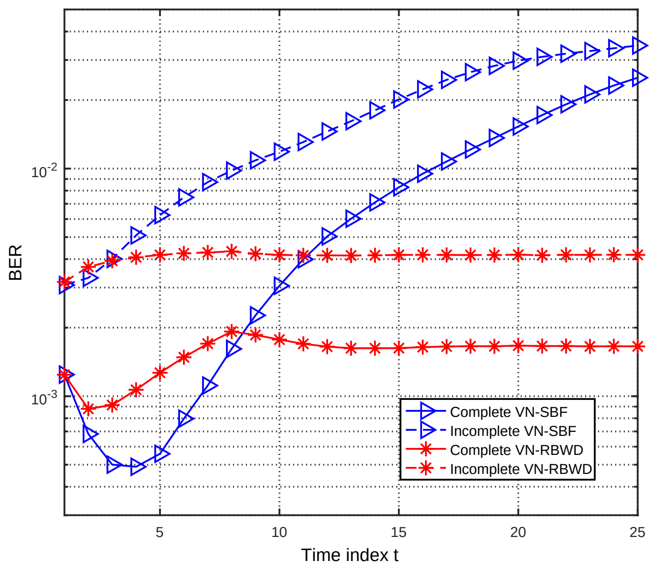


Figure 4.7: Error performance of complete and incomplete VNs for the $(7, 49)$ SC LDPC code at different time t, $E_b/N_0 = 6$ dB.

Algorithm 4.2 The proposed RBWD scheme

Inputs: $\hat{\mathbf{H}}, L, \tilde{W}, U, I_{\max}$

Initialize: $l = 0$ and $t = 1$
while $t \leq L$ **do**
  if $t = 1$ **then**
    set $\mathbf{y}_1^{(0)} = \mathbf{v}$
  else
    set $y_{t,i'}^{(0)} = \begin{cases} v_{i'}, & n' - 1 - Uc \leq i' < n' - 1 \\ u_{i'}, & 0 \leq i' < n' - 1 - Uc \end{cases}$
  end if
  while $l \leq I_{\max}$ **do**
    for $j' = 0 : (m' - 1)$ **do**
      $w_{j'} = \min_{i' \in \mathcal{M}(j')} |r_{i'}|$
    end for
    Update $l = l + 1$
    Compute $s^{(l)}$ by $\mathbf{y}_t^{(l)} \hat{H}^T$
    Determine $s'^{(l)} = (s_0^{(l)}, s_1^{(l)}, \dots s_{\tilde{W}'-1}^{(l)})$
    if $s'^{(l)} = \mathbf{0}$ or $l = I_{\max}$ **then**
      output $\mathbf{z}_t = \mathbf{y}_t^{(l)}$ and **break**
    end if
    for $i' = 0 : (n' - 1)$ **do**
      Estimate $E_{i'}^{(l)}$ as in (2)
    end for
    Update $\mathcal{F}$ as in (3)
    Flip $y_{t,i'}^{(l)}$ randomly, where $i' \in \mathcal{F}$
  end while
  Perform PMR as in (4), set $t = t + 1$ and $l = 0$
end while

Note that the performance gain of the proposed RBWD scheme originates from the discarding of unreliable messages from previous decoding window to perform message-passing decoding in the current window. To demonstrate this, we evaluate the BER performance of complete and incomplete VNs for various window positions. Fig. 4.7 depicts the BER for both complete and incomplete VNs of an SC LDPC code constructed from a $(7, 49)$-regular protograph LDPC code with $m_s = 6$, $L = 56$ at different window positions. The decoding window size $\tilde{W}$ is set to 14. It can be seen that for both the SBF-WBF algorithm and the proposed RBWD decoding scheme, incomplete VNs always have a higher BER than complete VNs. For instance, the BER of incomplete VNs by using RBWD scheme is nearly two times as that of the complete VNs for all window positions. The difference of BER for those two types of VNs can be even larger for the SBF-WBF algorithm. This indicates that the messages from incomplete VNs are less reliable than that from complete VNs.

It can also be seen that the BER for both complete and incomplete VNs of the SBF-WBF algorithm increases with the sliding of the decoding window. For example, the BER of complete and incomplete VNs for the SBF-WBF algorithm increases more than ten times from the first decoding window to the 25-th decoding window for the simulated SC LDPC code. On the other hand, the BER of both complete and incomplete VNs for the proposed RBWD scheme remains almost the same for all decoding window positions. This means that by only reserving the messages from the complete VNs in the RBWD scheme, we prevent the "contamination" of the reliable messages from the unreliable messages. As a result, the BER performance of the proposed decoding scheme is substantially improved.

4.8 Performance Analysis of the RBWD Scheme

In this section, we investigate the error rate performance and the decoding complexity of the proposed RBWD scheme. Binary phase-shift keying (BPSK) modulation and additive Gaussian noise channels are considered in all simulations. The maximum number of iterations is 200 for all windowed decoding schemes and it is 2000 for FBD.

4.8.1 Error Rate Performance

An SC LDPC code is constructed from a $(7, 49)$-regular protograph LDPC code with full edge-spreading, i.e., $\mathbf{B}_0 = \mathbf{B}_1 = \cdots = \mathbf{B}_6 = [1\ 1\ \cdots\ 1]_{1\times 7}$. We set the termination length $L = 56$, the resultant base matrix $\mathcal{B}_L$ is expanded with lifting size $U = 97$. As a result, we obtain a length-38024 $(7, 49)$ SC LDPC code with large VN degrees. The BER and frame error rate (FER) of the length-38024 $(7, 49)$ SC LDPC code decoded by various decoding schemes are shown in Fig. 4.8. Here MBF-PMR and SBF-PMR represent the RBWD scheme without applying the proposed PSC stopping rule. The BER and FER of the FBD and the sliding windowed decoder based on the SBF-WBF algorithm are also shown in the figure for comparison. We see that the proposed PMR method dramatically improves the error rate performance. Moreover, the proposed stopping rule further reduces the error floor and achieves the BER performance within 0.1 dB from that of the FBD.

Note that the proposed RBWD scheme also works for SC LDPC codes with small VN degrees. To clarify the generality, we constructed an SC LDPC code from a $(3, 6)$-regular protograph LDPC code. After applying the edge spreading matrices $\mathbf{B}_0 = \mathbf{B}_1 = \mathbf{B}_2 = [1\ 1]_{1\times 2}$ and set the termination length $L = 108$, a length-38016 $(3, 6)$ SC LDPC code can be obtained by graph expansion with lifting size $M = 176$. As shown in Fig. 4.9, the proposed RBWD scheme works for SC LDPC codes with small VN degrees in the sense that the BER performance of the RBWD scheme can approach that of the FBD.

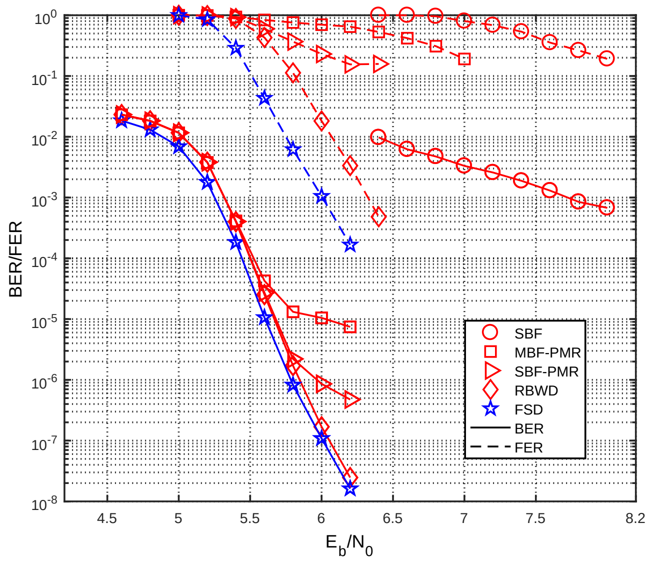


Figure 4.8: BER/FER performance of the length-38024 $(7, 49)$ SC LDPC code. The window size is $\tilde{W} = 14$ for windowed decoding.

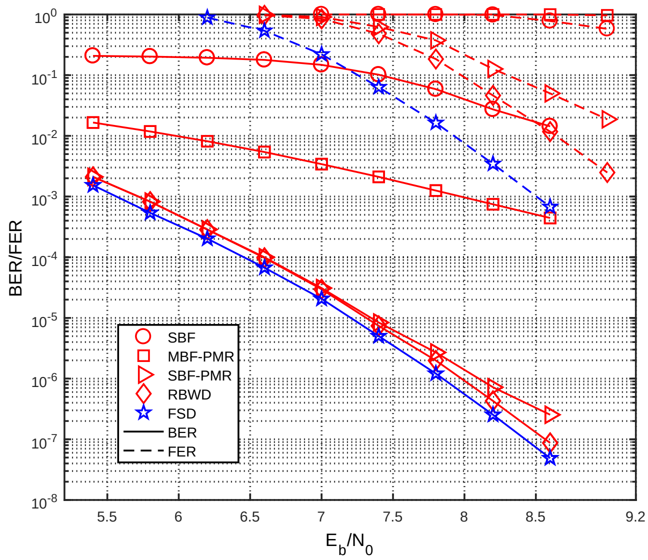


Figure 4.9: BER/FER performance of the length-38016 $(3, 6)$ SC LDPC code. The window size is $\tilde{W} = 6$ for windowed decoding.

4.8.2 Complexity Comparison

In this section, we compare the complexity of the proposed RBWD scheme with that of the MBF-PMR and the SBF-PMR schemes. Note that we only consider the decoding schemes based on the conventional WBF algorithm since it only exchanges one bit information between CNs and VNs, which has a lower decoding complexity than SPA. In addition, we define I_{avg} as the average number of updates processed by a VN in one decoded codeword, which can be given by

$$I_{avg} = \left(\sum\nolimits_{t=1}^{L} I_t\right) / L, \tag{4.21}$$

where I_t is defined as the total number of updates processed by a VN at the t-th window during the decoding process. The comparison of I_{avg} for the length-38024 $(7, 49)$ SC LDPC code decoded by MBF-PMR, SBF-PMR, and the proposed RBWD scheme is shown in Table 5.14. Note that for a fair comparison, we fix $W = 14$ for all windowed decoding schemes, i.e., each decoding window covers 9506 bits in order to keep the same decoding latency. It can be seen that for SNR from 5.6 dB to 6 dB, our proposed RBWD scheme requires about half number of updates compared to that of MBF-PMR and SBF-PMR.

Table 4.3: Average iteration comparison of the length-38024 $(7, 49)$ SC LDPC code decoded by various decoding schemes.

E_b/N_0 (dB)		5.2	5.4	5.6	5.8	6	6.2
I_{avg}	MBF-PMR	132	59	29	21	17	13
	SBF-PMR	132	57	29	22	18	15
	RBWD	129	48	17	11	8	7

4.9 Summary

In this chapter, we first proposed a construction method for SC LDPC codes from the conventional EG LDPC codes. The proposed construction method yields families of EG-SC LDPC codes by considering m-dimensional EG LDPC block codes as the underlying component codes, where $m > 2$. Various design examples for EG-SC LDPC code are illustrated, which are decoded by a low complexity WBF algorithm. We show that the proposed EG-SC LDPC codes achieve considerable convolutional gain over their EG LDPC code counterparts. Compared to protograph SC LDPC codes and regular LDPC codes of similar code lengths and code rates, the EG-SC LDPC codes show no error floor at the UBER as low as 10^{-9}, whereas the protograph SC LDPC codes and regular LDPC codes show an error floor around the UBER level of 10^{-9} and 10^{-7}, respectively. In addition, a RBWD scheme is also proposed for SC LDPC codes. The proposed scheme propagates the reliable messages from complete VNs between two consecutive decoding windows, which substantially improves the BER performance of the RBWD scheme within only 0.1 dB away from that of the FBD. The stopping rule adopted in the RBWD scheme further reduces the error floor by operating on the parity-check equations that only involve complete VNs. Numerical results show that the PSC method can reduce decoding complexity of the SC LDPC codes without error floor compared to the MBF-PMR and the SBF-PMR schemes.

Chapter 5

Enhanced Quasi-Maximum Likelihood Decoding for 5G LDPC Codes

5.1 Introduction

Low-density parity-check (LDPC) codes [70] and their variations, e.g., [46, 86, 89, 91, 92, 133, 161–164], have been proposed to wide applications, such as wireless communications, optical communications, and data storage, for their near-capacity performance under belief propagation (BP) decoding algorithms for moderate to long block lengths. In the coming fifth generation (5G) mobile networks, protograph-based raptor-like (PBRL) LDPC codes [133] are determined to be one of the channel coding technologies for the eMBB scenario [1]. Therefore, how to develop efficient and effective decoding algorithms with low complexity and good error performance for the 5G LDPC codes becomes an attractive research question recently. In this chapter, we aim to improve the performance of the

5G LDPC codes approaching that of their associated maximum-likelihood decoding. To be more specific, we firstly propose a two-dimensional scale-corrected min-sum algorithm for the 5G LDPC codes to approach the error performance of the sum-product algorithm. Then we propose an enhanced reprocessing method, so-called the enhanced quasi-maximum likelihood (EQML) decoding method, which further approaches the performance of the maximum-likelihood decoding for 5G short LDPC codes.

5.2 Problem Statement

As we move into the era of the fifth generation (5G) mobile networks, new services and applications, such as vehicle-to-everything and Internet of things, have been considered in the 5G standard recently. These applications require ultra-reliability and low decoding complexity for massive low-cost devices, which have drawn significant research efforts for short codes with low code rates [165–167]. Compared to the asymptotic performance for long LDPC codes, the performance limits in finite block lengths are more important for these applications [129,168]. Although the maximum likelihood (ML) decoding becomes possible for very short LDPC codes, its decoding complexity becomes prohibitively high when the block length is larger than a few tens of bits. To reduce the decoding complexity, there have been extensive studies on simplified decoding architectures for LDPC codes and their variations, such as [90] and [169]. However, they are commonly designed for long block lengths. On the other hand, it is shown in [106] that the sum-product algorithm (SPA) becomes sub-optimal and has considerable performance gap from the ML decoding for the LDPC codes with short block lengths, which is due to the existence of the small cycles in their associated Tanner graphs. Therefore, several quasi-ML (QML) decoding methods have been investigated [107–110] to achieve near-ML performance with a tolerable decoding complexity for the LDPC codes with short block lengths. The most common strategy adopted in these works is to introduce multiple rounds of decoding tests, so-called the reprocessing [109], after the failure of conventional BP decoding. More specifically, the decoder is reinitialized with a list of different decoder inputs during the reprocessing, where each input sequence is generated by substituting the channel outputs of the selected unreliable VNs with the maximum or minimum values. The conventional BP decoding is conducted with each input sequence, and the decoding output is stored if it generates a valid codeword. The 'best' codeword is chosen from the list of valid codewords as the decoder output according to a certain decision metric.

One of the earliest QML decoding method is the ordered statistic decoding (OSD) [107]. For a received signal, this decoding method chooses p most reliable bits and generates a list of $\sum_{q=1}^{p} \binom{p}{q}$ possible error combinations for these bits. Each combination is re-encoded into a codeword. In this way, a list of codewords is obtained, and one of them is chosen as the decoding output. Obviously, matrix transformation is required for different p most reliable bits in order to obtain the corresponding generator matrix, which makes the OSD unsuitable for hardware implementations. To avoid matrix transformation, the augmented BP (ABP) decoding and saturated min-sum (SMS) decoding were proposed in [108] and [109], respectively. In addition, a single bit flip-aided decoding method based on multitree search for the unreliable VNs was proposed in [110]. For these decoding methods, a conventional BP decoding is evoked for several rounds to generate a list of codewords, where the decoder is fed with a different input sequence derived from the same channel output sequence at each round of the decoding test. Nevertheless, for all these QML decoding methods with a small list size, the performance gap to the ML decoding is still considerable.

5.3 Main Contributions

In this work, to obtain the decoding performance near the SPA for the 5G LDPC codes, we propose an improved MSA-based decoding algorithm with a low decoding complexity compared to the SPA, where we only introduce one pair of universal parameters for all information bit lengths $K \in [40, 8448]$ and code rates $R \in [1/5, 8/9]$ in the 5G standard. To further approach the error performance of the ML decoding for the 5G short LDPC codes, we propose an enhanced QML decoding method with a new reprocessing architecture, which obtains a near-ML performance with a small list size. The main contributions of this work are summarized below:

- We propose an improved MSA-based decoding algorithm for the 5G LDPC codes to achieve the error performance of the SPA. The proposed decoding algorithm adopts the self-correction method, which reserves the reliable variable-to-check (V2C) messages and reduces the sign flips. Moreover, based on the NMSA, we use an additional scaling factor amplifies the magnitude of the extrinsic messages sending to the CNs that are connected to degree-1 VNs, which further improves the reliability of the reserved messages with the self-correction method.

- We propose an enhanced decoding architecture based on the proposed decoding algorithm such that the error performance of the 5G short LDPC codes can approach that of ML decoding with a desirable decoding complexity even with a small list size. By investigating the phenomenon of sign flips for VNs' extrinsic messages, we propose a novel node selection method from edge perspective to improve the accuracy of selecting unreliable VNs. To reduce the decoding complexity, we further introduce a stopping rule, namely partial pruning stopping (PPS) rule. The PPS rule effectively terminates the decoding tests on part of reprocessing sub-branches if a valid codeword is obtained.

- We derive the asymptotic bounds on frame error rate (FER) of the proposed EQML decoding method for a given maximum number of reprocessing stages and then approximate the lower bounds on FER by using a semi-analytical method. The derived lower bound on the FER is close to the simulation results in the low signal-to-noise ratio (SNR) region.

- We investigate the error performance of the proposed decoding algorithm and the EQML decoding method. It demonstrates that the proposed decoding algorithm can approach the error performance of the SPA for the 5G LDPC codes with small and large information bit lengths. Moreover, the EQML decoding method outperforms the SPA with the same decoding complexity for the 5G short LDPC codes and can approach the Polyanskiy-Poor-Verdú (PPV) bound within 0.4 dB in terms of FER performance. We also analyze the decoding complexity of the proposed EMQL decoding method with different stopping rules by using the average number of iterations for obtaining one decoded codeword. Simulation results show that the PPS rule has a lower decoding complexity compared to the list decoding stopping (LDS) rule in [108] without sacrificing the error performance.

5.4 The Two-Dimensional Scale-Corrected MSA Decoding for 5G LDPC Codes

In this section, we propose an improved MSA-based decoding algorithm, so-called the two-dimensional scale-corrected (2D-SC) MSA, to approach the error performance of the SPA for the 5G LDPC codes. We can see from (3.11) in Chapter 3 that there is a large amount of degree-1 VNs in the parity-check matrix compared to the LDPC codes in other standards such as the IEEE 802.11n and the IEEE 802.16e standard. This results in a severe deterioration in error performance when the conventional BP decoding algorithms are used.

To improve the MSA-based decoding algorithms, the evolution of the sign and magnitude for the V2C messages is utilized to facilitate the propagation of the reliable V2C messages. In [50], the sign flips of the V2C messages are explored to avoid the propagation of the unreliable V2C messages. In the following, we further investigate the sign fluctuation phenomenon for the 5G LDPC codes and propose a partial self-correction method, which reserves the reliable V2C messages via significantly reducing the number of sign flips for the V2C messages sent from VNs in H_{core}. In addition, we amplify the magnitude of the V2C messages sent to the CNs in H_{ex} by a scaling factor larger than 1 to increase their reliability. Moreover, a pair of universal scaling factors (α, β) for the 5G LDPC code families is applied to all C2V messages and the V2C messages sent to the CNs in H_{ex}, respectively, such that the error performance of the proposed decoding algorithm is close to that of the SPA for all information bit lengths $K \in [40, 8448]$ and code rates $R \in [1/5, 8/9]$ in the 5G standard.

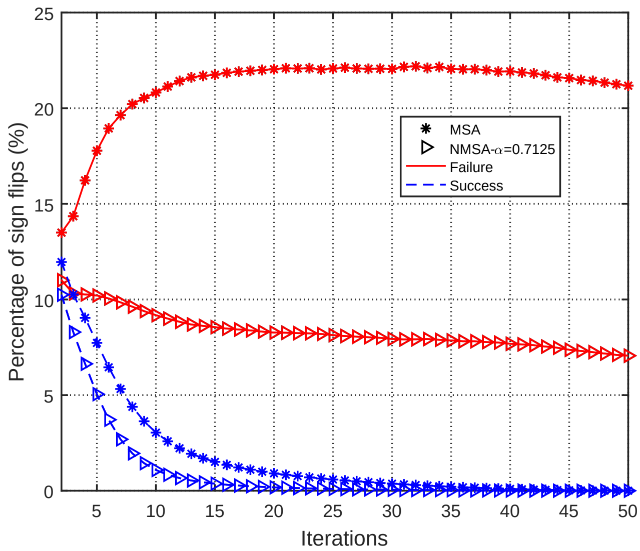


Figure 5.1: The percentage of sign flips for the V2C messages per iteration with the $R = 1/5$, $K = 120$ 5G LDPC code under AWGN channels. $E_s/N_0 = -2.6$ dB.

5.4.1 Sign Flips Reduction

At each iteration of the BP decoding algorithm, the reliability of the V2C messages sent from the VNs in H_{core} plays an important role in successful decoding compared to the degree-1 VNs since these degree-1 VNs always send their initial channel output as the V2C messages. Fig. 5.1 shows the percentage of sign flips for the V2C messages sent from the VNs in H_{core} per iteration. We can see that the percentage of sign flips behaves in completely different ways for successful and unsuccessful decoding cases. For the successful decoding case, the percentage of sign flips for both MSA and NMSA decoding diminish to zero with the number of iteration increases. Note that the percentage of sign flips decreases faster for the NMSA decoding compared to the MSA decoding, which attributes to the mitigation of the overestimation by the scaling factor in the NMSA. In contrast, this percentage saturates to a constant value for the MSA decoding after a rapid increase in the first few iterations for the unsuccessful decoding case. A similar tendency is also observed by using the NMSA decoding for the unsuccessful decoding case. When the NMSA decoding is adopted, although the percentage of sign flips for the V2C messages reduces slightly at the first few iterations, it saturates to a much high constant value compared to the case of successful decoding. Therefore, we have the conjecture that there exists a high percentage of the sign flips for the V2C messages when a decoding failure occurs.

To reduce the number of sign flips during the decoding iterations, we propose a partial self-correction (PSC) method for the V2C messages sent from the VNs in H_{core}. Define N_{core} as the number of VNs in H_{core}, the self-correction is performed as

$$Z_{nm}^{(i)} = \begin{cases} Z_{nm}^{\text{tmp}}, & \text{if } \operatorname{sign}(Z_{nm}^{\text{tmp}} \cdot Z_{nm}^{(i-1)}) > 0, 1 \leq n \leq N_{\text{core}} \\ 0, & \text{otherwise} \end{cases}, \tag{5.1}$$

where Z_{nm}^{tmp} is the temporary V2C messages computed by Eq. (3.18). Note that the proposed PSC method compares the signs of the V2C messages sent from the VNs in H_{core} at the current iteration with that at the previous iteration, and the associated V2C message is only sent to its neighboring CN if the two signs are the same. For the V2C messages with their signs contradictory with that in the previous iteration, they are set to zero by the self-correction method. As shown in the later section, the PSC method can reduce the number of sign flips efficiently and improve the decoding performance.

5.4.2 Message Amplification

Note that the C2V messages sent from the CNs in H_{ex} is upper bounded in magnitude by the channel output of their associated degree-1 VN when the SPA is adopted [105]. This phenomenon also happens when the conventional MSA or NMSA decoding is employed. Assume that a CN c_m in H_{ex} is connected to a VN $v_{\hat{n}}$ of degree 1. At the i-th iteration, for any v_n that is connected to c_m and $deg(v_n) \neq 1$, we have

$$\min_{v_{n'} \in \mathcal{H}(c_m) \backslash v_n} \left| Z_{n'm}^{(i-1)} \right| \leq \left| Z_{\hat{n}m}^{(i-1)} \right| = \left| L(v_{\hat{n}}) \right|. \tag{5.2}$$

It demonstrates a bounded effect that the magnitude of the C2V messages from c_m to any of its neighboring VNs except $v_{\hat{n}}$ is not greater than the magnitude of the V2C message sent from $v_{\hat{n}}$. Since $v_{\hat{n}}$ is a degree-1 node, we have $|Z_{\hat{n}m}| = |L(v_{\hat{n}})|$. We know that the reliability of a message can be indicated by the magnitude of its channel output. Thus, the reliability of the C2V messages sent from the CNs in H_{ex} highly relies on the magnitude of the channel output received on the degree-1 VNs. Obviously, the reliability of the C2V messages sent from a CN in H_{ex} to their neighboring VNs in H_{core} could be degraded if its connected degree-1 VN receives the channel output with a small magnitude. Moreover, the impact of this effect could be more severe when errors occur on the degree-1 VNs because the erroneous VNs usually have small channel outputs.

We further explore the evolution of the V2C messages for the conventional MSA and SPA to verify the above conjecture. Fig. 5.2 presents the percentage of the V2C messages sent to the CNs in H_{ex}, which is smaller in magnitude by using the conventional MSA than the SPA. We see that there is a high percentage of the V2C messages obtained by the conventional MSA decoder being smaller than that of SPA decoder in magnitude at the first few iterations. With the iteration increases, this value decreases dramatically to zero for the case of successful decoding while it gradually increases to almost 100% for unsuccessful decoding cases. This indicates that the reliability of the V2C messages sent to the CNs

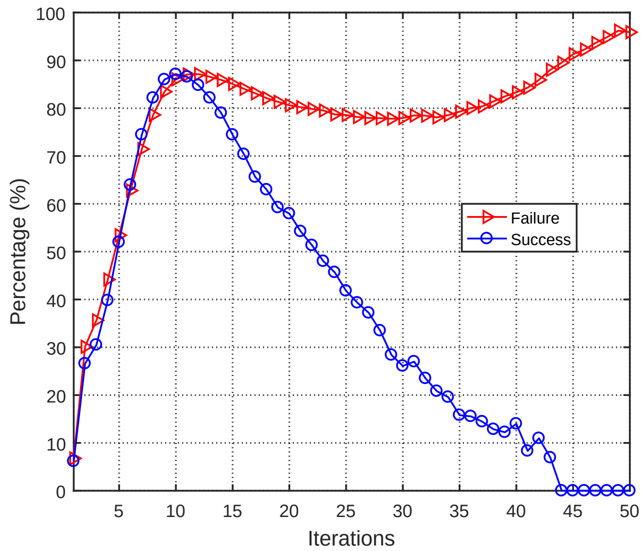


Figure 5.2: Percentage of V2C messages sent to the CNs in H_{ex}, which is smaller than that of SPA. $R = 1/5$, $K = 120$ and $E_s/N_0 = -2.6$ dB.

in H_{ex} decreases with iterations increase for unsuccessful decoding cases. This is because the associated C2V messages with bounded magnitude in the previous iterations counteract the propagation of the reliable messages.

To reduce the disadvantage of the bounded effect, the upper bound $|L(v_{\hat{n}})|$ need to be enlarged properly for these VNs. Therefore, we introduce a scaling factor β, where $\beta > 1$, to amplify the magnitude of the V2C messages sent to the CNs in H_{ex}. In this way, the V2C messages can be computed by

$$Z_{nm}^{\text{tmp}} = \begin{cases} \beta \cdot [L(v_n) + \sum\limits_{c_{m'} \in \mathcal{H}(v_n) \backslash c_m} Y_{m'n}^{(i)}], & M_{\text{core}} + 1 \leq m \leq M \\ \\ L(v_n) + \sum\limits_{c_{m'} \in \mathcal{H}(v_n) \backslash c_m} Y_{m'n}^{(i)}, & \text{otherwise} \end{cases}, \tag{5.3}$$

where M_{core} is defined as the number of CNs in H_{core}. As a result, the upper bound of the V2C messages sent to the CNs in H_{ex} is linearly scaled by β as

$$\min_{v_{n'} \in \mathcal{H}(\hat{c_m}) \backslash v_n} \left| M_{n'm}^{(i-1)} \right| \leq \left| Z_{n'm}^{(i-1)} \right| = \beta \cdot |L(v_{\hat{n}})| \,, \tag{5.4}$$

5.4.3 The 2D-SC MSA

Set the maximum number of iterations as $I_{\max}$. By combining the PSC method and the message amplification, we propose the 2D-SC MSA, which can be summarized in **Algorithm 5.1**.

Algorithm 5.1 The 2D-SC MSA

Initialize:
  $i = 0$
  For each n, m, set $Z_{nm}^{(0)} = L(v_n)$
Iterations:
while $i \le I_{\max}$ **do**
  $i = i + 1$
  for $m = 1 : M$ and each $v_n \in \mathcal{H}(c_m)$ **do**
    Update $Y_{mn}^{(i)}$ as in Eq. (3.21)
  end for
  for $n = 1 : N$ and each $c_m \in \mathcal{H}(v_n)$ **do**
    Compute Z_{nm}^{tmp} as in Eq. (5.3)
    Update $Z_{nm}^{(i)}$ as in Eq. (5.1)
  end for
  for $n = 1 : N$ **do**
    Compute $L_n^{(i)}$ as in Eq. (3.19)
    if $L_n^{(i)} < 0$ **then**
      $x_n = 1$
    else
      $x_n = 0$
    end if
  end for
  if $\mathbf{x} \cdot H^T = 0$ or $i = I_{\max}$ **then**
    Stop and output $\mathbf{x}$
  end if
end while

It is notable that the scaling factor $\alpha \in (0, 1)$ used for the C2V messages is the same as Eq. (3.21) in the NMSA. Because the NMSA demonstrates a rapid decreasing in the number of sign flips as shown in Fig. 5.1, we expect the decoding algorithm based on the NMSA to achieve a fast convergence speed.

In our proposed 2D-SC MSA, the PSC operation tries to retain the reliable V2C messages in each iteration based on their sign flips behavior. Then these reliable V2C messages sent to the CNs in H_{ex} are amplified by the scaling factor β. Since the C2V messages sent from the CNs in H_{ex} are directly affected by the vulnerable degree-1 VNs, enlarging the magnitude of these reliable V2C messages can further improve the reliability of these C2V messages computed in the next iteration, which is beneficial to the decoding performance of the 5G LDPC codes.

Note that different scaling factors are applied to the conventional MSA to decode irregular LDPC codes with long block lengths in [49] when computing both C2V and V2C messages. However, the scaling factors used for the calculation of the V2C messages in [49] are optimized within $(0, 1)$ by considering the degree differences among the adjacent VNs, which aims to further reduce the overestimation effect. For our proposed 2D-SC MSA, the scaling factor β aims at amplifying the reliable V2C messages sent to the CNs in H_{ex} for the next decoding iteration, which means that the initial channel values in Eq. (3.18) are also amplified if the associated V2C messages are considered to be reliable.

5.4.4 Comparison of FER Performance

We investigate the FER performance of the proposed 2D-SC MSA for the 5G LDPC codes with the details listed in Table 5.1 [1]. Quadrature phase-shift keying modulation and an AWGN channel are considered in the floating-point simulations. We set $I_{\max} = 50$. The scaling factor pair (α, β) is optimized as $(0.75, 1.25)$ for the 2D-SC MSA[1], which can obtain nearly the optimal performance for the 5G LDPC codes with all information bit lengths and code rates.

[1]The scaling factors here is searched to minimize the minimum mean-square error gap between the FER performance of the 2D-SC MSA and the SPA.

CodeType	Information Bit Lengths K	Code Rate R
BG1	{56, 96, 120, 176, 216, 256, 288, 320, 336, 376, 416, 576, 752, 912, 1088, 1248, 1728, 2688, 3840}	{1/5, 1/3, 2/5, 1/2, 2/3}
BG2	{4096, 5184, 8448}	{1/3, 2/5, 1/2, 2/3, 8/9}

Table 5.1: The Table of Simulated 5G LDPC Codes

Fig. 5.3 depicts the FER performance of the 5G LDPC codes with $K = 120$ and various code rates decoded by the proposed 2D-SC MSA and compared to that of the SPA, NMSA, and MSA. We can see that the FER performance of the MSA is far away from that of the SPA for code rates $R = 1/5, 1/3, 2/5, 1/2,$ and $2/3$. Although the NMSA has nearly the same FER performance as the SPA for high code rates, i.e., $R = 1/2, 2/3$, it has a significant performance gap from the SPA for low code rates $R = 1/5, 1/3, 2/5$. Nevertheless, our proposed 2D-SC MSA has nearly the same FER performance as that of the SPA for Code ensemble 1 with all simulated code rates.

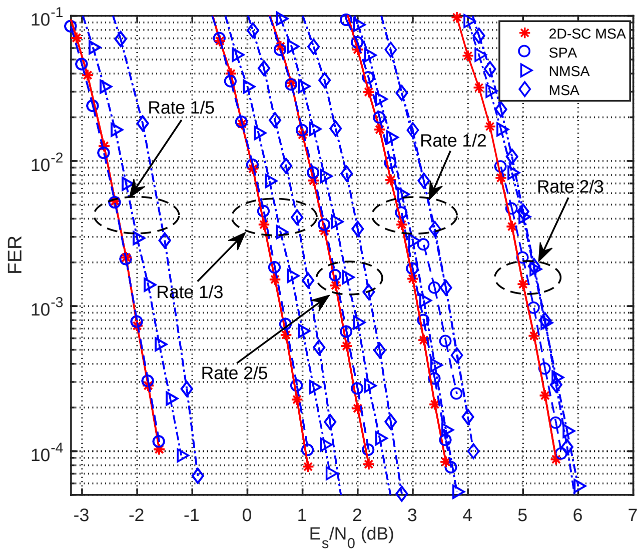


Figure 5.3: FER performance of the 5G LDPC codes decoded by different decoding algorithms with information bit lengths $K = 120$ and code rate $R = 1/5, 1/3, 2/5, 1/2, 2/3$.

The FER performance of the 5G LDPC codes with $K = 8448$ and various code rates, where $K = 8448$ is the maximum information bit lengths supported by the 5G LDPC codes, is shown in Fig. 5.4. It demonstrates that the performance gap between the MSA and the SPA is even more significant, particularly for low code rates, i.e., $R \leq 1/2$. Similarly, the FER performance of the NMSA is about $0.35 \sim 0.4$ dB away from the SPA for $R = 1/3, 2/5$ and $1/2$. However, the proposed 2D-SC MSA is less than 0.2 dB away from the SPA performance for $R \geq 1/2$, and it can approach the FER performance of the SPA within 0.3 dB for code rates $R = 1/3$ and $2/5$.

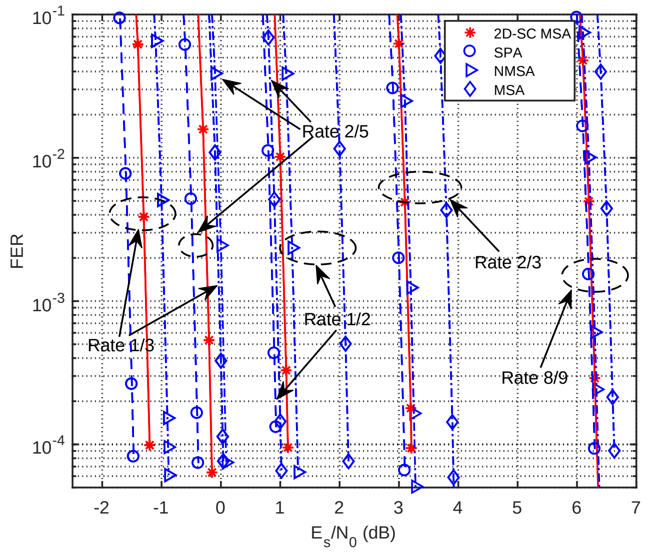
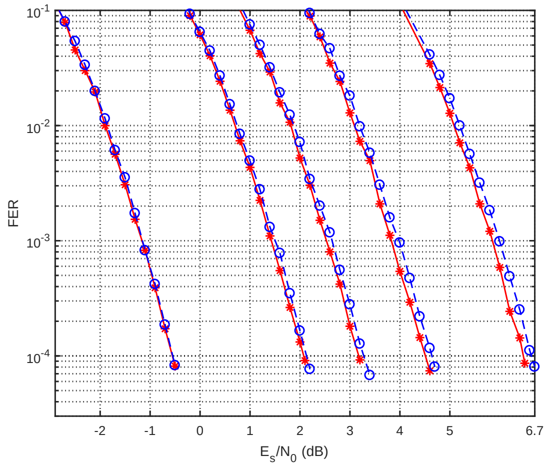
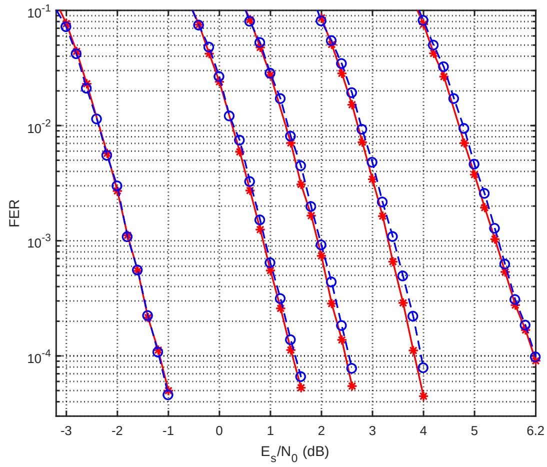


Figure 5.4: FER performance of the 5G LDPC codes decoded by different decoding algorithms with information bit lengths $K = 8448$ and code rate $R = 1/3, 2/5, 1/2, 2/3, 8/9$.

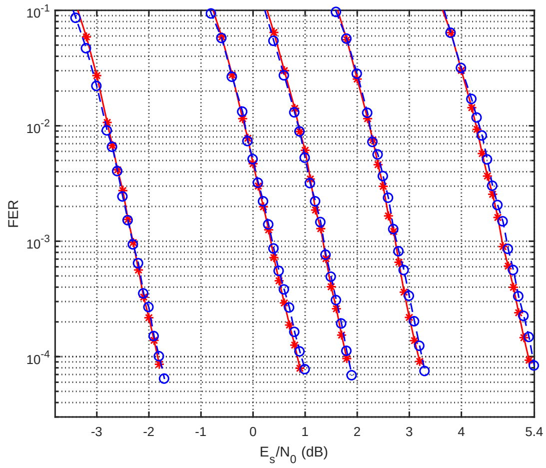
In addition, we also present the performance of the 5G LDPC codes with various information bit lengths K and different code rates R decoded by the proposed 2D-SC MSA and the SPA as follows. The red curves (solid lines) refer to the FER performance by using 2D-SC MSA and the blue curves (dash lines) represent the FER performance by using the SPA. When $K \leq 3840$, the markers and the corresponding code rates R (from left to right) is '$*$': $r = 1/5$, '$\circ$': $r = 1/3$, '$\triangleright$': $r = 2/5$, '$\diamond$': $r = 1/2$, '$\Box$': $r = 2/3$. When $K > 3840$, the markers and the corresponding code rates R (from left to right) is '$*$': $r = 1/3$, '$\circ$': $r = 2/5$, '$\triangleright$': $r = 1/2$, '$\diamond$': $r = 2/3$, '$\Box$': $r = 8/9$.



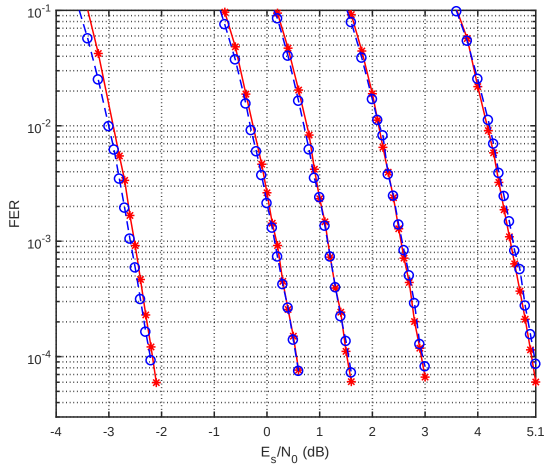
K=56



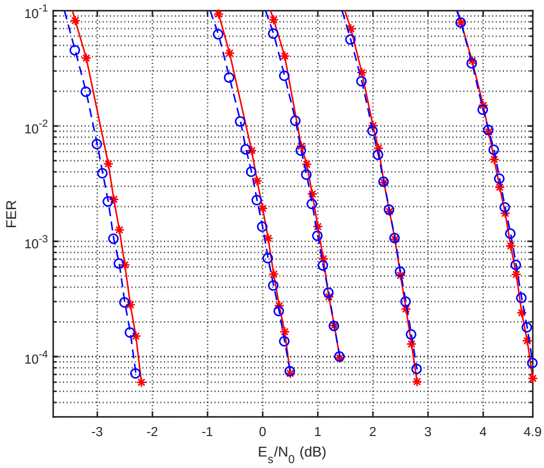
K=96



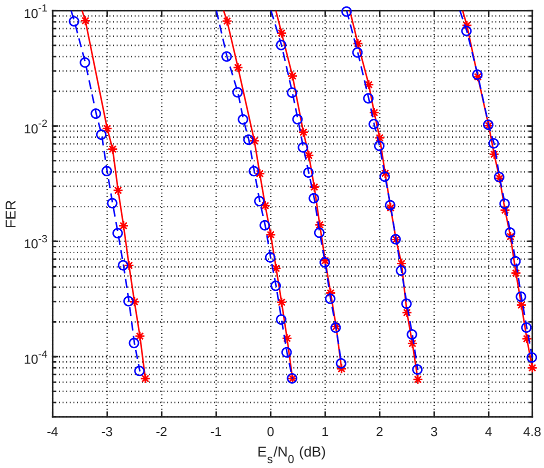
K=176



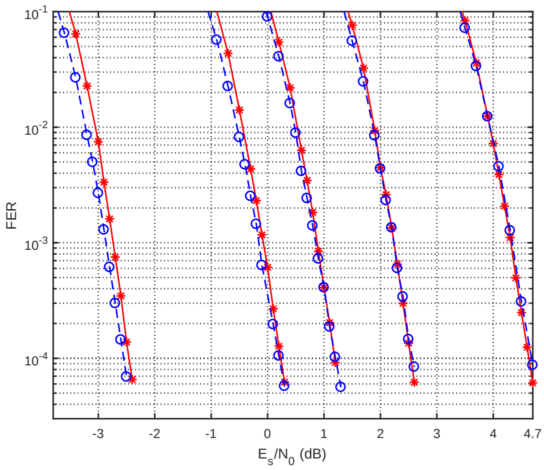
K=216



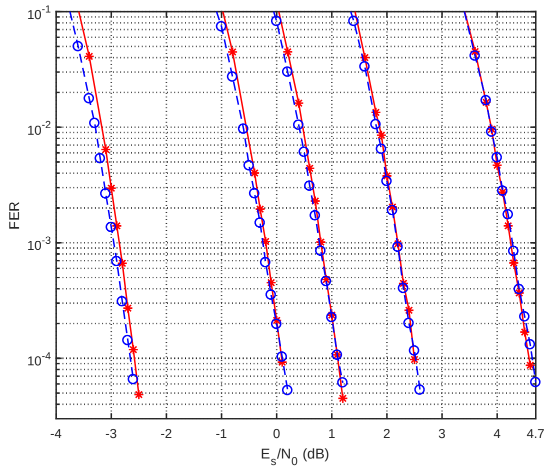
$K\!=\!256$



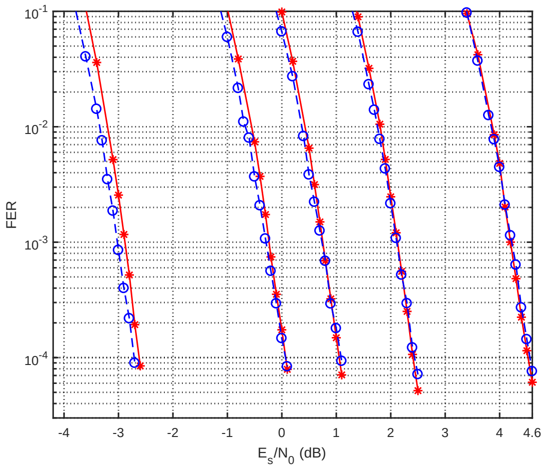
$K\!=\!288$



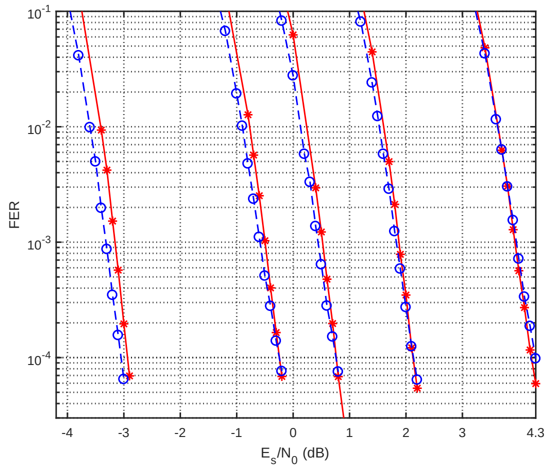
$K\!=\!320$



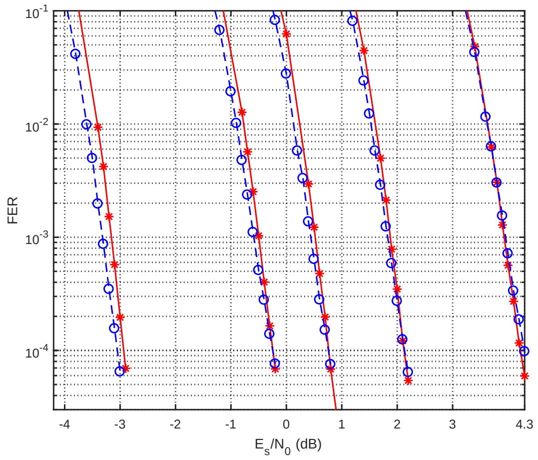
$K\!=\!336$



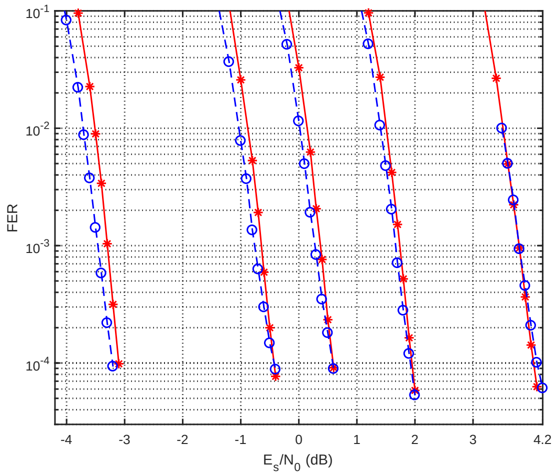
$K\!=\!376$



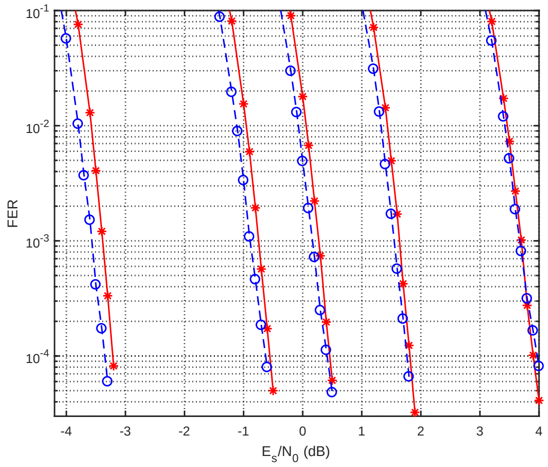
$K\!=\!416$



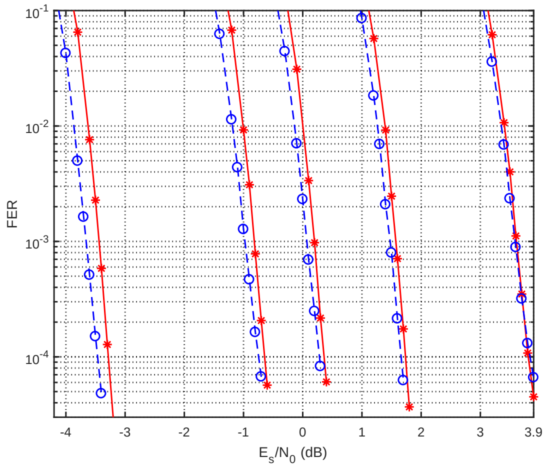
K=576



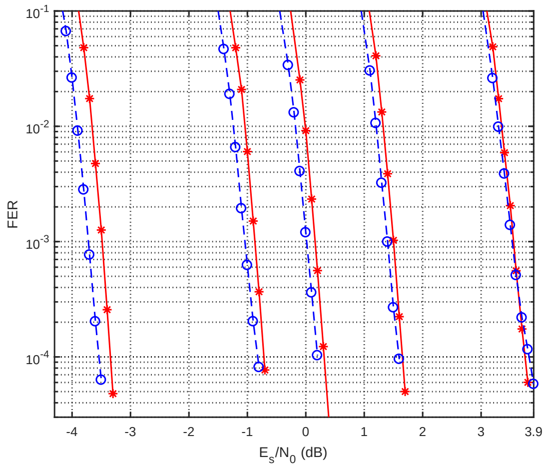
K=752



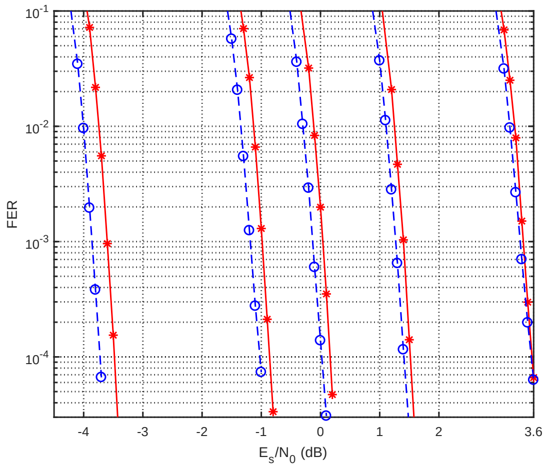
K=912



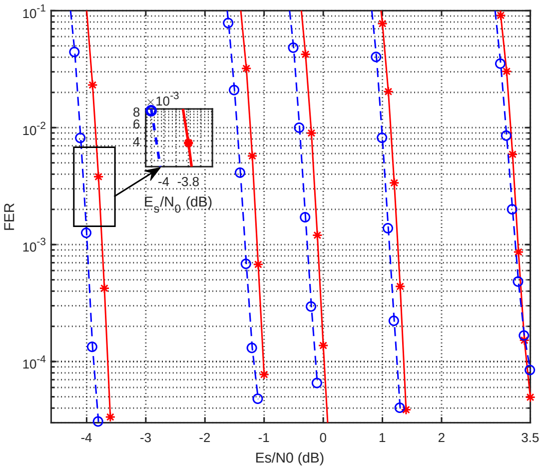
K=1088



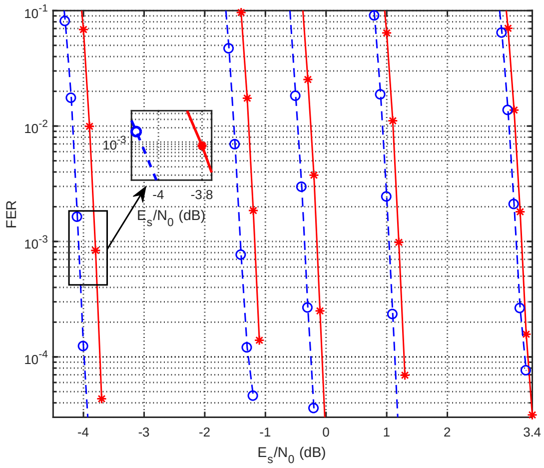
K=1248



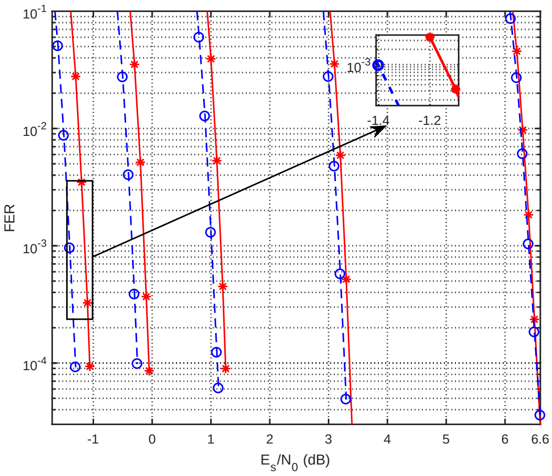
K=1728



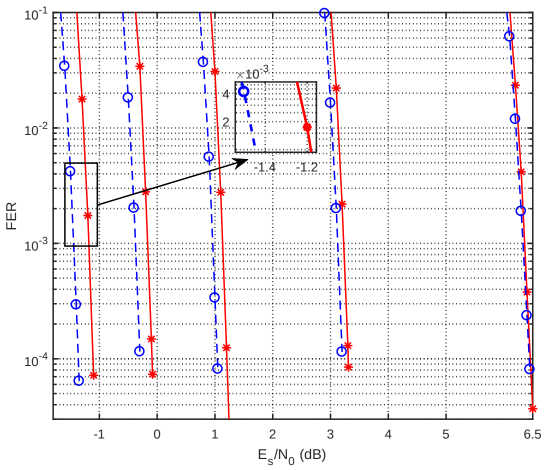
K=2688



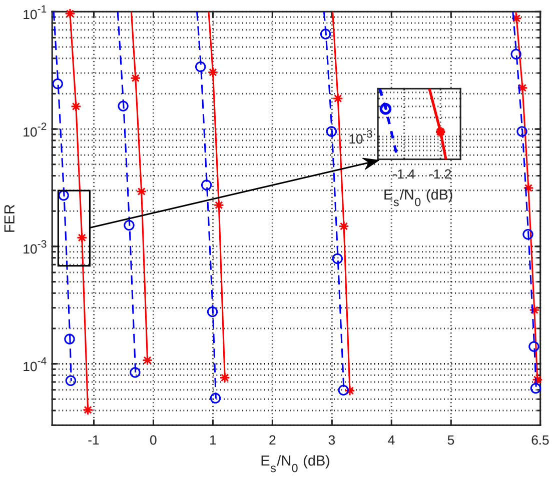
K=3840



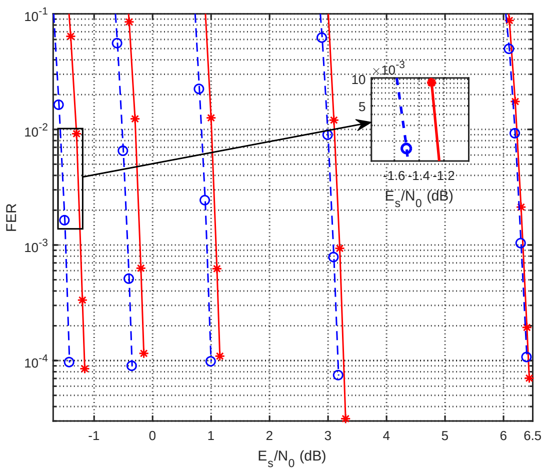
K=4096



K=5184



K=5568



K=6080

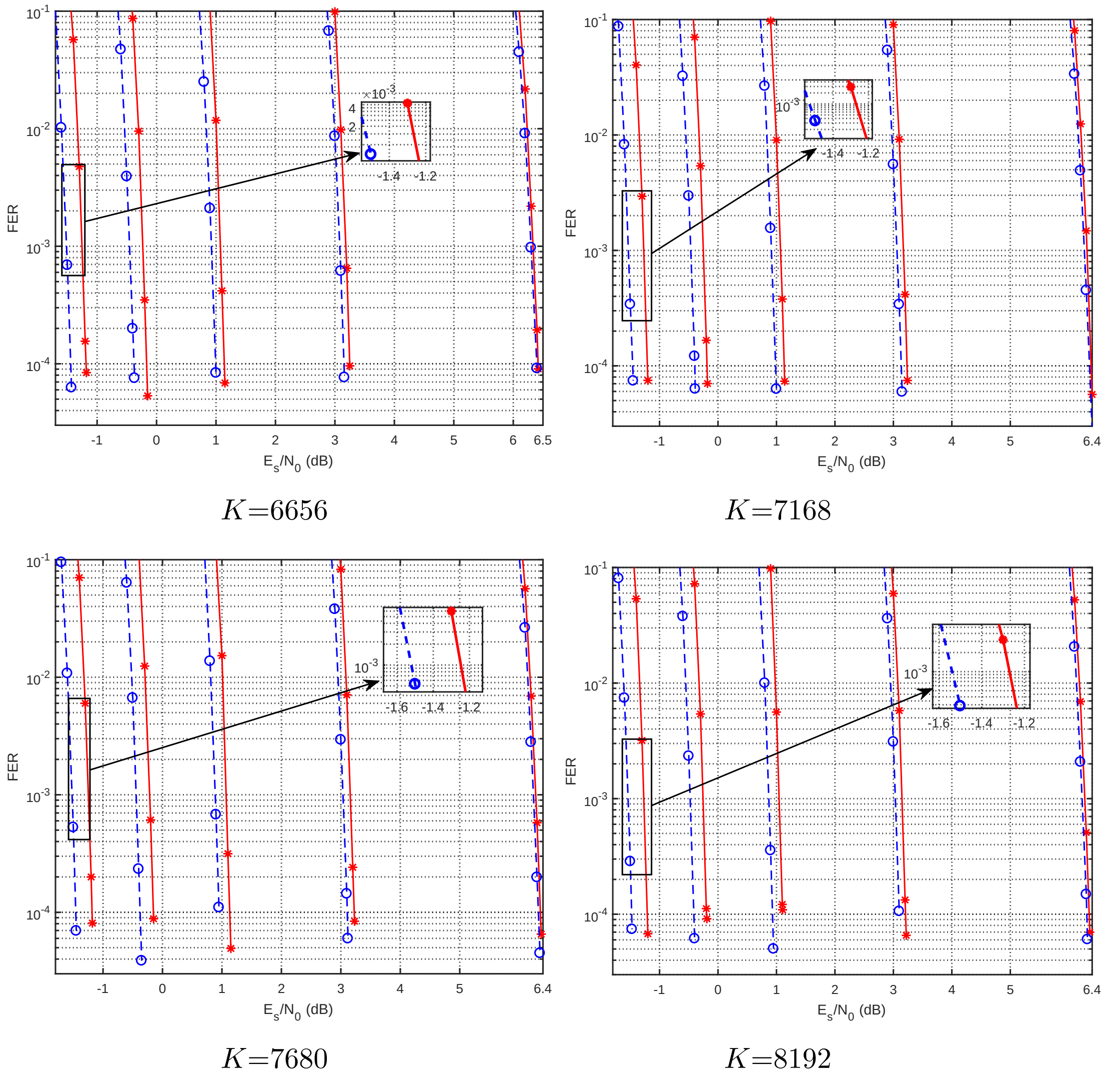


Figure 5.9: The FER performance of 5G LDPC codes with different information length K by using 2D-SC MSA

From the simulation results, we can see that the performance gap between the proposed 2D-SC MSA and the SPA, for a fixed code rate R, becomes larger when information bit lengths K increases. For the 5G LDPC codes with the same information bit lengths K, the performance gap enlarges as the code rate R decreases. The universality of the selected decoding parameters can be seen from the simulation results that the maximum performance gap of the proposed 2D-SC MSA from that of the SPA is less than 0.3 dB for all information bit lengths $56 \leq K \leq 8448$ and all code rates $1/5 \leq R \leq 8/9$.

5.5 The Enhanced QML Decoding Method

Although the proposed 2D-SC MSA improves the error performance of the 5G LDPC codes and it approaches the performance of the SPA, there is still a notable performance gap from the ML decoding for the 5G LDPC codes. This performance gap is due to the presence of the cycles with small girth in the Tanner graphs, which introduces the correlation in the messages and makes the SPA sub-optimal in terms of the error performance. To reduce the performance gap between the conventional BP decoding and the ML decoding for short LDPC codes, QML decoding methods were proposed in [108] and [109]. However, there is a high decoding complexity if we expect to obtain near-ML error performance. Therefore, in this section, we further improve the decoding method for the 5G short LDPC codes, aiming at approaching the error performance of the ML decoding with a desirable decoding complexity. In the following, we present our proposed enhanced QML (EQML) decoding method.

5.5.1 The Edge-wise Selection (EWS) Method

In Chapter 3, we see that the NWS method [108–110] selects the unreliable VNs based on the syndrome, which is a kind of hard information. To measure the reliability of each VN in a more precise way, here we would like to select the unreliable VNs based on the soft information exchanged during the conventional BP decoding and expect an improved error performance. Therefore, instead of focusing on the hard information given by syndrome, the sign evolution of the V2C messages is investigated for the node selection.

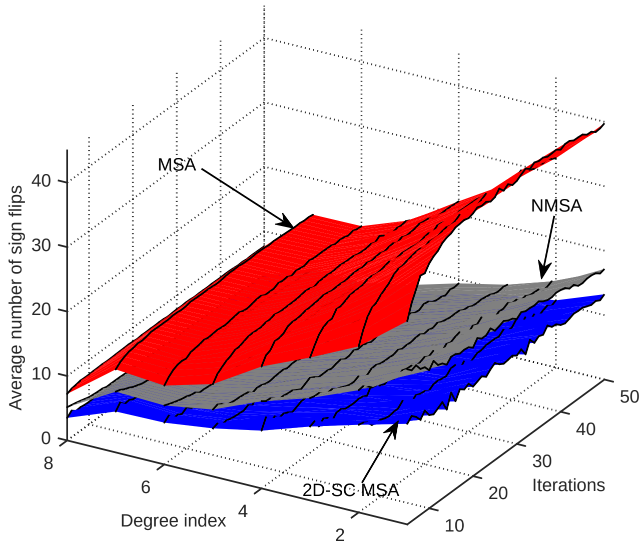


Figure 5.10: The average number of sign flips for the V2C messages per VN degree under decoding failure. An AWGN channel is considered. $R = 1/5$, $K = 120$. Note that $\alpha = 0.7125$ for the NMSA and $(\alpha, \beta) = (0.75, 1.25)$ for the 2D-SC MSA. $E_s/N_0 = -2.6$ dB. The VN degrees are shown in the order of $(5, 9, 10, 12, 14, 16, 22, 23)$ with the descending of the degree index.

Fig. 5.10 shows the average number of sign flips for the V2C messages per VN degree with the increase of iterations under the unsuccessful decoding case. Compared to the percentage of total sign flips per iteration in Fig. 5.1, we can see that the average number of sign flips per VN degree has a similar trend and the average number of sign flips of different decoding algorithms eventually saturates to a constant value. Apart from that, it also has a similar behavior as observed in [108] that the VNs with higher degrees are more prone to be in errors compared to other VNs. Although the 2D-SC MSA demonstrates its capabilities of reducing the number of sign flips compared to other decoding algorithms, there are still V2C messages having their sign fluctuated with the increase of iterations. Note that we observed similar behaviors for other 5G short LDPC codes with different block lengths.

Motivated by this observation, we propose a novel node selection method based on this sign flips phenomenon. Let $w_{k,n}$ be the number of sign flips for the

V2C message passed through the k-th edge of the n-th VN. Denoted by

$$w(v_n) = \sum_{k=1}^{d(v_n)} w_{k,n} \tag{5.5}$$

the total number of sign flips for the V2C messages of node v_n, where $d(v_n)$ refers to the degree of v_n.

At stage j reprocessing, the proposed node selection method chooses a new unreliable VN according to $w(v_n)$, which is computed from the decoding tests at the $(j-1)$-th stage. Define $w^{(T')}(v_n)$ as the number of sign flips for the V2C messages of node v_n obtained from the T'-th decoding test. For the T-th decoding test, only the node v_n with the maximum number of sign flips $w^{(T')}(v_n)$ is selected as the candidate node $v_s^{(T)}$. Then we have $\mathcal{V}_S^{(T)} = \left\{\mathcal{V}_S^{(T')} \cup v_s^{(T)}\right\}$. When $T' = 0$, $w^{(0)}(v_n)$ is obtained from the initial BP decoding. To avoid selecting the same VN in different reprocessing stages, we set $w^{(T)}(\mathcal{V_S}^{(T)}) = 0$ after the T-th decoding test.

Compared to the NWS method, the proposed node selection method explores more diversities on measuring the reliability of each VN since the selection criterion relies on the edge level rather than the node level messages. Thus, we would like to call it an EWS method. We would point out that the NWS method in [108] cannot be directly applied to the 5G LDPC codes since a few punctured VNs in the codeword do not have associated channel outputs. However, the EWS method estimates the reliability of each VN by the VN's extrinsic messages, which does not rely on the existence of the channel output of that VN.

5.5.2 The Partial Pruning Stopping (PPS) Rule

Define $\hat{\mathbf{r}}^{(T)}$ as the decoder input sequence for the T-th decoding test, which is obtained by replacing $\mathbf{r}(V_S^{(T)})$ by $\mathbf{m}_t$ on the channel output sequence. To collect

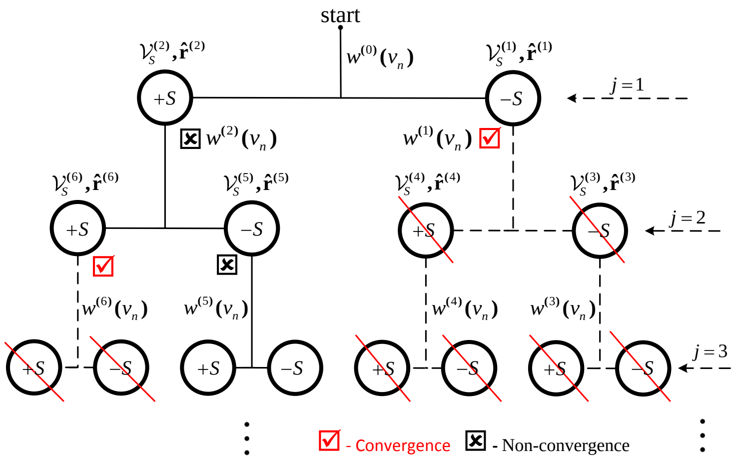


Figure 5.11: The general tree of the EQML decoding method with the PPS rule.

all possible output codewords, $2^{j_{\max}+1} - 2$ possible input sequences need to be tested before the decoder stops. This stopping rule is called list stopping rule (LDS) in [108]. Although the LDS rule guarantees the completion of the output codewords, which is beneficial to the error performance, it has a high complexity. To reduce complexity, we propose a PPS rule to avoid some decoding tests in the reprocessing. Define $\mathbf{x}^{(T)}$ and $\mathcal{X}$ as the output codeword after T decoding tests and the set of all valid codewords collected in the reprocessing, respectively. After the T-th decoding test, we save $\mathbf{x}^{(T)}$ in $\mathcal{X}$ if it is a valid codeword and perform the PPS rule, which stops the decoding tests on its associated sub-branches thereafter. The decoding tests on other sub-branches continue until $j_{\max}$ is reached, or there is no more active decoding test.

Fig. 5.11 depicts the general tree of the proposed EQML decoding method with the PPS rule. A branch is considered to be converged if there is a valid codeword found in that branch. Compared to the LDS rule, the proposed PPS rule reduces the decoding complexity by pruning the reprocessing on the sub-branches followed by a converged branch, which is shown as dash lines in Fig. 5.11.

Note that an alternative stopping rule was proposed in [109], where the reprocessing stops once the number of output codewords exceeds a preset threshold. However, the threshold has to be optimized in advance by computer simulations in order to obtain good performance. Compared to that, our proposed PPS rule can make a good balance between the decoding complexity and the error performance, which is desirable for practical applications.

Algorithm 5.2 The EQML Decoding Method

Perform BP decoding with $I_{\max}$
if a valid codeword is found **then**
  Output the codeword
else
  Initialize: $T = 1, j = 1, \mathcal{V}_S^{(0)} = \emptyset, T_F = 2^{j_{\max}+1} - 2$
  while $j \leq j_{\max}$ **do**
    for $t = 1 : 2^j$ **do**
      $T' = \left\lfloor \frac{T-1}{2} \right\rfloor$
      for $n = 1 : N$ **do**
        Compute $w^{(T')}(v_n)$ according to Eq. (5.5)
      end for
      Select $v_s^{(T)}$ as $v_s^{(T)} = \arg\max_{1 \leq n \leq N} w^{(T')}(v_n)$
      Determine $\mathcal{V_S}^{(T)} = \{\mathcal{V}_S^{(T')} \cup v_s^{(T)}\}$
      Generate $\mathbb{M}$ by enumerating $\mathbf{r}(\mathcal{V}_S^{(T)})$ to $\pm S$
      Determine $\hat{\mathbf{r}}^{(T)}$ by setting $\mathbf{r}(\mathcal{V}_S^{(T)}) = \mathbf{m}_t$
      Perform BP decoding with $\hat{\mathbf{r}}^{(T)}$ and $I_{\max}$
      Perform PPS rule
      $T = T + 1$
    end for
    $j = j + 1$
  end while
  if $\mathcal{X} \neq \emptyset$ **then**
    Output $\mathbf{x}_{best} = \arg\min_{\mathbf{x}^{(T)} \in \mathcal{X}} \sum_{n=1}^{N} \left| r(v_n) - {x_n}^{(T)} \right|^2$
  else
    Declare decoding failure.
  end if
end if

5.5.3 The EQML Decoding Method

Set the maximum number of decoding iterations as $I_{\max}$ for all BP decoding process. Define the output codeword of the EQML decoding method as $\mathbf{x}_{best}$. By utilizing the EWS method and the PPS rule in the reprocessing, the proposed EQML decoding method is summarized in **Algorithm 5.2**.

Note that the proposed node selection method can efficiently work together with the PSC method in the 2D-SC MSA by adding the associated counters at each VNs. Whenever a V2C message is set to zero by the self-correction method, the counter is incremented by one. At the end of each decoding test, the number of self-correction operations computed by the counter can be used for the EWS method in the reprocessing if there is a decoding failure. Moreover, the complexity of the self-correction method is associated with the size of H_{core} instead of the size of H since the PSC method only focuses on the V2C messages sent from the VNs in H_{core}. Thus, it results in only slightly increased memory demands according to the size of H_{core}, i.e., $\mathcal{O}(M \cdot N_{\text{core}})$.

5.6 Error Performance and Complexity Analysis

In this section, we first derive the lower bound on FER for the proposed EQML decoding method with a given $j_{\max}$ by considering different error cases. Then, the decoding complexity of the proposed EQML decoding method with a given $j_{\max}$ is evaluated through the average number of iterations performed for one decoded codeword.

5.6.1 Error Performance Analysis

The lower bounds on FER under the ML decoding for linear block codes have been investigated by many researchers, e.g., [170–173]. We first present the expression of FER for the proposed EQML decoding method and then derive its asymptotic bound. Note that there are two cases that the proposed EQML decoding method fails to obtain the correct codeword. That is either no valid codeword is obtained or the output codeword is not the correct one. Define $P_{EL}^{j_{\max}}$ as the probability that the EQML decoding method does not obtain a valid codeword and outputs an empty list after $j_{\max}$ stages of the reprocessing. We also define $P_{UE}^{j_{\max}}$ as the probability that the EQML decoding method obtains a wrong codeword at the end of $j_{\max}$ reprocessing stages, where an undetected error happens in this case. The FER of the EQML decoding method with $j_{\max}$ reprocessing stages, denoted by $P_e^{j_{\max}}$, is given by

$$P_e^{j_{\max}} = P_{EL}^{j_{\max}} + P_{UE}^{j_{\max}}. \tag{5.6}$$

In the following, we present the details of the derivation. For a given maximum $j_{\max}$ reprocessing stage, the proposed EQML decoding method operates consecutively from one stage to the next. To derive $P_{EL}^{j_{\max}}$, we need to find the probability that the EQML decoding method misses valid codewords at each stage. Without loss of generality, let E_{EL}^{t} be the event that the t-th decoding test at the j-th reprocessing stage fails to obtain a valid codeword. Define P_{EL}^{j} as the probability that the reprocessing at the j-th stage outputs an empty list of codewords. That is

$$P_{EL}^{j} = P(E_{EL}^{1}, E_{EL}^{2}, \cdots, E_{EL}^{2^j}) = \prod\nolimits_{t=1}^{2^j} P(E_{EL}^{t}). \tag{5.7}$$

Note that the EQML decoding method outputs an empty list of codewords if all $2^{j_{\max}+1} - 1$ decoding tests are unable to obtain a valid codeword. Referring to the initial BP decoding test as the stage-0 decoding, we have

$$P_{EL}^{j_{\max}} = \prod_{j=0}^{j_{\max}} P_{EL}^{j} = \prod_{j=0}^{j_{\max}} \prod_{t=1}^{2^j} P(E_{EL}^{t}), \tag{5.8}$$

It is known that the EQML decoding method chooses the best codeword $\mathbf{x}_{best}$, which is the closest to the received sequence, among all valid codewords obtained from the decoding tests. To derive $P_{UE}^{j_{\max}}$, we first discuss the probability that a wrong codeword is obtained at each reprocessing stage. Let E_{UE}^{t} be the event that the t-th decoding test at the j-th stage reprocessing obtains a wrong codeword. Denoted by P_{UE}^{j} the probability that a wrong codeword is obtained at stage-j reprocessing, and it is given by

$$P_{UE}^{j} = P(E_{UE}^{t} | \mathbf{x}_{\text{best}} = \mathbf{x}^{t}) = (1 - P_{EL}^{j}) \cdot P(\mathbf{x}_{\text{best}} \neq \mathbf{z}), \tag{5.9}$$

where $\mathbf{z}$ is the transmitted codeword.

Note that the PPS rule applied to the EQML decoding method terminates the subsequent decoding tests when a valid codeword is obtained at the j-th reprocessing stage. As a result, the probability that the EQML decoding method with $j_{\max}$ reprocessing stages obtains a wrong codeword can be given by

$$P_{UE}^{j_{\max}} = \sum_{j=0}^{j_{\max}} P_{UE}^{j} \cdot \prod_{j'=0}^{j-1} P_{EL}^{j'}, \tag{5.10}$$

In Fig. 5.12, a binary tree illustrates the FER performance of the proposed EQML decoding method with the first two reprocessing stages. The probabilities of every event associated with the decoding tests are shown in the figure. According to Fig. 5.12, we conclude the FER of the EQML decoding method from (5.8) and

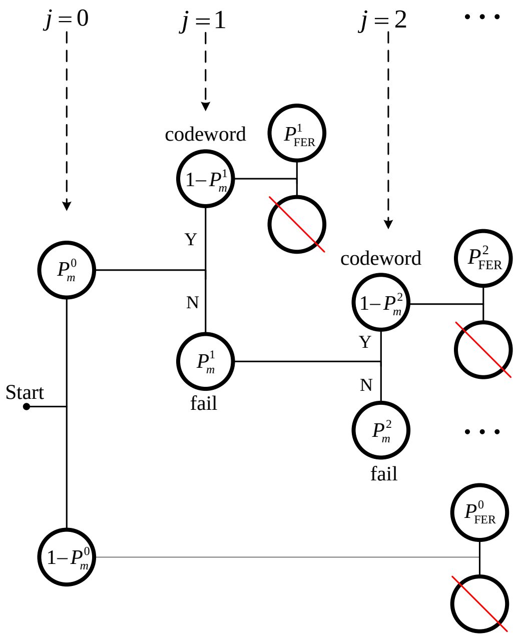


Figure 5.12: The binary tree of the error probability for the proposed EQML decoding method operations.

(5.10) as

$$P_e^{j_{\max}} = \prod_{j=0}^{j_{\max}} P_{EL}^j + \sum_{j=0}^{j_{\max}} (1 - P_{EL}^j) \cdot P(\mathbf{x}_{\text{best}} \neq \mathbf{z}) \cdot \prod_{j'=0}^{j-1} P_{EL}^{j'}. \tag{5.11}$$

According to (5.11), we derive the lower bound on FER performance for the proposed EQML decoding method. Suppose that the all-zero codeword is transmitted over the AWGN channel by using binary phase-shift keying (BPSK) modulation. We define the normalized SNR as $1/\sigma^2$ [74], where σ^2 is the noise variance. Motivated by [173], the lower bound on FER for an LDPC code under ML decoding over AWGN channels can be given by

$$P_{\mathrm{ML}} \geq n_d \cdot Q\left(\sqrt{\frac{d_{\min}}{\sigma^2}}\right), \tag{5.12}$$

where $d_{\min}$ represents the minimum Hamming weight of the LDPC code and n_d is the number of codewords with $d_{\min}$. Therefore, the lower bound on FER of the

EQML decoding method is

$$P_e^{j_{\max}} \geq \prod_{j=0}^{j_{\max}} P_{EL}^{j} + n_d \cdot Q\left(\sqrt{\frac{d_{\min}}{\sigma^2}}\right) \cdot \sum_{j=0}^{j_{\max}} (1 - P_{EL}^{j}) \cdot \prod_{j'=0}^{j-1} P_{EL}^{j'}. \tag{5.13}$$

5.6.2 Decoding Complexity Analysis

Apart from the error performance, the decoding complexity is also a critical issue for practical implementation. In fact, reducing decoding complexity is significantly important for various applications, such as remote control, factory automation, and smart grid. Here we evaluate the decoding complexity of the proposed EQML decoding method for a given $j_{\max}$.

Since the conventional BP decoding is executed in multiple times in the reprocessing, we use the average number of iterations for decoding one codeword to evaluate the decoding complexity. Define $I_{l,f}$ as the number of iterations used during the l-th reprocessing test of the f-th received codeword. Let I_{avg} be the average number of iterations used for decoding one codeword. Then we have

$$I_{\text{avg}} = \frac{1}{F} \sum\nolimits_{f=1}^{F} (I_{0,f} + \sum\nolimits_{l=1}^{(2^{j_{\max}+1}-2)} I_{l,f}), \tag{5.14}$$

where F represents the total number of codewords transmitted.

Note that the decoding complexity of a transmitted codeword consists of two parts, i.e., the iterations performed in the conventional BP decoding and the reprocessing. If the conventional BP decoding does not converge to a valid codeword after a preset $I_{\max}$, the reprocessing is performed which largely increases the decoding complexity. More specifically, for a preset $I_{\max}$, the decoding complexity of the proposed EQML decoding method increases exponentially with $j_{\max}$ and is equal to $(2^{j_{\max}+1} - 2) \cdot I_{\max}$ in the worst case.

5.7 Numerical Results

In this section, we demonstrate the FER performance of the proposed EQML decoding method and compare it with that of the pure SPA decoding. We also compare the decoding complexity of the EQML decoding method with different stopping criteria and $j_{\max}$. We consider the 5G LDPC codes with information bit lengths $K = 120$, code rates $R = 1/5$ and $1/3$, which have the associated code block lengths $N = 600$ and 360, respectively. All the simulation settings used are the same as shown in Section 5.4.4. Note that the 2D-SC MSA is used for each BP decoding test in the EQML decoding method with $I_{\max} = 50$.

5.7.1 The FER Performance

In Fig. 5.13 and Fig. 5.14, we investigate the FER performance of the proposed EQML decoding method with different stopping criteria and $j_{\max}$. The FER performance of the SPA decoding is also shown in Fig. 5.13 and Fig. 5.14. For a fair comparison, we set the maximum number of iterations performed by the SPA decoding the same as it is used by the EQML decoding method with the LDS rule, i.e., $I_{\text{SPA}} = (2^{j_{\max}+1} - 1) \cdot I_{\max}$. For convenience, we represent I_{SPA} by T_F, where $T_F = 2^{j_{\max}+1} - 1$, refers to the number of the decoding tests to be conducted.

We can see from Fig. 5.13 and Fig. 5.14 that the proposed EQML decoding method with $j_{\max} = 4$ outperforms the SPA decoding with $T_F = 31$ for $N = 600$ and 360 by about 0.2 dB. The performance gain increases to 0.3 dB when $j_{\max} = 6$ compared to the SPA decoder with $T_F = 127$. Note that the Polyanskiy-Poor-Verdú (PPV) bounds [129] for the block lengths $N = 600$ and 360 are also shown in Fig. 5.13 and Fig. 5.14, respectively, where the performance gaps between the EQML decoding method with $j_{\max} = 6$ and the PPV bound for

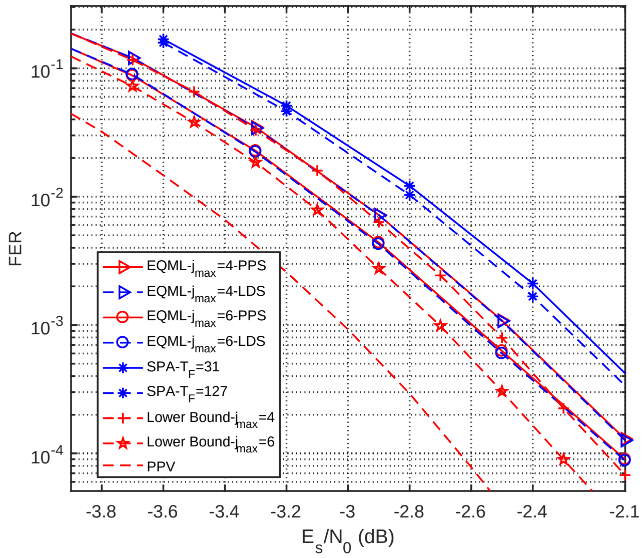


Figure 5.13: FER performance of the EQML decoding method for the 5G LDPC code with $N = 600, R = 1/5$.

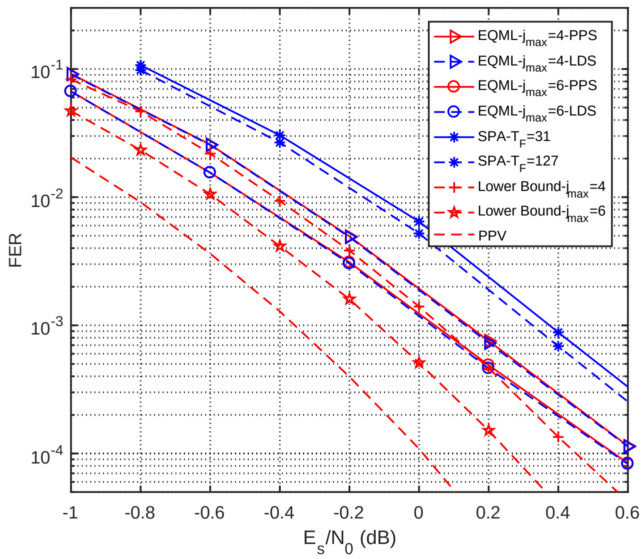


Figure 5.14: FER performance of the EQML decoding method for the 5G LDPC code with $N = 360, R = 1/3$.

different block lengths are within 0.5 dB. More importantly, the EQML decoding method with the PPS rule achieves almost the same FER performance compared to that of the LDS rule for both $j_{\max} = 4$ and 6.

In Fig. 5.13 and Fig. 5.14, we also show the proposed lower bound on the FER of the EQML decoding method with $j_{\max} = 4$ and 6 for the two LDPC short codes. More specifically, the number of codewords with the minimum Hamming distance is computed by the method introduced in [174]. The average probability of an empty list at each reprocessing stage is obtained from the simulations. In other words, the plotted lower bound is based on a semi-analytical approach. We can see from the figures that the gap between the lower bound and the simulation result is close for both LDPC codes particularly in the low SNR region.

5.7.2 Decoding Complexity Comparison

We further compare the decoding complexity of the proposed EQML decoding method with different stopping criteria and $j_{\max}$ in terms of the average number of iterations I_{avg}.

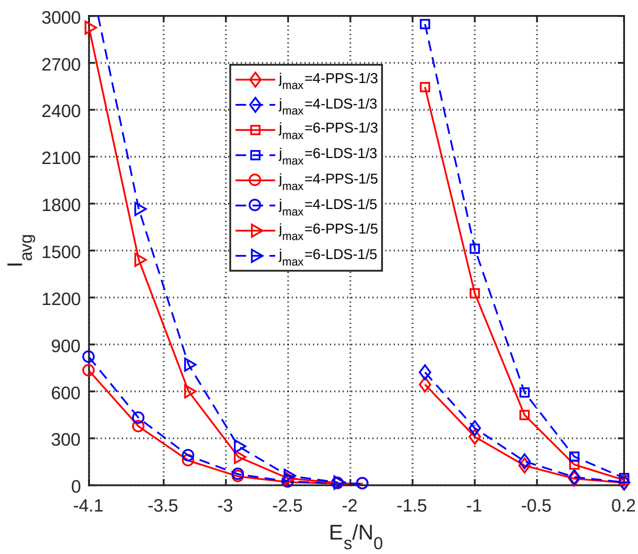


Figure 5.15: The comparison of I_{avg} for the 5G LDPC code with the LDS and the PPS rules, $N = 360$ and 600.

The comparison of the average number of iterations I_{avg} for the block lengths

$N = 360$ and 600 with the EQML decoding method for different stopping rules is shown in Fig. 5.15. We can see that our proposed EQML decoding method with the PPS rule requires a less number of I_{avg} for both $j_{\max} = 4$ and 6 compared to that with the LDS rule. In particular, there is about 15 % less I_{avg} required for the E_s/N_0 from -3.7 dB to -3 dB with $N = 600$, and for the E_s/N_0 -1.2 dB to -0.6 dB with $N = 360$. Moreover, the reduction of the decoding complexity can be about 25 % in the SNR region from -3.7 dB to -2.8 dB and from -1 dB to -0.2 dB for $N = 600$ and $N = 360$, respectively, when $j_{\max}$ is equal to 6.

5.8 Summary

In this chapter, we proposed an EQML decoding method for the 5G LDPC codes. Firstly, we proposed an improved decoding algorithm, called the 2D-SC MSA to approach the FER performance of the SPA for the 5G LDPC codes. The proposed 2D-SC MSA adopts self-correction method and message amplification, where the self-correction method reduces the frequency of sign flips for the V2C messages, and the message amplification reinforces the reliability of the V2C messages, respectively. Then we employed the EQML decoding method to further improve the FER performance of the 5G short LDPC codes. The EQML decoding method adopts the EWS method to improve the accuracy of the node selection for unreliable VNs. In addition, the PPS rule was proposed for the EQML decoding method to reduce the decoding complexity. We also evaluated the FER performance of the proposed EMQL decoding method by using the semi-analytical method and examined the decoding complexity for different stopping criteria and $j_{\max}$. Simulation results show that the proposed 2D-SC MSA can approach the FER performance of the SPA within 0.3 dB. The EQML decoding method outperforms the SPA for the 5G short LDPC codes and approaches the PPV bound within 0.4 dB in the low SNR region when $j_{\max} = 6$. In addition, the proposed PPS rule has a lower decoding complexity compared to the LDS rule without performance loss.

Chapter 6

The AMP-aided Decoding Scheme of 5G LDPC Codes

6.1 Introduction

In the previous chapter, we present the design of the EQML decoding method for the 5G LDPC codes with FER performance close to the ML decoding and a relatively low decoding complexity. In this chapter, we consider improving the performance of the 5G short LDPC codes with code rates $R \leq 0.5$ by using one-step reprocessing based on the approximate message passing (AMP) detector.

6.1.1 Overview of the general AMP algorithm

The AMP algorithm was first proposed in [121] and further studied in [175, 176]. The algorithm exploits an iterative refining process to recover the sparse unknown signal $\mathbf{x}$ from a noisy measurement $\mathbf{y}$ via using the Gaussian approximation during message passing. Compared to the popular class of reconstruction schemes based on linear programming methods, e.g., the least absolute shrinkage and selection operator (LASSO) algorithm [177], the AMP algorithm is an elegant approach that achieves identical performance as linear programming with a lower implementation complexity. Consider the scenario of noisy measurement, which is modeled as

$$\mathbf{y} = \mathbf{M}\mathbf{x} + \mathbf{w}, \tag{6.1}$$

where $\mathbf{M}$ is the $n \times M$ measurement matrix and $\mathbf{w} \sim \mathcal{N}(0, \sigma^2 \mathbf{I})$.

The AMP algorithm starts from an initial guess $\hat{\mathbf{x}}^0 = \mathbf{0}$ of the target signal $\mathbf{x}$ and the initial residue $\mathbf{z}^0 = \mathbf{y}$. The algorithm proceeds iteratively according to

$$\begin{aligned} \hat{\mathbf{x}}^{t+1} &= \eta_t \left(\mathbf{M}^* \mathbf{z}^t + \hat{\mathbf{x}}^t \right) \\ \mathbf{z}^t &= \mathbf{y} - \mathbf{M}\hat{\mathbf{x}}^t + \frac{1}{\delta} \mathbf{z}^{t-1} \left(\sum_{i=1}^{N} \frac{\eta'_{t-1} \left(\mathbf{M}_i^* \mathbf{z}_i^{t-1} + \hat{x}_i^{t-1} \right)}{N} \right). \end{aligned} \tag{6.2}$$

Here $\eta_t(*)$ is a *threshold* function applied componentwise to the input vector, $\hat{\mathbf{x}}^t \in \mathbb{R}^N$ is the current estimate of $\mathbf{x}$ at t-th iteration, $\hat{x}_i^{t-1}$ is the i-th element of $\hat{\mathbf{x}}^{t-1}$, and $\mathbf{z}^t \in \mathbb{R}^n$ is the current residue. Further, $\eta'_t(*)$ is the first derivative of $\eta_t(*)$ with respect to the input. The term $\delta = n/N$ is known as the under-sampling fraction, which is an important parameter for the study of sparsity-undersampling trade-off [121]. Finally, $\mathbf{M}^*$ represents the transpose of $\mathbf{M}$, and $\mathbf{M}_i$ represents the i-th column of the $\mathbf{M}$. Note that it is commonly assume that each element of $\mathbf{M}$ follows identical and independent Gaussian distribution and each column is normalized to have unit norm, i.e., $m_{i,j} \in \mathcal{N}(0, 1/n)$. This assumption can ensure the measurement matrix satisfies the restricted isometry property (RIP) [178], and facilitate the algorithm analysis.

Recall from previous that during the iterative denoising process the variance of the operation $\mathbf{P}(\mathbf{x} - \hat{\mathbf{x}}^t)$ reduces with respect to the iteration t, assme that

$\mathbf{P}$ does not correlate with $\mathbf{x}$ and $\hat{\mathbf{x}}$. Note that in most of the time, this is not the case, instead $\hat{\mathbf{x}}^t$ after the first iteration becomes strongly dependent. The derivation of the correction term

$$\frac{1}{\delta}\mathbf{z}^{t-1}\left(\sum_{i=1}^{N}\frac{\eta'_{t-1}\left(\mathbf{M}_i^*\mathbf{z}_i^{t-1}+\hat{x}_i^{t-1}\right)}{N}\right)$$

in (6.10) ensure that the correlations between $\mathbf{x}$ and $\hat{\mathbf{x}}^t$ at minimal [121].

6.1.2 State Evolution

For the iterative process in the AMP algorithm, a *state variable* τ_t and its evolution with respect to iteration $t = 1, 2, \ldots$ are introduced to characterize the convergence performance during the iterative process [121, 176]. In particular, for a large-scale system, where $n, N, k \to \infty$ with fixed n/N and k/N, and the sparsity k denotes the number of non-zero elements in sequence $\mathbf{x}$, $k \ll N$, the state evolution of τ_t is given by [176]

$$\begin{aligned}
\tau_t^2 &= \sigma^2 + \frac{\mathbb{E}\left[\|\hat{\mathbf{x}}^t - \mathbf{x}\|_2^2\right]}{\delta} \\
&= \sigma^2 + \frac{\mathbb{E}\left[\|\eta_{t-1}(\mathbf{M}^*\mathbf{z}^{t-1} + \hat{\mathbf{x}}^{t-1}) - \mathbf{x}\|_2^2\right]}{\delta} \\
&\equiv \sigma^2 + \frac{\mathbb{E}\left[\|\eta_{t-1}(\mathbf{x} + \tau_{t-1}\mathbf{v}) - \mathbf{x}\|_2^2\right]}{\delta},
\end{aligned} \tag{6.3}$$

where $\delta = n/N$, σ^2 is the variance of noise vector $\mathbf{w}$ and $\mathbf{v} \sim \mathcal{N}(0, \mathbf{I})$. The state evolution begins with

$$\tau_0^2 = \sigma^2 + \frac{\mathbb{E}\left[\|\mathbf{x}\|_2^2\right]}{\delta}. \tag{6.4}$$

Note that the input of the function $\eta(*)$ in the third equality of (6.3) can be interpreted a noisy estimate of $\mathbf{x}$, i.e., $\mathbf{x} + \tau_t\mathbf{v}$. The expectation in (6.3) is taken over the random variables $\mathbf{x}$ and $\mathbf{v}$.

From (6.3), it can be observed that τ_t^2 characterizes the mean squared error (MSE) of each estimate $\hat{\mathbf{x}}^t$. Hence, it implies that the convergence of AMP algorithm in terms of MSE can be tracked by exploiting the evolution of state variable τ^2. Obtaining τ_t for each iteration via state evolution in (6.3) requires a high computational complexity. Thus, an empirical estimate of τ_t [179], i.e.,

$$\tau_t = \frac{1}{\sqrt{n}} \|\mathbf{z}^t\|_2, \tag{6.5}$$

is commonly used during the implementation of AMP algorithm.

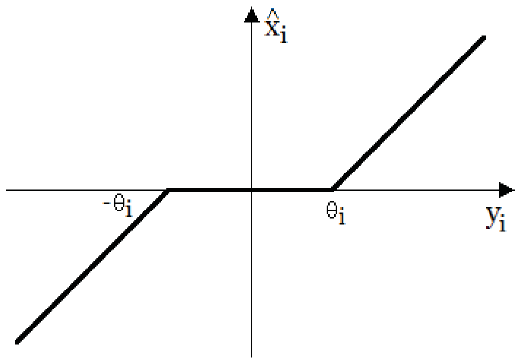


Figure 6.1: Soft-thresholding function with threshold θ.

6.1.3 Threshold function $\eta(*)$

The threshold function $\eta_t(*)$ described in the previous subsections depends on iteration. For commonly used *soft-threshoding* denoiser [121], we have $\eta_t(y) = \eta_t(y; \theta_t)$ that is given by the function

$$\hat{x} = \eta(y, \theta) = \begin{cases} y - \theta, & \text{if } y > \theta \\ 0, & \text{if } -\theta \le y \le \theta \cdot \\ y + \theta, & \text{if } y < -\theta \end{cases} \tag{6.6}$$

Here, $\eta_t(y;\theta_t)$ takes a different θ in each iteration, which is commonly set to the empirical τ_t given in (6.5). An example of soft-thresholding function is illustrated in Figure 6.1.

6.2 Problem Statement

As shown in [107–110] and the previous chapter, the QML decoding methods utilize the idea of list decoding in the reprocessing, where several rounds of decoding tests are conducted after the failure of the initial decoding test. Only the decoding output from the successful decoding tests are collected to form a list of the decoded codewords and the best codeword is chosen from the list as the decoder output in the end according to certain decision metric. Although these QML decoding methods can approach the FER performance of the ML decoding, they still adopt exhaustive decoding tests based on all possible combinations of the unreliable VNs. To achieve a desirable error performance, a large number of repeated decoding tests is required to increase the probability of correcting the unreliable VNs, especially for the LDPC codes with long block lengths. This results in a high decoding complexity. In addition, there is no guarantee that a valid codeword is always obtained after each decoding test. If all the decoding tests in the reprocessing fail to obtain a valid codeword, a decoding failure is declared and the error rate performance has no improvement in this case while it still causes a fairly high decoding complexity due to the conducted decoding tests.

6.2.1 Main Contributions

In this work, we propose an AMP-aided reprocessing scheme to combat the aforementioned problems with a low decoding complexity, which improves the performance of 5G short LDPC codes with code rates $R \leq 0.5$. The main contributions of this work are summarized below:

- We formulate the decoding of LDPC codes as a compressed sensing (CS) problem, where we use a sparse error vector to indicate the reliability of each VN in the codeword and reconstruct the error vector by the AMP algorithms, which are commonly used techniques to solve the CS problem.

- We proposed an improved decoding scheme based on the reprocessing for the 5G short LDPC codes with code rates $R \leq 0.5$. In the proposed decoding scheme, only one decoding test is conducted in the reprocessing, where the decoder input sequence is updated by the detector based on the AMP algorithm and the designed bit-flipping rule from the channel output sequence. More specifically, the detector estimates the reliability of the channel output for each node v_n from the residue signal by using the AMP algorithm [121], where the residue signal is constructed by removing the decoded signal from the associated channel output sequence. After that, the signs of the channel outputs on the unreliable VNs are flipped according to the preset threshold to generate the updated decoder input sequence. A new round of decoding test is conducted afterwards and the decoder outputs the valid codeword if the decoding test successes.

- We further propose an AMP-Enhanced Quasi-Maximum Likelihood (EQML) decoding scheme for the 5G LDPC codes, where the AMP detector is used as a post-process operation for the unsuccessful decoding tests in the reprocessing of the EQML decoding. In this way, we can increase the probability of obtaining a valid codeword, and the error performance of the EQML decoding can be further improved.

- We investigate the error performance for the AMP-aided decoding scheme with both one-time belief-propagation (BP) decoding and the EQML decoding. Some properties of the proposed AMP-aided decoding scheme such as false flip rate (FFR), the denoiser success rate over the total number of decoding failure (DSRF) and the denoiser success rate over total transmissions (DSRT) are also analyzed and discussed.

We would like to emphasis that the AMP algorithms used in the conventional CS methods aim to recover the sparse unknown signal from a noisy measurement via using the Gaussian approximation during message passing [175, 176]. For example, it solves the user identification and channel estimation problems in massive machine-type communications [180, 181]. However, the AMP algorithm here provides the additional information about the reliability of the received signal, where the VN positions with high probability in error are determined by the AMP algorithm for bit-flipping to obtain the updated decoder input sequence. In this way, the number of correct VNs in the decoder input sequence increases and the probability of the successful decoding in the next round of decoding test can be improved.

6.3 Decoding Model

Assume that $\mathbf{s} = [s_1, s_2, \ldots, s_N]$ is the modulated signal of the transmitted codeword $\mathbf{z}$ with $s_n = (-1)^{z_n}$. Let $\mathbf{w} = [w_1, w_2, \ldots, w_N]$ be the noise vector with each independent $w_j \sim \mathcal{N}(0, \sigma^2)$ being a i.i.d. Gaussian random variable with zero-mean and variance σ^2. The received signal $\mathbf{y} = [y_1, y_2, \ldots, y_N]$ can be represented as

$$\mathbf{y} = \mathbf{s} + \mathbf{w}. \tag{6.7}$$

Denote the error vector for a given temporary decision of the codeword $\hat{\mathbf{z}}$ by $\mathbf{e} = [e_1, e_2, \ldots, e_N]$, where $\mathbf{e} \in \mathbb{R}^N$. As we know that for a well-designed LDPC code, most of the VNs tend to approach the correct values under the conventional BP decoding within the first few iterations. Thus, the number of unreliable VNs after the BP decoding is small compared to the block length of the codeword. Since $e_n \neq 0$ if $\hat{z}_n \neq z_n$, motivated by the above observations, $\mathbf{e}$ tends to be a sparse vector with a small number of nonzero elements compared to the zeros after a few iterations. Define $\hat{\mathbf{s}}$ as the modulated signal of $\hat{\mathbf{x}}$, where $\hat{s}_n = (-1)^{\hat{z}_n}$. For the given residue signal $\hat{\mathbf{y}}$, the error vector $\mathbf{e}$ is obtained by

$$\hat{\mathbf{y}} = \mathbf{y} - \hat{\mathbf{s}} = \mathbf{s} - \hat{\mathbf{s}} + \mathbf{w} = \mathbf{e} + \mathbf{w}. \tag{6.8}$$

Hence, the error vector $\mathbf{e}$ is an integer vector of length N with $e_n \in \{0, \pm 2\}$, where $e_n = +2$ indicates that $s_n = 1$, $\hat{s}_n = -1$, and $e_n = -2$ indicates that $s_n = -1$ and $\hat{s}_n = 1$.

Define $\mathbf{M} \in \mathbb{R}^{M \times N}$ as the measurement matrix of size $M \times N$, we formulate the decoding model as

$$\mathbf{u} = \mathbf{M}\hat{\mathbf{y}} = \mathbf{M}(\mathbf{e} + \mathbf{w}) = \mathbf{M}\mathbf{e} + \underbrace{\mathbf{M}\mathbf{w}}_{\triangleq \mathbf{w}'}, \tag{6.9}$$

where $\mathbf{u} = [u_1, u_2, \ldots, u_M]$ is the vector of measurements. We can see from the above decoding model that $\mathbf{u}$ is the noiseless measurements of the residual signal

$\hat{\mathbf{y}}$ and is also the noisy measurements of the error vector $\mathbf{e}$. Therefore, to recover the error vector $\mathbf{e}$ from noisy measurements $\mathbf{u}$ is equivalent to recover the residual signal $\hat{\mathbf{y}}$ from the noiseless measurements by considering the given noise $\mathbf{w}'$.

6.4 The Proposed AMP-Aided Decoding Scheme

In this section, we introduce the general framework and the setup of the proposed AMP-aided decoding scheme. Fig. 6.2 illustrates the general flow diagram of the proposed decoding method. After the failure of the initial BP decoding, the reprocessing is activated, where the reliability of the channel output for each v_n is evaluated based on $\mathbf{e}$ by using the AMP algorithm [121]. Then the decoder input sequence $\hat{\mathbf{r}}$ is updated by flipping the signs of the LLR values on the unreliable VNs. The conventional BP decoding is conducted thereafter with $\hat{\mathbf{r}}$ as the decoder input. If a valid codeword is found output the decoded codeword, otherwise a decoding failure is declared.

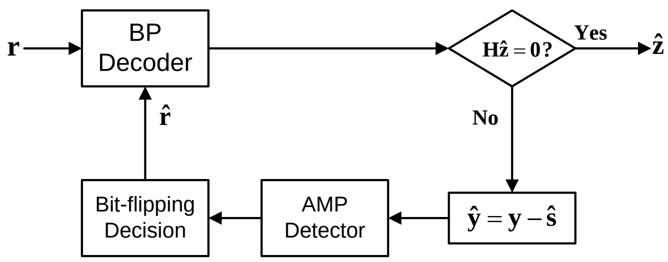


Figure 6.2: The flow diagram of the proposed AMP-aided decoding scheme.

Denoted by $I_{\max}$, the maximum number of iterations for the conventional BP decoding. The proposed AMP-aided decoding scheme is described in **Algorithm 6.1**. In the following, we would discuss the setup of the proposed AMP-aided decoding scheme.

6.4.1 The AMP Detector

Denote the estimate error vector $\hat{\mathbf{e}}$ and the residue vector at the i-th iteration by $\hat{\mathbf{e}}^{(i)}$ and $\mathbf{y}^{(i)}$, respectively, where $\hat{\mathbf{e}}^{(i)} \in \mathbb{R}^N$ and $\mathbf{y}^{(i)} \in \mathbb{R}^M$. The AMP algorithm starts from an initial guess $\hat{\mathbf{e}}^{(0)} = \mathbf{0}$ and the initial residue $\mathbf{y}^{(0)} = \mathbf{u}$. Let n be the index of the column, i.e., $n = 1, 2, \ldots N$. The AMP algorithm proceeds iteratively according to

$$
\begin{aligned}
\hat{\mathbf{e}}^{(i+1)} &= \eta\left(\mathbf{M}^*\mathbf{y}^{(i)} + \hat{\mathbf{e}}^{(i)}\right) \\
\mathbf{y}^{(i)} &= \mathbf{u} - \mathbf{M}\hat{\mathbf{e}}^{(i)} + \frac{1}{\delta}\mathbf{y}^{(i-1)}\left(\sum_{n=1}^{N} \frac{\eta'\left(\mathbf{M}_n^*\mathbf{y}_n^{(i-1)} + \hat{e}_n^{(i-1)}\right)}{N}\right).
\end{aligned} \tag{6.10}
$$

Here $\mathbf{M}_n^*$ represents the transpose of $\mathbf{M}$ and $\eta = M/N$ is known as the undersampling fraction in [121]. Note that $\eta(\cdot)$ is a *threshold* function applied componentwise to the input vector at the i-th iteration and $\eta'(\cdot)$ refers to the first derivative of $\eta(\cdot)$ with respect to the input.

The Measurement matrix M

In the proposed AMP-aided decoding scheme, we set the number of measurements equal to the block lengths of transmitted codewords, which guarantees the sufficient number of measurements for each VN, i.e., $M = N$. This results in a square measurement matrix $\mathbf{M}$ of size $N \times N$. Furthermore, each element in the measurement matrix $\mathbf{M}$ is generated from i.i.d. real Gaussian distribution with zero-mean and variance $1/N$, i.e.,

$$
\theta_{m,n} \sim \mathcal{N}(0, 1/N), m, n \in 1, 2, \ldots, N. \tag{6.11}
$$

Algorithm 6.1 The Proposed AMP-Aided Decoding Algorithm

Inputs: $\mathbf{r}$, $I_{\max}$
Output: decoded codeword $\hat{\mathbf{x}}$

```
Perform BP decoding with I_max.
if Hx̂ = 0 then
    Output the decoded codeword x̂
else
    Perform AMP estimation:
    Compute residue signal ŷ = e + w as in Eq. (6.8)
    Perform AMP detector with M and ŷ to obtain ê
    Flip the sign of r according to ê and generates r̂
    Perform BP decoding with r̂ and I_max
    if Hx̂ = 0 then
        Output the decoded codeword x̂
        return
    else
        Declare decoding fail
    end if
end if
```

The particular choice of $\mathbf{M}$ is convenient for two reasons: first, it guarantees the convergence of the AMP algorithm [121]. Secondly, each column of $\mathbf{M}$ has a unit norm, i.e., $\mathbf{M}^*\mathbf{M}$ is an $N \times N$ identity matrix as the norm of each column of $\mathbf{M}$ is one, which ensures that the vectors $\mathbf{w}' = \mathbf{M}\mathbf{w}$ and $\mathbf{M}\mathbf{e}$ satisfies the restricted isometric property [178].

Soft-threshold function

The *soft-thresholding* [121] is used for the AMP detector, which can be characterized as

$$\eta(y, \tau^{(i)}) = \text{sign}(y) \cdot \max\left(|y| - \tau^{(i)}, 0\right), \tag{6.12}$$

where the state variable at the i-th iteration $\tau^{(i)}$ is empirically computed based on the current residue $\mathbf{y}^{(i)}$ as

$$\tau^{(i)} = \frac{\|\mathbf{y}^{(i)}\|_2}{\sqrt{N}}. \tag{6.13}$$

6.4.2 The Bit-flipping Rule

After we obtain $\hat{\mathbf{e}}$, the update decoder input is generated by flipping the sign of the decoder input sequence $\hat{\mathbf{r}}$ on the unreliable VN positions so that some erroneous VNs can be corrected. As indicated by $\hat{\mathbf{e}}$, the node v_n is considered to be in a decoding error if

$$|\hat{e}_n| > \lambda, \tag{6.14}$$

where λ is a predetermined threshold value optimized empirically.

After determining the VNs in error by (6.14) according to the magnitude of $\hat{e}_n$, these VNs are further selected by considering the sign of $\hat{e}_n$ and $\hat{s}_n$. We consider two ideal cases of decoding errors without the effect of noise in the residue signal. If $\hat{e}_n$ has a '+' sign, it implies that the transmitted s_n should be $+1$ and $\hat{s}_n = -1$ since $e_n = s_n - \hat{s}_n = +1 - (-1) = +2$. While we have $\hat{s}_n = 1$ and $s_n = -1$ if $\hat{e}_n$ has a '$-$' sign. Therefore, for given $\hat{e}_n$ and $\hat{s}_n$, there is a high possibility that the node v_n is unreliable if $\text{sign}(\hat{e}_n) \neq \text{sign}(\hat{s}_n)$.

To further improve the accuracy of selecting unreliable VNs, we also consider the sign of y_n for the decision of the VNs which are satisfied to be flipped under the condition of $\text{sign}(\hat{e}_n) \neq \text{sign}(\hat{s}_n)$. More specifically, the n-th VN will be flipped if $\text{sign}(\hat{e}_n) \neq \text{sign}(y_n)$ since in this case, we have $\text{sign}(y_n) = \text{sign}(\hat{s}_n) \neq \text{sign}(\hat{e}_n)$. This means that a decoding error of $\hat{s}_n$ most likely happens because the sign of the received signal is incorrect. In addition, we consider the case of $\text{sign}(\hat{e}_n) = \text{sign}(y_n)$ given that $\text{sign}(\hat{e}_n) \neq \text{sign}(\hat{s}_n)$. With the existence of channel noise, there is a probability that the received y_n has its sign flipped due to the perturbations of the noise, which results in the sign of y_n the same as that of $\hat{e}_n$. Thus, in our proposed bit-flipping rule, we introduce a predetermined threshold γ to further proceed with the decision for the unreliable VNs. If $|y_n| < \gamma$ and $\text{sign}(\hat{e}_n) = \text{sign}(y_n)$, the sign of y_n also needs to be flipped, where the decoding error in this case is most probably caused by the small magnitude of the received signal y_n.

Based on the above discussions, we propose the bit-flipping rule to update the decoder input sequence, which generates a modified LLR sequence for the decoding test in the reprocessing. The detail of the flipping rule is summarized in **Algorithm** 6.2.

Algorithm 6.2 The Proposed Bit-flipping Algorithm

Inputs: $\mathbf{r}$, $\hat{\mathbf{e}}$

Output: The modified LLR sequence $\hat{\mathbf{r}}$

for $j = 1 : N$ **do**
  if $|\hat{e}_n| > \lambda$ **then**
    if $\text{sign}(\hat{e}_n) \neq \text{sign}(\hat{s}_n)$ **and** $\text{sign}(\hat{e}_n) \neq \text{sign}(y_n)$ **then**
      $\hat{r}(v_n) = -r(v_n)$
    end if
    if $\text{sign}(\hat{e}_n) \neq \text{sign}(\hat{s}_n)$ **and** $\text{sign}(\hat{e}_n) = \text{sign}(y_n)$ **then**
      if $|y_n| < \gamma$ **then**
        $\hat{r}(v_n) = -r(v_n)$
      end if
    end if
  end if
end for

Note that the QML decoding methods, like the saturate min-sum decoding scheme in [109], may select more than one unreliable VNs at a time based on the node selection methods and conducts multiple decoding tests, which attempts to obtain a better error performance by sacrificing the decoding complexity. However, the AMP-aided decoding scheme flips multiple unreliable VNs at one time according to the AMP detector while there is only one decoding test performed in the reprocessing, which reduces the decoding complexity compared to the QML decoding methods based on listing all possible combinations for part of the codeword.

6.5 Performance Analysis of the AMP-Aided Decoding Scheme

6.5.1 The FER Performance of the AMP-Aided Decoding Scheme

Table 6.1: Simulation Setup and Parameters Settings

Channel	AWGN
Modulation	QPSK
Decoding Algorithms	SPA 2D-SC MSA Layered 2D-SC MSA
Algorithm Coefficients	$(\alpha, \beta) = (0.75, 1.25)$
Maximum BP Iterations	50
Maximum AMP Itertions	100
AMP Algorithm	soft-thresholding $\eta(y, \lambda)$
AMP Decision Threshold	$\lambda = 1.5$, $\gamma = 0.5$
Measurement Matrix Size	$N \times N$
Information Bit Lengths K	56, 120, 320, 752
Code Rates R	1/5, 1/3, 2/5, 1/2

In this section, we first show frame error rate (FER) performance of the AMP-aided decoding scheme for the 5G LDPC codes with different decoding algorithms under AWGN channels. We present the FER for the 2D-SC MSA proposed in the previous chapter, the layered 2D-SC MSA, and the AMP-aided 2D-SC MSA and layered 2D-SC MSA. For comparison, we also demonstrate the error performance of the SPA. For clear illustration and consistency, in all the figures presented in this section, we denote AMP-aided 2D-SC MSA as 'AMP-2D-SC MSA' and AMP-aided layered 2D-SC MSA as 'AMP-L2D-SC MSA', and the conventional 2D-SC MSA and layered 2D-SC MSA as '2D-SC MSA' and 'L2D-SC MSA', respectively. The code rates of the 5G LDPC codes presented in the simulations are $R = 1/5, 1/3, 2/5, 1/2$. The decision threshold for the bit-flipping rule in the AMP detector is empirically optimized as $\lambda = 1.5$ and $\gamma = 0.5$, respectively. The code lengths and rates simulated are $K = 56, 120, 320, 752$ and $R = 1/5, 1/3, 2/5, 1/2$. It is worth to mention that in our simulation environment, we construct the measurement matrix $\mathbf{M}$ beforehand for different K, and use these measurement matrices for multiple decodings of different channel realizations in a range of SNR. This will make no difference in terms of average error rate compared to the scenario where a new measurement matrix $\mathbf{M}$ is generated for very channel realization. A complete list of simulation setup and parameters is given in TABLE 6.1.

Simulation results of 5G LDPC codes with $R = 1/5, K = 56, 120, 320, 752$

Figures 6.3 - 6.6 illustrate the FER performance of rate 1/5 BG2 LDPC codes for information lengths $K = 56, 120, 320$ and 752, respectively. From the figures, we observe that the AMP-2D-SC MSA and AMP-L2D-SC MSA outperform their counterpart approximately 0.1 dB for all K. Compared to SPA, the improvement of AMP-L2D-SC MSA is approximately 0.3 dB at about FER $10^{-2} \sim 10^{-4}$ for information length $K = 56$, where the improvement of AMP-2D-SC MSA over SPA is approximately 0.15 dB at about FER $10^{-2} \sim 10^{-4}$. Such an improvement over SPA reduces as K increases. For instance, for $K = 120$, the gain for

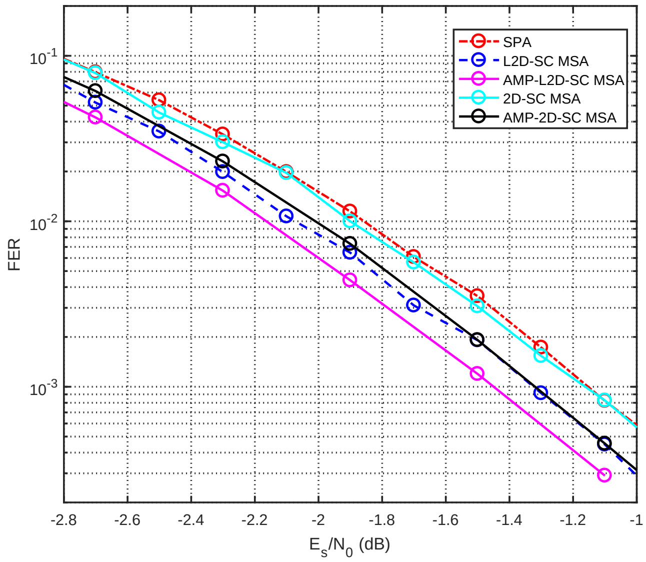


Figure 6.3: FER for BG2 LDPC code with $R = 1/5$, $K = 56$.

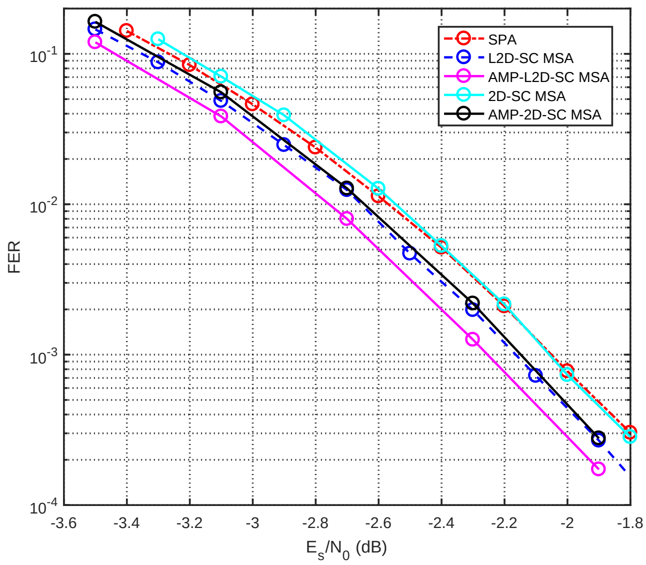


Figure 6.4: FER for BG2 LDPC code with $R = 1/5$, $K = 120$.

AMP-L2D-SC MSA is approximately 0.2 dB at about FER $10^{-2} \sim 10^{-4}$ and the gain for AMP-2D-SC MSA is about 0.1 dB at about FER $10^{-2} \sim 10^{-4}$. For $K = 320$, the gain for AMP-LMS is approximately 0.1 dB at about FER 10^{-4}, and no gain for AMP-2D-SC MSA compared to the performance of SPA. For $K = 752$, the performance of AMP-L2D-SC MSA approaches the performance of

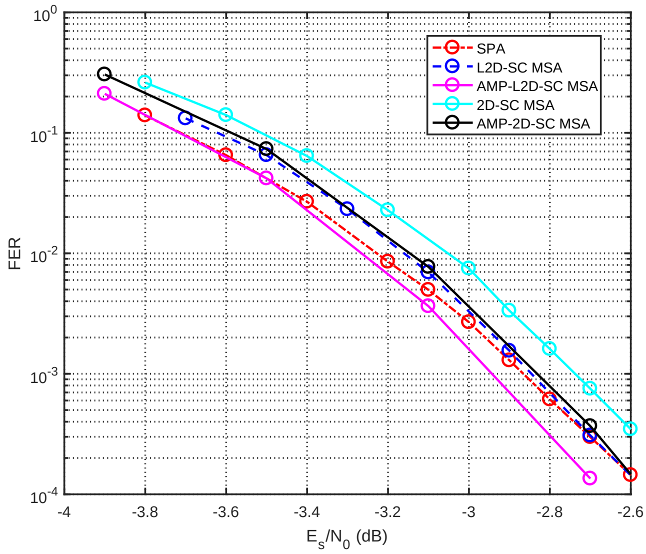


Figure 6.5: FER for BG2 LDPC code with $R = 1/5$, $K = 320$.

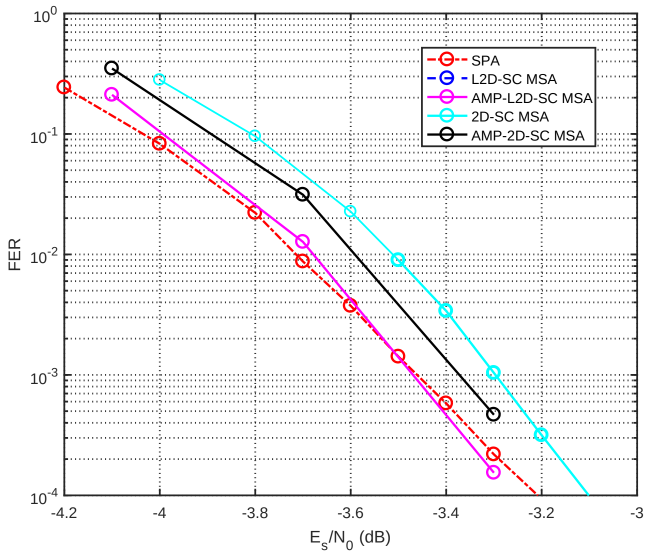


Figure 6.6: FER for BG2 LDPC code with $R = 1/5$, $K = 752$.

SPA and outperforms SPA in error floor region. Furthermore, the performance of AMP-L2D-SC MSA outperforms AMP-2D-SC MSA by approximately 0.1 dB at about FER $10^{-2} \sim 10^{-4}$ for $K = 56, 120, 320$ and 752. Also from the figure, instead of using layered BP decoding, the performance of AMP-2D-SC MSA achieves that of L2D-SC MSA decoding for $K = 56, 120$ and 320.

Simulation results of 5G-LDPC codes with $R = 1/3, K = 56, 120, 320, 752$

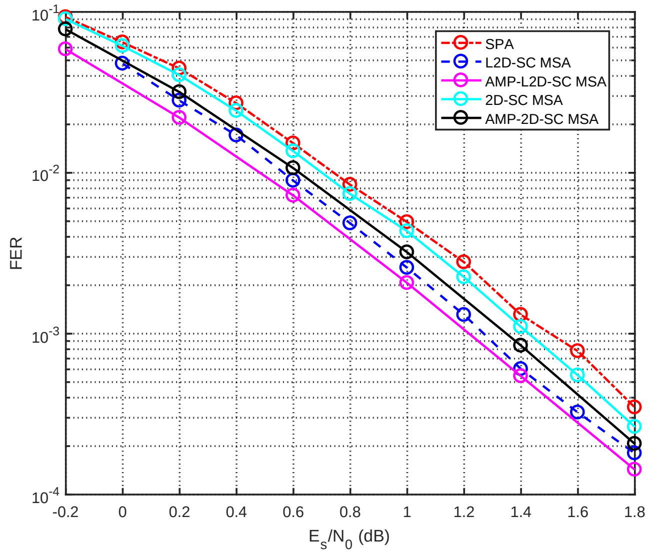


Figure 6.7: FER for BG2 LDPC code with $R = 1/3$, $K = 56$.

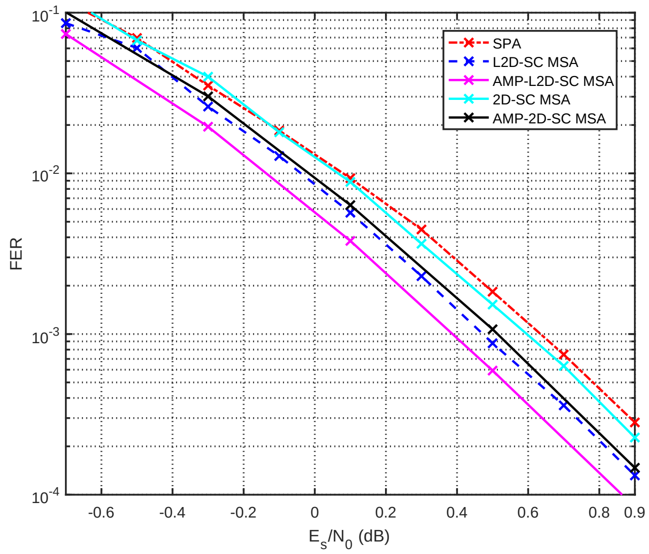


Figure 6.8: FER for BG2 LDPC code with $R = 1/3$, $K = 120$.

Figures 6.7 - 6.10 illustrate the FER performance of rate 1/3 BG2 LDPC codes for information lengths $K = 56, 120, 320$ and 752, respectively. The AMP-2D-SC MSA and AMP-L2D-SC MSA outperforms their conventional counterpart by approximately $0.05 \sim 0.1$ dB for $K = 56$, whereas the performance gain for

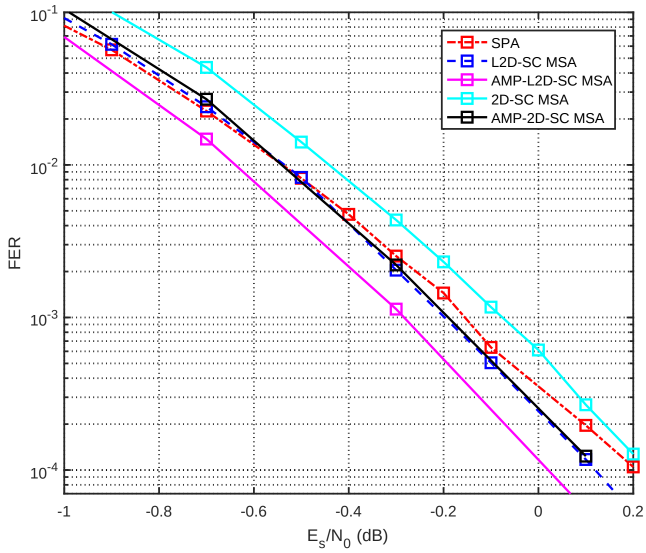


Figure 6.9: FER for BG2 LDPC code with $R = 1/3$, $K = 320$.

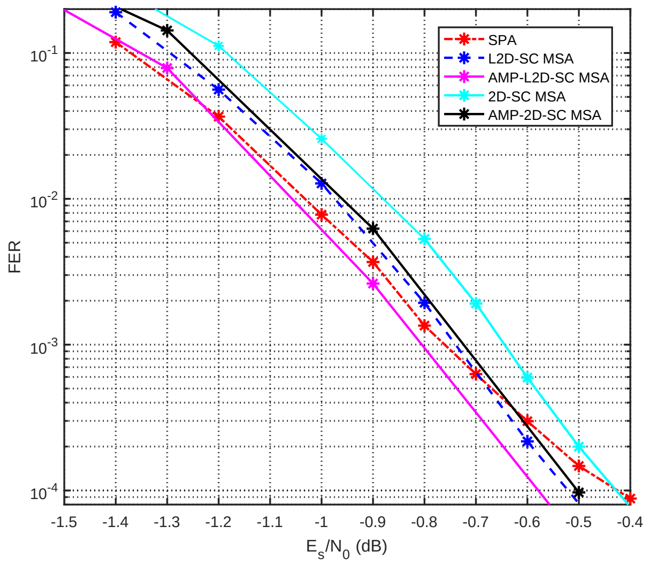


Figure 6.10: FER for BG2 LDPC code with $R = 1/3$, $K = 752$.

$K = 120, 320$ and 752 is 0.1 dB. In addition, the improvement of AMP-2D-SC MSA over 2D-SC MSA is higher compared to that of AMP-2D-SC MSA over L2D-SC MSA, in particularly, for large K. Compared to SPA, the performance of AMP-L2D-SC MSA shows an approximately 0.3 dB gain for $K = 56$ at FER $10^{-2} \sim 10^{-4}$, and the gain reduces as K increases. Morever, for $K = 320$ and

752, the performance of both AMP-L2D-SC MSA and AMP-2D-SC MSA show no error floor at FER 10^{-4}, which outperforms SPA in the error floor region.

Simulation results of 5G-LDPC codes with $R = 2/5, K = 56, 120, 320, 752$

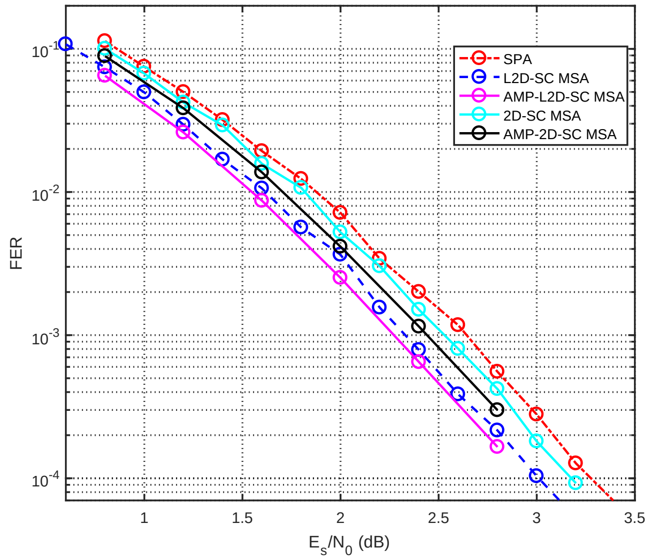


Figure 6.11: FER for BG2 LDPC code with $R = 2/5$, $K = 56$.

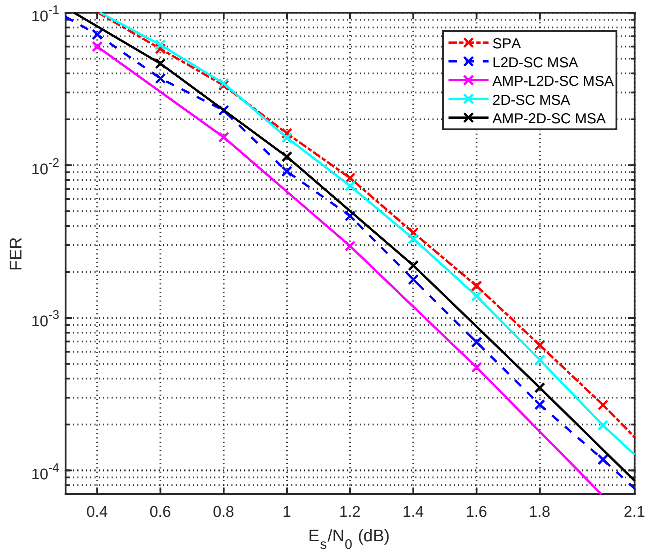


Figure 6.12: FER for BG2 LDPC code with $R = 2/5$, $K = 120$.

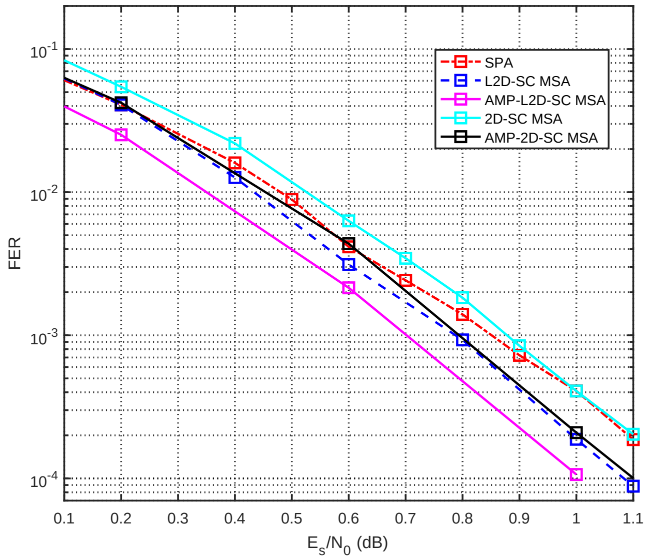


Figure 6.13: FER for BG2 LDPC code with $R = 2/5$, $K = 320$.

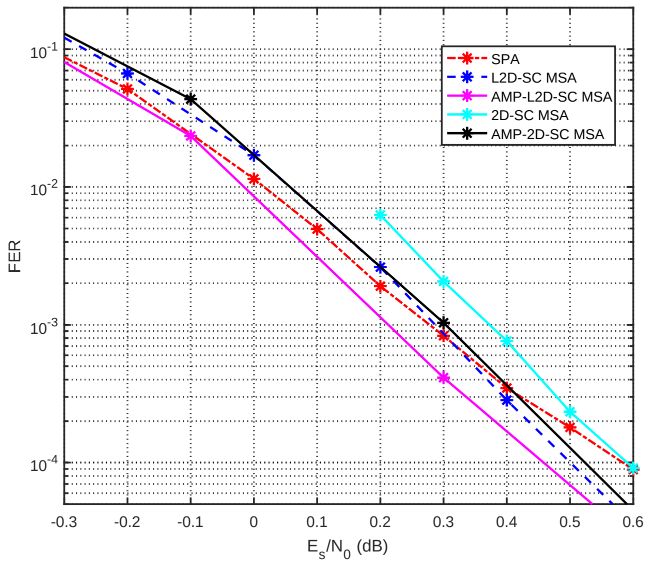


Figure 6.14: FER for BG2 LDPC code with $R = 2/5$, $K = 752$.

Figures 6.11 - 6.14 illustrate the FER performance of rate 2/5 BG2 LDPC codes for information lengths $K = 56, 120, 320$ and 752, respectively. From the figures, for $K = 120, 320$ and 752, the AMP-2D-SC MSA and AMP-L2D-SC MSA outperforms their conventional counterpart by approximately 0.1 dB, whereas for $K = 56$ the improvement is less than 0.05 dB.

Simulation results of 5G-LDPC codes with $R = 1/2, K = 56, 120, 320752$

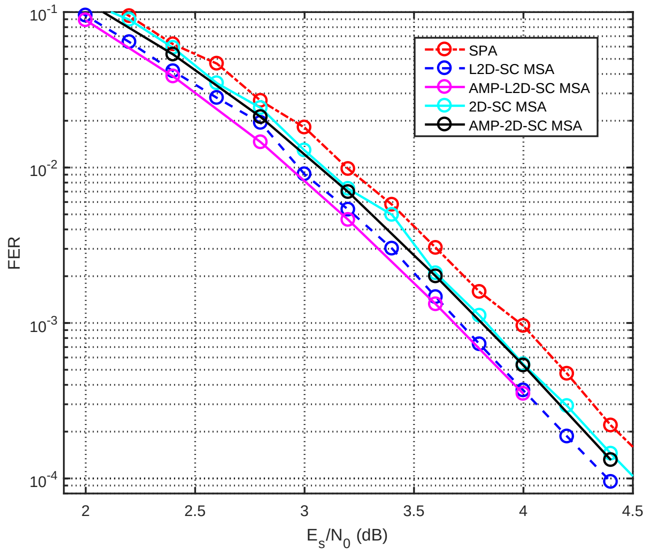


Figure 6.15: FER for BG2 LDPC code with $R = 1/2$, $K = 56$.

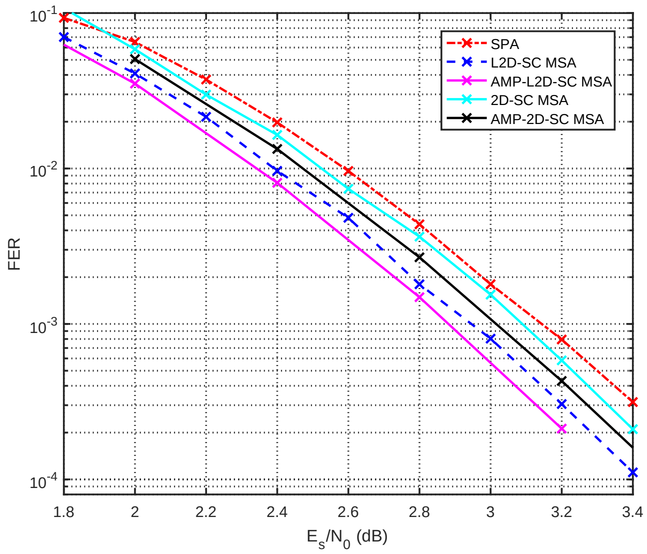


Figure 6.16: FER for BG2 LDPC code with $R = 1/2$, $K = 120$.

Figures 6.15 - 6.18 illustrate the FER performance of rate 1/2 BG2 LDPC codes for information lengths $K = 56, 120, 320$ and 752, respectively. The AMP-2D-SC MSA and AMP-L2D-SC MSA is similar to their conventional counterpart for $K = 56$ and has an approximately 0.05 dB gain for $K = 120$. For $K = 320$

and 752, there is approximately 0.1 dB gain for both AMP-2D-SC MSA and AMP-L2D-SC MSA compared to their conventional counterpart.

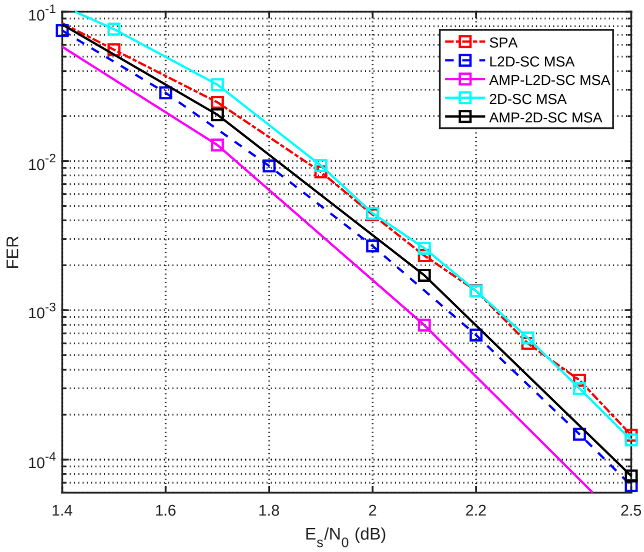


Figure 6.17: FER for BG2 LDPC code with $R = 1/2$, $K = 320$.

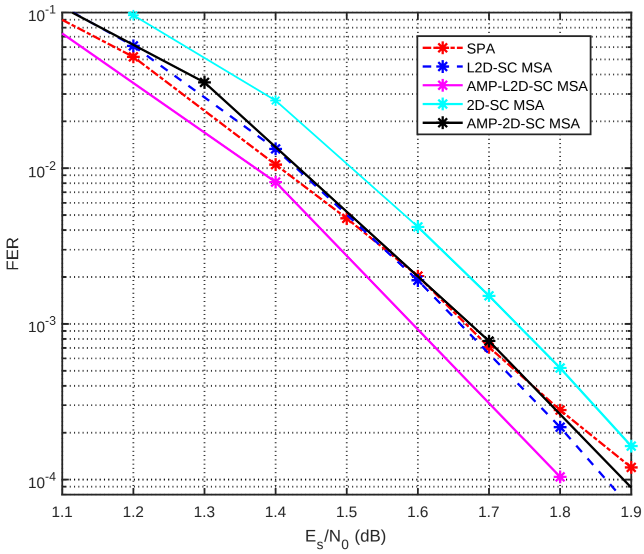


Figure 6.18: FER for BG2 LDPC code with $R = 1/2$, $K = 752$.

6.5.2 Other Properties of the AMP-Aided Decoding Scheme

In addition to FER performance, we further investigate other properties with respect to the performance of AMP-aided decoding scheme such as false flip rate (FFR), the denoiser success rate over total number of decoding failure (DSRF) and the denoiser success rate over total transmissions (DSRT). The definition of each metric is described in below.

Let $\mathcal{F} = \{v_n | n \in \{1, 2, \ldots, N\}\}$ be the set of VNs been flipped according to the decision rule. The FFR for given thresholds λ and γ is defined as

$$\text{FFR} = \frac{|\{n|\text{sign}(\hat{r}(v_n)) \neq \text{sign}(x_n), \forall n \in \{1, 2, \ldots, N\}\}|}{|\mathcal{F}|}, \tag{6.15}$$

where $|\mathcal{F}|$ is the cardinality of set $\mathcal{F}$ and x_n is the modulated bit been transmitted. It measures the number of times of the bits being flipped incorrectly compared to the transmitted bits. Hence, it shows the amount of incorrectly received bits that are corrected before feeding into the decoder for the reprocessing.

Furthermore, let $\mathbb{S}_D$ be the number of times the second round of decoding test succeeds after the AMP detector, and define $\mathbb{F}_D$ as the number of times the second round of decoding test fails. Note that $\mathbb{F}_D$ is equivalent to the total number of erroneous frames collected as given in Table 6.1. We define the AMP successful rate in terms of the total times of decoding failure as

$$\text{DSRF} = \frac{\mathbb{S}_D}{\mathbb{F}_D + \mathbb{S}_D}. \tag{6.16}$$

Similarly, the AMP successful rate in terms of the total number of transmissions is defined as

$$\text{DSRT} = \frac{\mathbb{S}_D}{\text{Total number of transmissions}}. \tag{6.17}$$

We would show these proprieties of the AMP-aided decoding scheme in the following.

Simulation results of 5G-LDPC codes with $R = 1/5, K = 56, 120, 320, 752$

Figure 6.19 illustrates the relation between SNR and FFR for different K. From the figure, the FFR decreases as SNR increases. This is reasonable as the noise power reduces, with the same thresholds λ and γ, the accuracy of $\hat{\mathbf{e}}$ becomes higher. As a consequence, the probability of decoding success after the second round of BP decoding is increased. Note that the decreasing rate of FFR with respect to SNR is more effective for $K = 120, 320$ and 752 compared to $K = 56$, where the FFR for $K = 56$ reaches a minimal point at about $Es/No = -1.5$ dB.

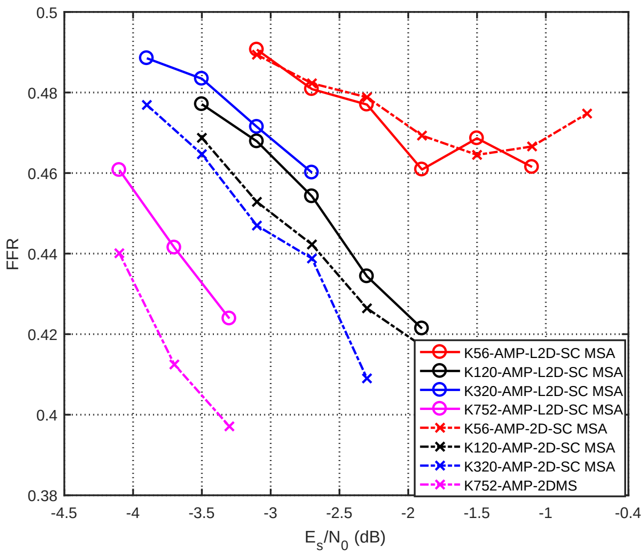


Figure 6.19: FFR of AMP-aided decoding scheme for BG2 LDPC codes with $R = 1/5$, $K = 56, 120, 320, 752$.

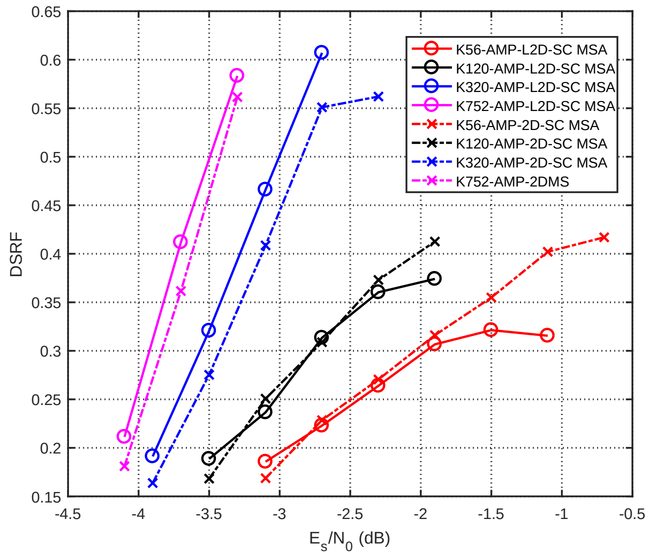


Figure 6.20: DSRF of AMP-aided decoding scheme for BG2 LDPC codes with $R = 1/5$, $K = 56, 120, 320, 752$.

Figure 6.20 shows the DSRF performance for different K. It can be seen from the figure that the DSRF increases as SNR increases for all K and for both AMP-LMS and AMP-2DMS. The DSRF reaches a peak point for $K = 56$ and $K = 120$ for both AMP-2D-SC MSA and AMP-L2D-SC MSA, while the DSRF starts to reach the peak point for $K = 320$. Moreover, for $K = 56$ and $K = 120$, the peak value of DSRF for AMP-2D-SC MSA is higher than that of AMP-L2D-SC MSA. This implies that with the assistant of AMP detector, the conventional 2D-SC MSA decoding for short LDPC codes has a higher probability of decoding success in the second round of decoding test than that of AMP-L2D-SC MSA.

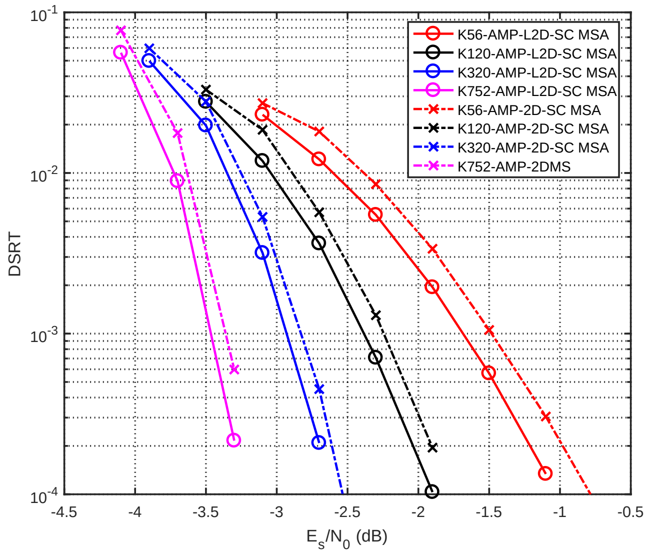


Figure 6.21: DSRT of AMP-aided decoding for BG2 LDPC codes with $R = 1/5$, $K = 56, 120, 320, 752$.

The DSRT for rate 1/5 LDPC codes with $K = 56, 120, 320$ and 752 is plotted in Figure 6.21. From the figure, the DSRT for all K exponentially decrease as SNR increases, and the DSRT for AMP-2D-SC MSA shows a constant gain compared to that of AMP-L2D-SC MSA for all SNR ranges.

Simulation results of 5G-LDPC codes with $R = 1/3, K = 56, 120, 320, 752$

The FFR plotted in 6.22 shows that as SNR increases, the number of times the false flip happened is decreasing. Hence, more correct bits are provided as the input to the BP decoder in the second round of decoding. Moreover, as also can be seen from the figure, the FFR has a minimal point, for which the FFR will start to increase again once the SNR value exceeds this point. We can see that the FFR of AMP-2D-SC MSA reaches the minimal point at a faster rate compared to that of AMP-L2D-SC MSA. In addition, the FFR of AMP-L2D-SC MSA tends to have a lower minimal point compared to that of AMP-2D-SC MSA.

The DSRF, illustrated in Figure 6.23, for rate 1/3 LDPC codes shows that

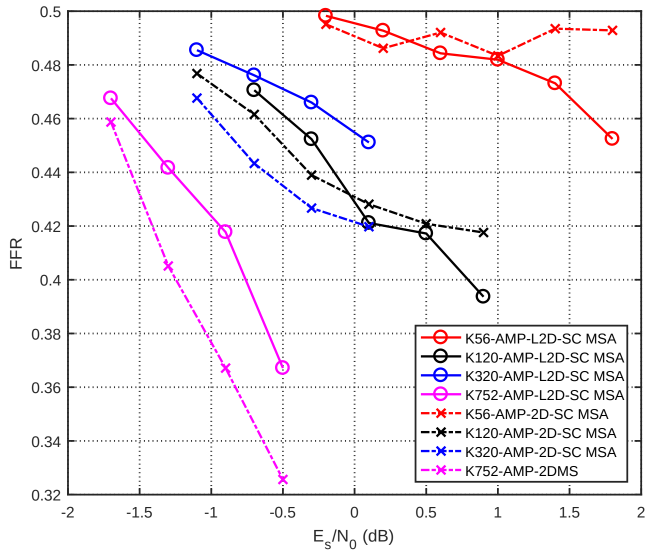


Figure 6.22: FFR of AMP-aided decoding scheme for BG2 LDPC codes with $R = 1/3$, $K = 56, 120, 320, 752$.

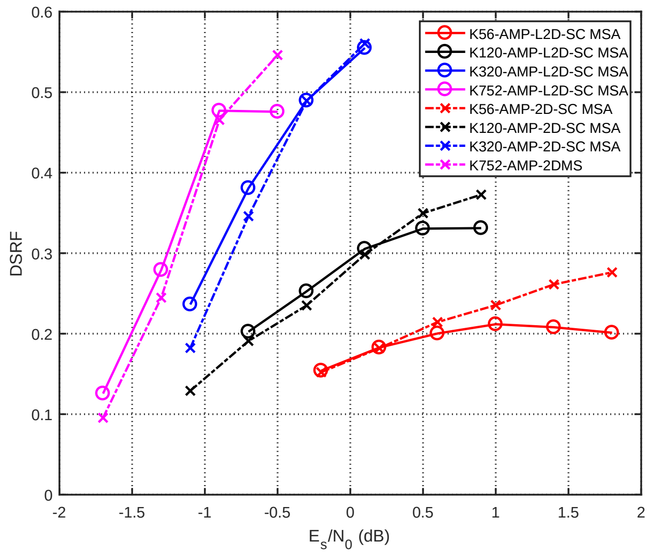


Figure 6.23: DSRF of AMP-aided decoding for BG2 LDPC codes with $R = 1/3$, $K = 56, 120, 320, 752$.

the detector success rate is approximately the same for both AMP-2D-SC MSA and AMP-L2D-SC MSA. In addition, as can be seen from the figure, for all K, the DSRF for AMP-L2D-SC MSA is higher than that of AMP-2D-SC MSA in a range of SNR and drop below to the DSRF of AMP-2D-SC MSAS as SNR passes a point. Moreover, for large K, the DSRF for AMP-2D-SC MSA and AMP-L2D-SC MSA is approximately 0.5 and more, which indicates that the AMP detector reduces the FER by at least a half for the LDPC codes with block lengths at least $K = 320$.

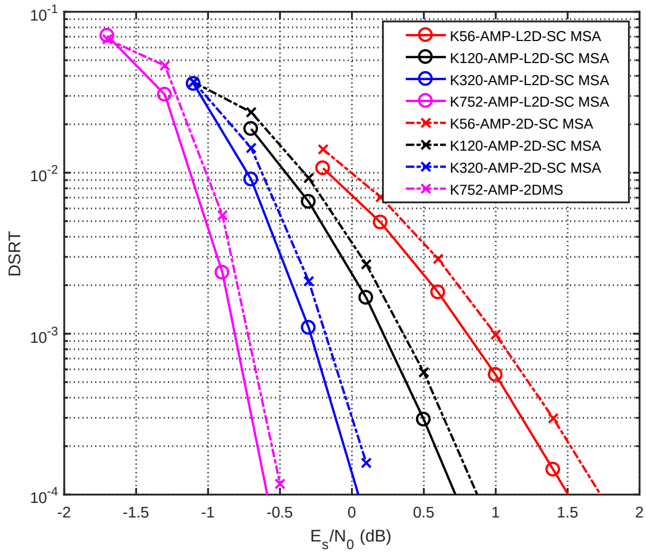


Figure 6.24: DSRT of AMP-aided decoding for BG2 LDPC codes with $R = 1/3$, $K = 56, 120, 320, 752$.

The DSRT for rate 1/3 LDPC codes with $K = 56, 120, 320$ and 752 is plotted in Figure 6.24. It shows that the DSRT for all K exponentially decreases as SNR increases. Furthermore, the DSRT for AMP-2D-SC MSA shows a constant gain compared to that of AMP-L2D-SC MSA for all SNR ranges, where the gain increases when K decreases.

Simulation results of 5G-LDPC codes with $R = 2/5, K = 56, 120, 320, 752$

The FFR plotted in 6.25 shows that within the provided SNR ranges, the minimal point is shown for codes with $K = 56$ and $K = 120$. It is shown that the FFR of AMP-2D-SC MSA reaches the minimal point at a faster rate compared to that of AMP-L2D-SC MSA. In addition, the FFR of AMP-L2D-SC MSA tends to have a lower minimal point compared to that of AMP-2D-SC MSA. Similar to the

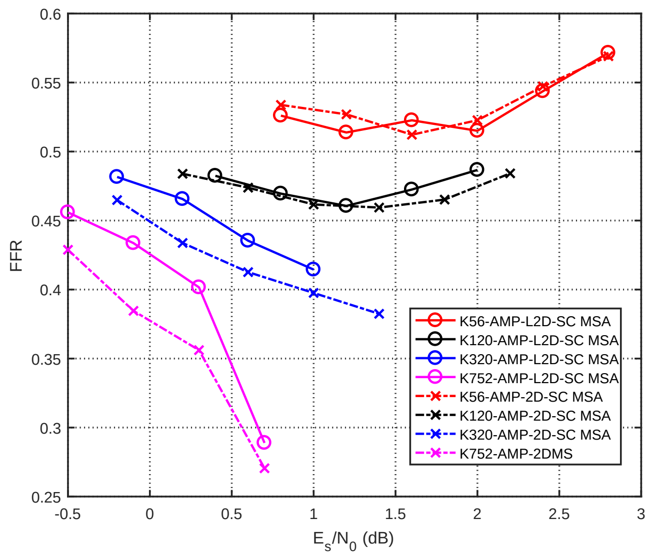


Figure 6.25: FFR of AMP-aided decoding for BG2 LDPC codes with $R = 2/5$, $K = 56, 120, 320, 752$.

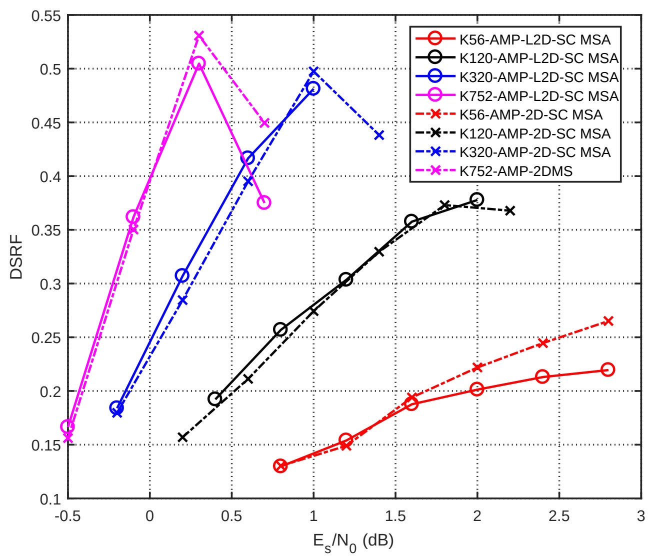


Figure 6.26: DSRF of AMP-aided decoding for BG2 LDPC codes with $R = 2/5$, $K = 56, 120, 320, 752$.

rate 1/3 codes, the DSRF, illustrated in Figure 6.26, for rate 2/5 LDPC codes shows that the detector success rate for AMP-2D-SC MSA and AMP-L2D-SC MSA is approximately the same in a range of SNR for all K. The DSRF shows a peak point for $K = 320$ and $K = 752$, and start reaching a peak for $K = 56$ and $K = 120$. Moreover, the peak point for AMP-2D-SC MSA is higher than that of AMP-L2D-SC MSA, and the peak value reduced compared to the peak value for codes with rate 1/3.

The DSRT for rate 2/5 LDPC codes with $K = 56, 120, 320$ and 752 is plotted in Figure 6.27. It is shown that the DSRT for all K exponentially decreases as SNR increases, and the DSRT for AMP-2D-SC MSA shows a constant gain compared to that of AMP-L2D-SC MSA for all SNR ranges, where the gain also increases when K decreases.

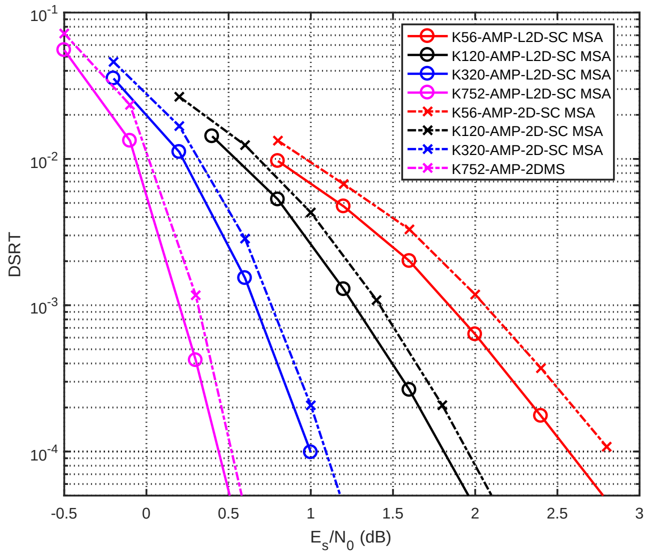


Figure 6.27: DSRT of AMP-aided decoding for BG2 LDPC codes with $R = 2/5$, $K = 56, 120, 320, 752$.

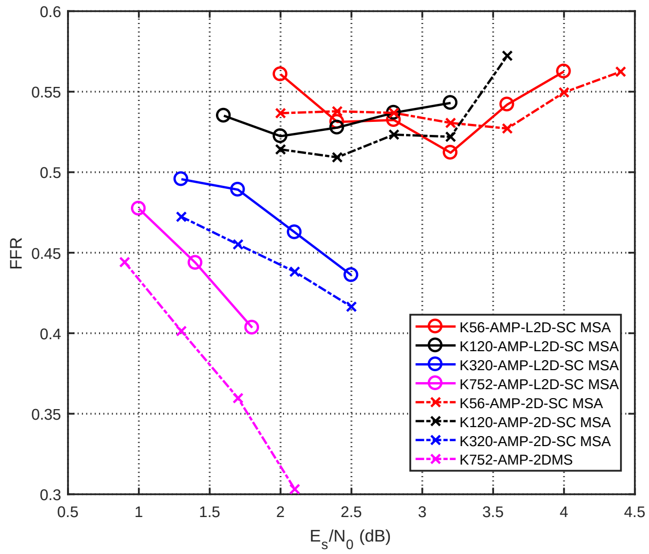


Figure 6.28: FFR of AMP-aided decoding for BG2 LDPC codes with $R = 1/2$, $K = 56, 120, 320, 752$.

Simulation results of 5G-LDPC codes with $R = 1/2, K = 56, 120, 320, 752$

Similar to the performance of other code rates, the FFR presented in 6.28 shows that within the provided SNR ranges, the minimal point is shown for both codes with $K = 56$ and $K = 120$. In addition, the FFR of AMP-2D-SC MSA is lower than the FFR of AMP-L2D-SC MSA.

As illustrated in Figure 6.29, the DSRF for rate 1/2 LDPC codes shows that the detector success rate for AMP-2D-SC MSA and AMP-L2D-SC MSA is almost the same for $K = 120, 320$ and 752. Also, the peak point for all K is further reduced compared to rate-2/5 codes, and the peak point is almost the same for both AMP-2D-SC MSA and AMP-L2D-SC MSA.

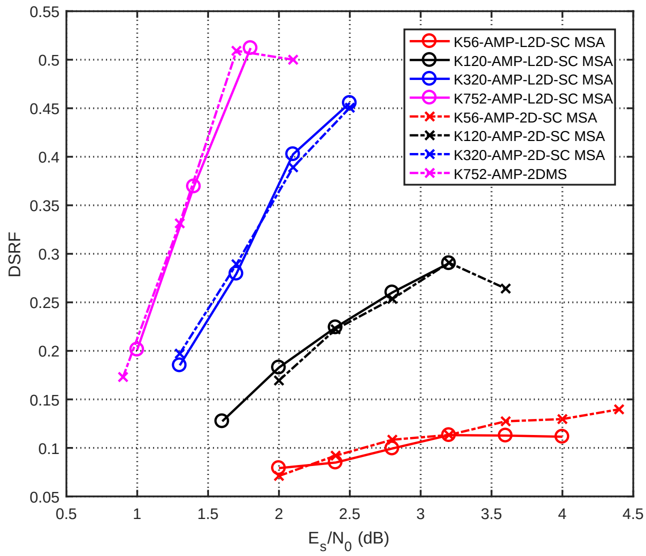


Figure 6.29: DSRF of AMP-aided decoding for BG2 LDPC codes with $R = 1/2$, $K = 56, 120, 320, 752$.

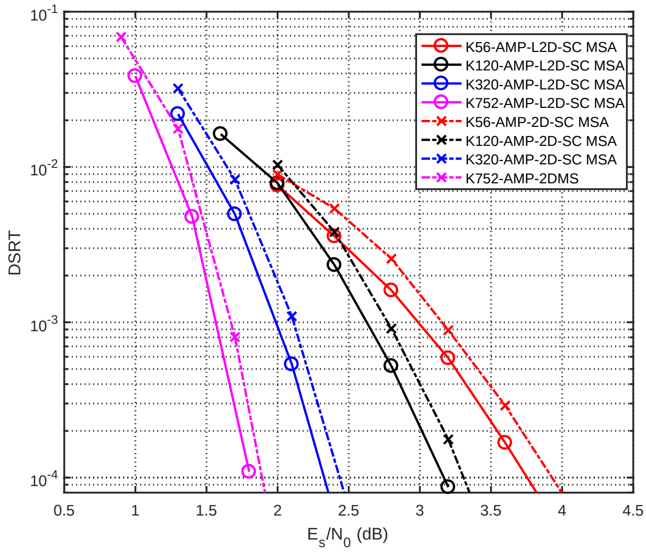


Figure 6.30: DSRT of AMP-aided decoding for BG2 LDPC codes with $R = 1/2$, $K = 56, 120, 320, 752$.

The DSRT for rate 2/5 LDPC codes with $K = 56, 120, 320$ and 752 is demonstrated in Figure 6.30. We can see that the DSRT for all K exponentially decreases as SNR increases, and the DSRT for AMP-2D-SC MSA shows a constant gain compared to that of AMP-L2D-SC MSA for all SNR ranges. In addition, the gain increases as K decreases.

6.5.3 Discussions about the AMP-Aided Decoding Scheme

As shown in Figure 6.5.1-6.5.1, several points could be drawn based on some observations. The first point is that the relation between FER, DSRF and DSRT. Given that the FER is computed as

$$\text{FER} = \frac{\mathbb{F}_D}{\text{Total number of transmissions}}. \tag{6.18}$$

The radio between DSRT and DSRF is

$$\frac{\text{DSRT}}{\text{DSRF}} = \frac{\frac{\mathbb{S}_D}{\text{Total number of transmissions}}}{\frac{\mathbb{S}_D}{\mathbb{F}_D+\mathbb{S}_D}} = \frac{\mathbb{F}_D + \mathbb{S}_D}{\text{Total number of transmissions}} = \text{FER} + \text{DSRT}. \tag{6.19}$$

Note that the result of Eq. 6.19 is equivalent to the FER of the conventional BP decoding without the AMP detector. Thus, DSRT is the amount of gain in terms of FER that a conventional BP decoding algorithm can get after applying the AMP detector.

Furthermore, after rearrange (6.19), FER can be represented as

$$\text{FER} = \frac{\text{DSRT}(1 - \text{DSRF})}{\text{DSRF}}. \tag{6.20}$$

This implies that to be able to reduce the FER for a given SNR, the most straightforward way is to increase the DSRF. However, several aspects also affect the DSRF. For instance, the AMP algorithm and the thresholds λ and γ for decision are important aspects that affect the DSRF significantly.

Based on the above discussions and the simulations results of DSRT, it can be deduced that the amount of coding gain achieved by the 2D-SC MSA is higher than that of L2D-SC MSA when using the AMP-aided decoding scheme. Particularly, for a fixed R, the shorter the block length, larger the gain that 2D-SC MSA achieved from AMP detector than that of L2D-SC MSA. Furthermore, the proposed AMP-aided decoding scheme is more effective for BG2 5G LDPC codes with small R. This can be seen from the DSRF plot, where for the 5G LDPC codes with low code rate R, the successful probability of the denoiser is the highest compared to others with the same information bit lengths K and a higher code rate R.

6.6 The AMP-EQML Decoding Scheme

In this section, we propose an AMP-aided QML decoding scheme for the 5G LDPC codes. We know that the QML decoding fails if and only if all the decoding tests in the reprocessing fails, which means there is no valid codeword found and an empty set is obtained thereafter. Due to further simulations, it is noticeable that there exists a large number of unsuccessful decoding tests in the reprocessing, especially when the information bit lengths K of the 5G LDPC codes increases. Thus, the probability of obtaining an empty set becomes higher.

To verify this suspect, we show in Figure 6.31 the probability of the QML decoding scheme outputs a valid codeword compared to that of the QML decoding scheme outputs an empty set in terms of the total number of conducted decoding tests. The code used here to obtain these results is the 5G LDPC code with information bit lengths $K = 320$ and the code rate $R = 0.2$. The QML decoding method used here is the proposed EQML decoding method in Chapter 5 with L2D-SC MSA, and the AMP detector applied in the simulations is the same as the one proposed in Section 6.4. We can see from the figure that the probability of obtaining an empty set is much higher than that of obtaining a valid codeword, and it increases with the increasing of SNR.

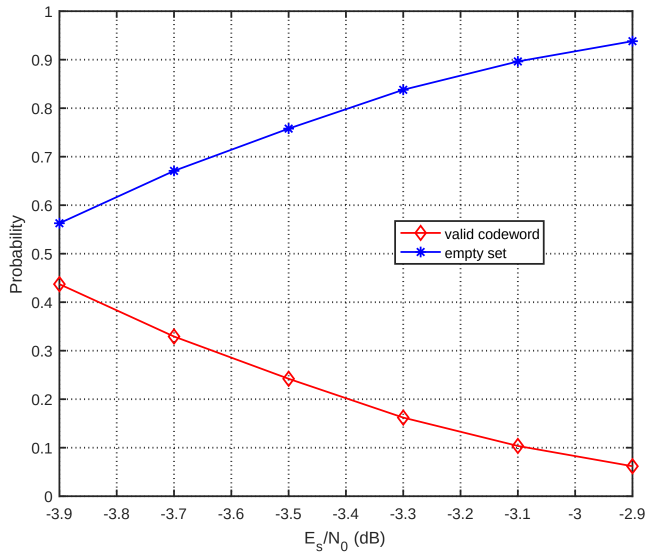


Figure 6.31: The probability of the QML decoding scheme outputs a valid codeword versus an empty set.

6.6.1 The AMP-Aided Post-Processing

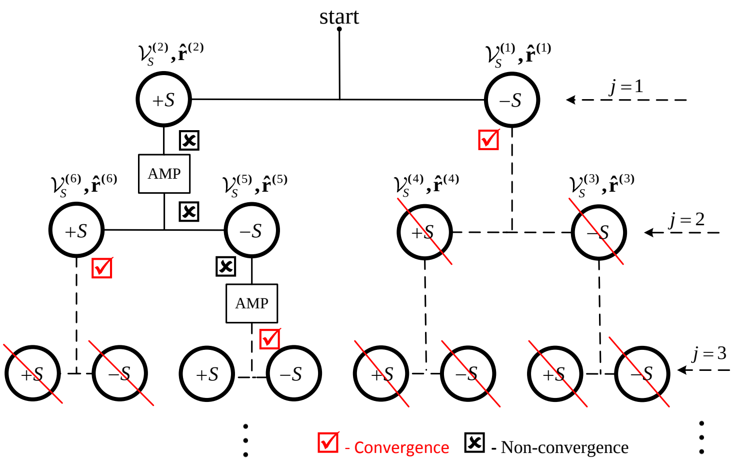


Figure 6.32: The general tree of the AMP-EQML decoding scheme.

Motivated by this observation, we introduce the AMP-aided post-processing for each unsuccessful decoding test to increase the chances of obtaining a valid codeword, so-called the AMP-EQML decoding scheme More specifically, the AMP detector is adopted after the failure of the decoding test in the EQML reprocessing, where one more decoding test is performed as indicated by Fig. 6.2. If the decoder outputs a valid codeword, we save it in the candidate set for a final decision.

Figure 6.32 shows the general tree of the proposed AMP-EQML decoding scheme, where the AMP detector is performed after the failure of the 2nd and the 5th decoding test. To appropriately reduce the decoding latency, we extend the PPS rule to the post-processing, which stops the remaining decoding tests on the associated sub-branches if a valid codeword is obtained by the AMP-aided post-processing. Note that we make sign flip decisions in the AMP detector without considering the saturated VNs since they are regarded as corrected bits in the reprocessing. The proposed AMP-EQML decoding scheme is described in **Algorithm 6.3**.

To verify the benefit from the AMP detector to the EQML decoding method, we show the probability of empty set by using the proposed AMP-EQML decoding scheme and compared with that by using the EMQL decoding method in Figure 6.33. It can be seen that the probability of obtaining the empty set reduces by approximately 5% when the AMP detector is adopted. Therefore, we can see that the AMP detector plays an important role in increasing the probability of decoding success in the reprocessing of the EQML decoding, where there are more candidate codewords generated for the selection of the best codeword. Hence, the probability of obtaining the correct codeword also increases consequently, which introduces addition performance gain.

Algorithm 6.3 The AMP-EQML Decoding Scheme

Perform BP decoding with $I_{\max}$
if a valid codeword is found **then**
  Output the codeword
else
  Initialize: $T = 1, j = 1, \mathcal{V}_S^{(0)} = \emptyset, T_F = 2^{j_{\max}+1} - 2$
  while $j \leq j_{\max}$ **do**
    for $t = 1 : 2^j$ **do**
      $T' = \left\lfloor \frac{T-1}{2} \right\rfloor$
      for $n = 1 : N$ **do**
        Compute $w^{(T')}(v_n)$ according to Eq. (5.5)
      end for
      Select $v_s^{(T)}$ as $v_s^{(T)} = \underset{1 \leq n \leq N}{\arg\max}\, w^{(T')}(v_n)$
      Determine $\mathcal{V_S}^{(T)} = \{\mathcal{V}_S^{(T')} \cup v_s^{(T)}\}$
      Generate $\mathbb{M}$ by enumerating $\mathbf{r}(\mathcal{V}_S^{(T)})$ to $\pm S$
      Determine $\hat{\mathbf{r}}^{(T)}$ by setting $\mathbf{r}(\mathcal{V}_S^{(T)}) = \mathbf{m}_t$
      Perform BP decoding with $\hat{\mathbf{r}}^{(T)}$ and $I_{\max}$
      if A valid codeword found **then**
        Save output codeword $\mathbf{x}^{(T)}$ in $\mathcal{X}$
        Perform PPS rule
      else
        Perform AMP decoding scheme with $\hat{\mathbf{r}}^{(T)}$
        if A valid codeword found **then**
          Save output codeword as $\mathbf{x}^{(T)}$ in $\mathcal{X}$
          Perform PPS rule
        end if
      end if
      $T = T + 1$
    end for
    $j = j + 1$
  end while
  if $\mathcal{X} \neq \emptyset$ **then**
    Output $\mathbf{x}_{best} = \underset{\mathbf{x}^{(T)} \in \mathcal{X}}{\arg\min} \sum_{n=1}^{N} \left| r(v_n) - x_n{}^{(T)} \right|^2$
  else
    Declare decoding failure.
  end if
end if

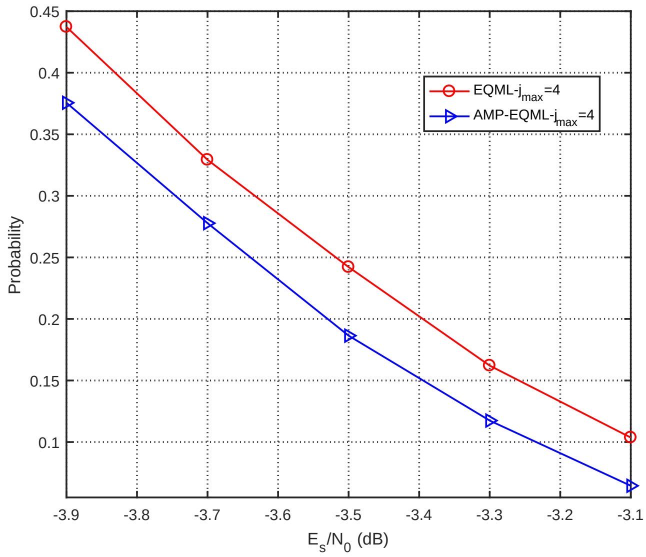


Figure 6.33: The comparison of probability for the QML decoding scheme and the proposed AMP-EQML decoding scheme outputs an empty set.

6.6.2 FER Performance of the AMP-EQML Decoding Scheme

In this section, we present the simulation results of the proposed AMP-EQML decoding scheme for the 5G LDPC codes with information bit lengths $K = 320, 752$ and code rates $R = 1/5, 1/3, 2/5$ and $1/2$, respectively. For the simulation setup, it can be referred to Table 6.1.

Simulation results for codes with $K = 320$

Figure 6.34 to 6.37 show the FER performance of the 5G LDPC codes with information lengths $K = 320$ and rate $1/5, 1/3, 2/5, 1/2$, respectively. Note that the Polyanskiy-Poor-Verdú (PPV) bounds [129] for different code rates are also shown in the figures for comparison. We observe that the AMP-EQML decoding scheme outperforms the EQML decoding for both $j_{\max} = 4$ and 6. For example, compared to the EQML decoding, it is approximately 0.05 dB gain for $R = 1/5$ by using the proposed AMP-EQML decoding scheme with $j_{\max} = 4$ at about FER $10^{-2} \sim 10^{-4}$, whereas an 0.03 dB improvement of FER can also be obtained for information length $K = 320$ with $j_{\max} = 6$. However, the performance gain reduces with the increasing of the code rate. This is due to the fact that for moderate to long information length K, with the maximum number of saturated VNs $j_{\max} = 4$ or 6, the portion of the saturated VNs increases compared to the length of the whole codeword. Thus, the performance of the EQML decoding method for the codes with high rate codes is closer to the ML performance compared to the codes with low code rates. Therefore, the margin of the FER performance for the EQML decoding compared to the PPV bound decreases, which means a lower potential improvement in FER performance by using the post-processing.

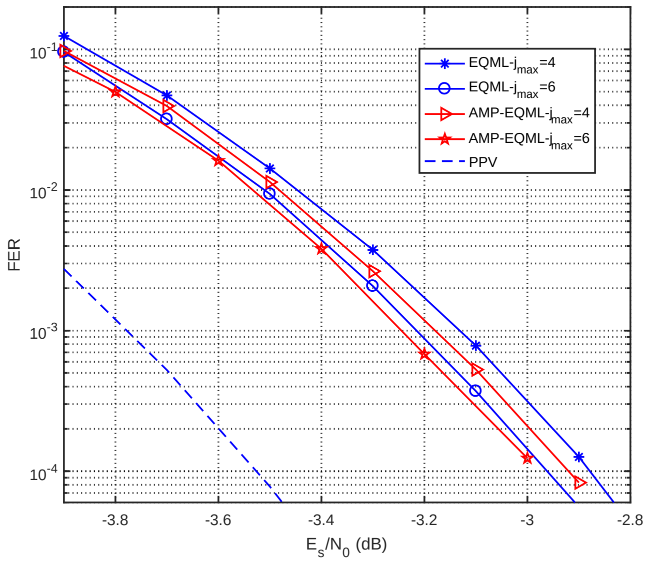


Figure 6.34: FER for BG2 LDPC code with $R = 1/5$, $K = 320$.

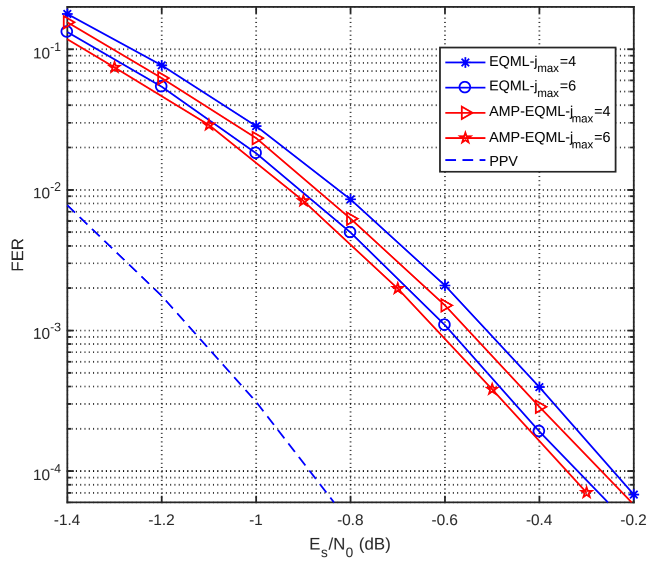


Figure 6.35: FER for BG2 LDPC code with $R = 1/3$, $K = 320$.

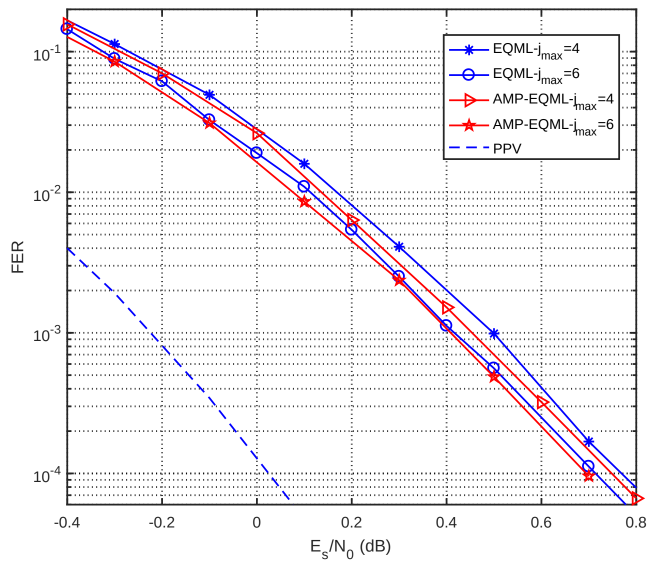


Figure 6.36: FER for BG2 LDPC code with $R = 2/5$, $K = 320$.

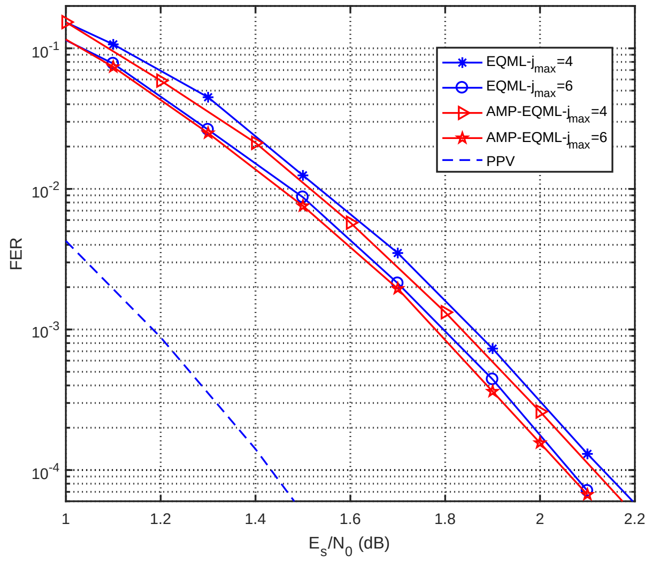


Figure 6.37: FER for BG2 LDPC code with $R = 1/2$, $K = 320$.

Simulation results for codes with $K = 752$

Figures 6.38 - to 6.41 demonstrate the FER performance of BG2 LDPC codes with information lengths $K = 752$ and rates $1/5, 1/3, 2/5, 1/2$, respectively. Similar to the case of $K = 320$, for $j_{\max} = 4$, there is about 0.05 dB improvement in FER performance for the proposed AMP-EQML decoding scheme with code rate $R = 1/5$, while the performance gain reduces to about 0.04 dB for code rate R is equal to $1/3, 2/5$, and $1/2$. For $j_{\max} = 6$, there is still approximate 0.03 dB gain in FER for the proposed AMP-EQML decoding scheme compared to that of the EQML decoding with code rate $R = 1/5, 1/3$. However, the performance gain decreases to 0.02 for code rate $R = 2/5$ and $1/2$. Compared to the FER performance with different code rates for $K = 320$, the proposed AMP-EQML decoding scheme has a larger gain for high code rates, e.g. $R = 2/5, 1/2$, as the improvement of the post-processing for long codeword length is larger than that of short ones with the insufficient number of saturated VNs. Further, for $R = 1/5$, the performance of AMP-EQML with $j_{\max} = 4$ can nearly achieve the performance of EQML decoding with $j_{\max} = 6$.

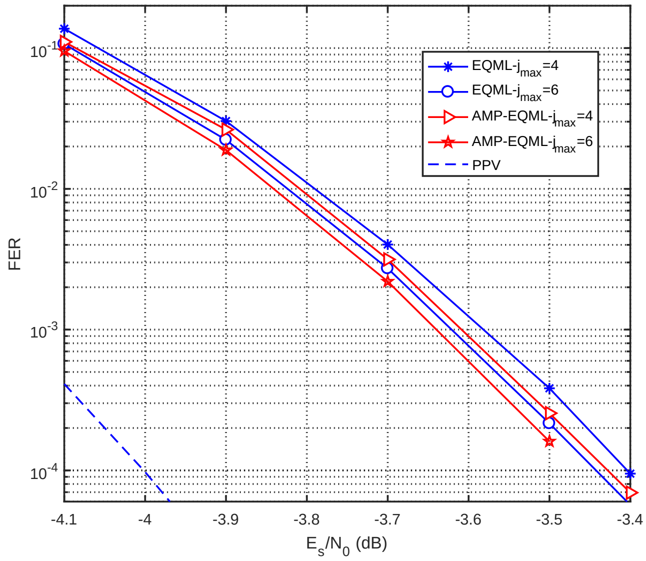


Figure 6.38: FER for BG2 LDPC code with $R = 1/5$, $K = 752$.

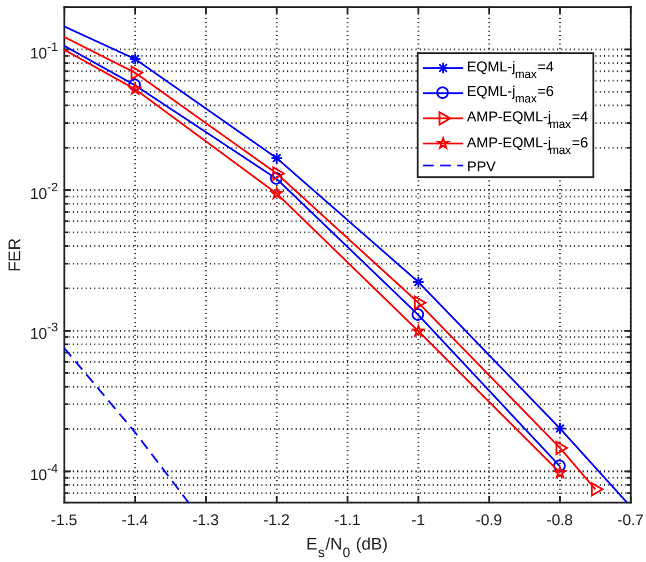


Figure 6.39: FER for BG2 LDPC code with $R = 1/3$, $K = 752$.

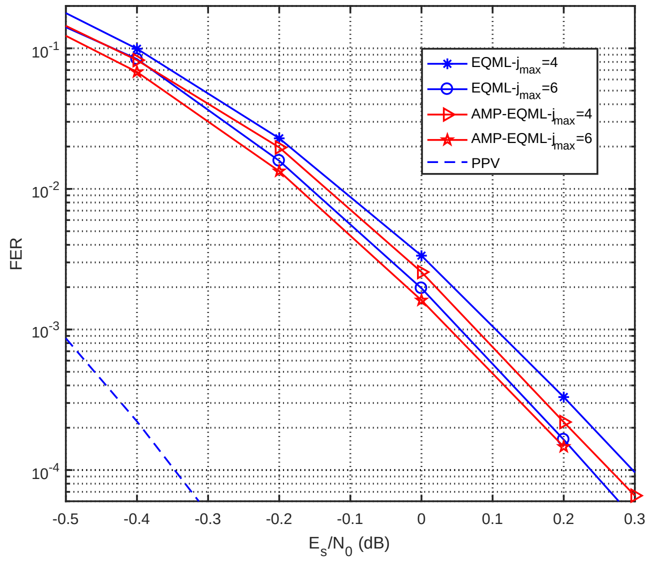


Figure 6.40: FER for BG2 LDPC code with $R = 2/5$, $K = 752$.

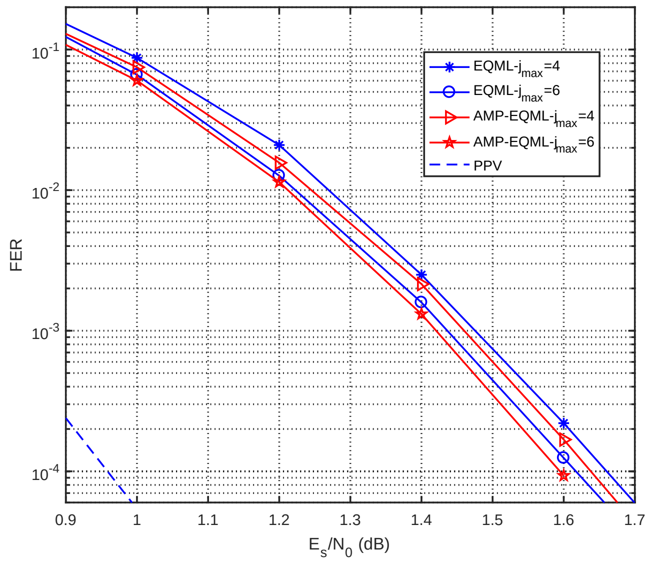


Figure 6.41: FER for BG2 LDPC code with $R = 1/2$, $K = 752$.

6.7 Summary

In this work, we proposed a novel AMP-aided decoding scheme on top of the conventional BP decoding. After the failure of the conventional BP decoding, the AMP-BP decoding scheme utilizes the output of the conventional BP decoding, and combines with the iterative AMP detector to estimate the positions of the unreliable bits. The sign of the associated original LLR values on these bit positions are flipped to generate a new decoder input sequence. Then the proposed decoding scheme performs only one round of decoding test with the modified input LLR sequence. Simulation results show that the proposed AMP-aided decoding scheme outperforms the counterpart of one-time BP decoding by approximately 0.1 dB for the 5G LDPC codes with various block lengths and low code rates. Simulation results show that the proposed AMP-BP decoding scheme outperforms the conventional BP decoding approximately 0.1 dB under different BP decoding scenarios for 5G raptor-like LDPC codes of various code rates and lengths. In addition, the proposed AMP-aided decoding scheme is then introduced to the EQML decoding method. We showed that with the aid from the AMP detector, the probability of an empty set of the EQML decoding method can be reduced by an approximately 5%. Simulation results show that the proposed AMP-EQML decoding scheme outperforms EQML decoding approximately $0.03 \sim 0.05$ dB for information bit length of $K = 320$ and 752.

Chapter 7

Conclusions and Future Prospects

In this chapter, we first conclude this thesis and then list some future research directions arising from our works.

7.1 Conclusions

In this thesis, the problems of constructing LDPC codes with high error correction capability and developing advanced decoding methods with good error performance have been studied. Specifically, we have proposed a construction method of SC LDPC codes and a reliability-based window decoding scheme to achieve high error correction capability and low error floor. In addition, we have also introduced novel decoding methods for 5G communication systems that require ultra-high reliability and low decoding complexity. The conclusion of this thesis is drawn as follows.

In Chapter 1, we have presented an overview of the 5G mobile networks and the motivations of this thesis. Afterwards, we have reviewed and discussed the related works in the literature about the construction and decoding methods of LDPC codes. We have provided the organization of this thesis. The main contributions of each conducted work have also been presented in this thesis.

In Chapter 2, we have introduced some basic concepts and knowledge of channel coding techniques in modern digital communication systems. It includes definitions, properties, decoding methods, and performance evaluation of channel coding.

In Chapter 3, we have provided detailed background knowledge of LDPC codes, which consists of definitions, different representations, decoding methods, and some examples of LDPC codes.

In Chapter 4, we have developed a new construction method of binary SC LDPC codes based on EG. In particular, we have proposed a two-dimensional edge-spreading process to generate a base matrix for the SC LDPC codes. The parity-check matrix of the constructed SC LDPC code is then obtained by unwrapping the base matrix and the lifting operation. We have evaluated the error performance of the constructed EG-SC LDPC codes by using a WBF decoding algorithm. It shows that the error performance of the constructed EG-SC LDPC codes is superior to that of their EG LDPC code counterparts, and there is no error floor compared to the constructed protograph SC LDPC codes and regular LDPC codes. We have further proposed an RBWD scheme for the SC LDPC codes based on a partial message reservation method and a partial syndrome check stopping rule. It is shown that the RBWD scheme significantly improves the error floor performance compared to the sliding window decoder with the WBF algorithm.

In Chapter 5, we have introduced an EQML decoding method for the 5G LDPC codes. We have proposed a two-dimensional scale-corrected min-sum algorithm based on partial self-correction and message amplification for the EQML decoding method. This results in the error performance near the SPA. We also have proposed a reprocessing architecture to further approach the error performance of the maximum likelihood decoding for 5G short LDPC codes. A novel node selection method based on the sign fluctuation of V2C messages has been proposed for the reprocessing. We have also presented a partial pruning stopping rule in the reprocessing to reduce the decoding complexity. A lower bound on the error performance has been derived by using the semi-analytical method to predict the error performance of the EQML decoding method. We have shown that the proposed EQML decoding method also outperforms the SPA with the same decoding complexity and approaches the PPV bound within 0.4 dB.

In Chapter 6, we have designed a detector aided decoding scheme based on the AMP algorithm for the 5G LDPC codes with a code rate of less than 1/2. We have proposed a decoding model for the AMP-aided decoding scheme, which aims to recover the error vector from the output of the decoder. The AMP algorithm has been used for selecting the unreliable VNs and flipping the sign of their associated channel output. The updated decoder input sequence is then generated, and one decoding test is conducted afterwards. We have shown that the proposed AMP-aided decoding scheme achieves a 0.1 dB gain over the counterpart of one-time decoding for the 5G LDPC codes with various block lengths and low code rates. Moreover, an AMP-EQML decoding scheme has also been presented for the decoding of the 5G LDPC codes, where the AMP algorithm is used in the reprocessing of the EQML decoding to further improve the error performance. We have analyzed some properties of the AMP-aided decoding scheme and shown that it can outperform the EQML decoding approximately $0.03 \sim 0.05$ dB for information bit length of $K = 320$ and 752.

7.2 Future Prospects

The explosive growth of traffic demand caused by the increasing number of smart devices gives rise to new challenges for future communication systems in terms of higher throughput, better quality-of-service, ultra-reliability. This thesis has addressed some of these challenges via constructing LDPC codes with high error correction capability and improving their error performance with advanced decoding methods. However, there are still some research issues that are worth further being studied and investigated. In the following, we propose some future research directions arising from the works presented in this thesis.

7.2.1 Construction of SC LDPC Codes with Large Girth

In Chapter 4, the proposed EG-SC LDPC codes are constructed by lifting and unwrapping the base matrix of their underlying EG LDPC codes with girth 6. The impact of spatially-coupling on the distribution of cycles for the constructed codes needs to be studied. Although there have been a few methods to construct LDPC block codes with large girth based on computer searching, how to develop a systematic way to optimize an SC LDPC code with a larger girth is still a very challenging problem. In addition, the convolutional gain and the improvement of error performance for SC LDPC codes with large girth under low-complexity decoding algorithms, such as WBF algorithm and BF algorithm, is also worth further investigation.

7.2.2 Designing Decoding Algorithms with Low Complexity for LDPC Codes

It is known that ultra-high-speed decoders have become a tendency to satisfy the increase of data rate and throughput demands in future communication systems.

This requires the decoders to have an ultra-low decoding complexity, where the hard-decision decoding algorithms are favored by applying to this case. Moreover, the presented decoding methods in this thesis mainly focus on improving the error performance by using the soft-decision decoding algorithms. As a result, there may be a large decoding latency caused by multiple times of decoding tests in the reprocessing. Therefore, it is worth developing the more simplified reprocessing architectures for hard-decision decoding algorithms with low complexity.

Bibliography

[1] 3GPP, “NR; Multiplexing and channel coding,” Release 15, Technical Specification (TS) 38.212, 2017.

[2] W. C. Y. Lee, *Mobile Cellular Telecommunications Systems.* New York, NY, USA: McGraw-Hill, Inc., 1990.

[3] E. Dahlman, *3G Evolution: HSPA and LTE for Mobile Broadband.* Elsevier Academic Press, 2007.

[4] S. Sesia, I. Toufik, and M. Baker, *LTE - The UMTS Long Term Evolution: From Theory to Practice.* Wiley, 2011.

[5] I. F. Akyildiz, D. M. Gutierrez-Estevez, R. Balakrishnan, and E. Chavarria-Reyes, “Full length article: Lte-advanced and the evolution to beyond 4g (b4g) systems,” *Phys. Commun.*, vol. 10, pp. 31–60, Mar. 2014.

[6] R. Chávez-Santiago, M. Szydełko, A. Kliks, F. Foukalas, Y. Haddad, K. E. Nolan, M. Y. Kelly, M. T. Masonta, and I. Balasingham, “5g: The convergence of wireless communications,” *Wireless Personal Communications*, vol. 83, no. 3, pp. 1617–1642, Aug. 2015.

[7] D. W. N. Marco Breiling, F. B. Christian Rohde, and R. Schober, *SUDAS: mmWave relaying for 5G outdoor-to-indoor communications.* Institution of Engineering and Technology, 2016.

[8] M. Vaezi, Z. Ding, and H. Poor, *Multiple Access Techniques for 5G Wireless Networks and Beyond.* Springer, 2018.

[9] 3GPP, “5G; Service requirements for next generation new services and markets,” 3rd Generation Partnership Project (3GPP), TS 22.261, Mar. 2017. [Online]. Available: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3107

[10] ITU-R, “ITU-R M.[IMT-2020.TECH PERF REQ] - minimum requirements related to technical performance for IMT-2020 radio interface(s),” ITU-R M.2410-0, TR, Nov. 2017. [Online]. Available: https://www.itu.int/dms_pub/itu-r/opb/rep/R-REP-M.2410-2017-PDF-E.pdf

[11] 3GPP, “Study on new radio (NR) access technology physical layer aspects,” 3rd Generation Partnership Project (3GPP), TR 38.802, Mar. 2017. [Online]. Available: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3066

[12] ——, "Study on physical layer enhancements for NR ultra-reliable and low latency case (URLLC)," 3rd Generation Partnership Project (3GPP), TR 38.824, Jul. 2018. [Online]. Available: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3498

[13] V. W. S. Wong, R. Schober, D. W. K. Ng, and L. Wang, *Key Technologies for 5G Wireless Systems.* Cambridge, UK: Cambridge Univ. Press, 2017.

[14] J. G. Andrews, S. Buzzi, W. Choi, S. V. Hanly, A. Lozano, A. C. K. Soong, and J. C. Zhang, "What will 5G be?" *IEEE J. Select. Areas Commun.*, vol. 32, no. 6, pp. 1065–1082, Jun. 2014.

[15] F. Boccardi, R. W. Heath, A. Lozano, T. L. Marzetta, and P. Popovski, "Five disruptive technology directions for 5G," *IEEE Commun. Mag.*, vol. 52, no. 2, pp. 74–80, Feb. 2014.

[16] Y. Saito, Y. Kishiyama, A. Benjebbour, T. Nakamura, A. Li, and K. Higuchi, "Non-orthogonal multiple access (NOMA) for cellular future radio access," in *Proc. IEEE Veh. Techn. Conf.*, Jun. 2013, pp. 1–5.

[17] Y. Liu, Z. Qin, M. Elkashlan, Z. Ding, A. Nallanathan, and L. Hanzo, "Nonorthogonal multiple access for 5G and beyond," *Proc. of IEEE*, vol. 105, no. 12, pp. 2347–2381, Dec. 2017.

[18] L. Dai, B. Wang, Y. Yuan, S. Han, C.-L. I, and Z. Wang, "Non-orthogonal multiple access for 5G: Solutions, challenges, opportunities, and future research trends," *IEEE Commun. Mag.*, vol. 53, no. 9, pp. 74–81, Sep. 2015.

[19] Z. Wei, D. W. K. Ng, J. Yuan, and H. Wang, "Optimal resource allocation for power-efficient mc-noma with imperfect channel state information," *IEEE Trans. Commun.*, vol. 65, no. 9, pp. 3944–3961, Sep. 2017.

[20] Z. Wei, D. W. K. Ng, and J. Yuan, "Joint pilot and payload power control for uplink MIMO-NOMA with MRC-SIC receivers," *IEEE Commun. Lett.*, vol. 22, no. 4, pp. 692–695, Apr. 2018.

[21] Z. Wei, J. Yuan, D. W. K. Ng, M. Elkashlan, and Z. Ding, "A survey of downlink non-orthogonal multiple access for 5G wireless communication networks," *ZTE Commun.*, vol. 14, no. 4, pp. 17–25, Oct. 2016.

[22] X. Chen, Z. Zhang, C. Zhong, and D. W. K. Ng, "Exploiting multiple-antenna techniques for non-orthogonal multiple access," *IEEE J. Select. Areas Commun.*, vol. 35, no. 10, pp. 2207–2220, Oct. 2017.

[23] X. Sun, N. Yang, S. Yan, Z. Ding, D. W. K. Ng, C. Shen, and Z. Zhong, "Joint beamforming and power allocation in downlink noma multiuser

mimo networks," *IEEE Trans. Wireless Commun.*, vol. 17, no. 8, pp. 5367–5381, Aug. 2018.

[24] X. Chen, R. Jia, and D. W. K. Ng, "The application of relay to massive non-orthogonal multiple access," *IEEE Trans. Commun.*, vol. 66, no. 11, pp. 5168–5180, Nov. 2018.

[25] Z. Ding, D. W. K. Ng, R. Schober, and H. V. Poor, "Delay minimization for noma-mec offloading," *IEEE Signal Process. Lett.*, vol. 25, no. 12, pp. 1875–1879, Dec. 2018.

[26] M. Qiu, Y. Huang, S. Shieh, and J. Yuan, "A lattice-partition framework of downlink non-orthogonal multiple access without sic," *IEEE Trans. Commun.*, vol. 66, no. 6, pp. 2532–2546, Jun. 2018.

[27] M. Qiu, Y. Huang, J. Yuan, and C. Wang, "Lattice-partition-based downlink non-orthogonal multiple access without sic for slow fading channels," *IEEE Trans. Commun.*, vol. 67, no. 2, pp. 1166–1181, Feb. 2019.

[28] M. Qiu, Y. Huang, and J. Yuan, "Downlink non-orthogonal multiple access without sic for block fading channels: An algebraic rotation approach," *IEEE Trans. Wireless Commun.*, vol. 18, no. 8, pp. 3903–3918, Aug. 2019.

[29] L. Xiang, D. W. K. Ng, X. Ge, Z. Ding, V. W. S. Wong, and R. Schober, "Cache-aided non-orthogonal multiple access: The two-user case," *IEEE J. Select. Topics Signal Process.*, vol. 13, no. 3, pp. 436–451, Jun. 2019.

[30] R. Jia, X. Chen, C. Zhong, D. W. K. Ng, H. Lin, and Z. Zhang, "Design of non-orthogonal beamspace multiple access for cellular internet-of-things," *IEEE J. Select. Topics Signal Process.*, vol. 13, no. 3, pp. 538–552, Jun. 2019.

[31] X. Chen, R. Jia, and D. W. K. Ng, "On the design of massive non-orthogonal multiple access with imperfect successive interference cancellation," *IEEE Trans. Commun.*, vol. 67, no. 3, pp. 2539–2551, Mar. 2019.

[32] S. He, Y. Wu, D. W. K. Ng, and Y. Huang, "Joint optimization of analog beam and user scheduling for millimeter wave communications," *IEEE Communications Letters*, vol. 21, no. 12, pp. 2638–2641, Dec. 2017.

[33] L. Zhao, D. W. K. Ng, and J. Yuan, "Multi-user precoding and channel estimation for hybrid millimeter wave systems," *IEEE J. Select. Areas Commun.*, vol. 35, no. 7, pp. 1576–1590, Jul. 2017.

[34] L. Zhao, G. Geraci, T. Yang, D. W. K. Ng, and J. Yuan, "A tone-based aoa estimation and multiuser precoding for millimeter wave massive mimo," *IEEE Trans. Commun.*, vol. 65, no. 12, pp. 5209–5225, Dec. 2017.

[35] L. Zhao, Z. Wei, D. W. K. Ng, J. Yuan, and M. C. Reed, "Multi-cell hybrid millimeter wave systems: Pilot contamination and interference mitigation," *IEEE Trans. Commun.*, vol. 66, no. 11, pp. 5740–5755, Nov. 2018.

[36] Z. Wei, L. Zhao, J. Guo, D. W. K. Ng, and J. Yuan, "Multi-beam noma for hybrid mmwave systems," *IEEE Trans. Commun.*, vol. 67, no. 2, pp. 1705–1719, Feb. 2019.

[37] Z. Wei, D. W. K. Ng, and J. Yuan, "Noma for hybrid mmwave communication systems with beamwidth control," *IEEE J. Select. Topics Signal Process.*, vol. 13, no. 3, pp. 567–583, Jun. 2019.

[38] H. Q. Ngo, E. G. Larsson, and T. L. Marzetta, "Energy and spectral efficiency of very large multiuser mimo systems," *IEEE Trans. Commun.*, vol. 61, no. 4, pp. 1436–1449, Apr. 2013.

[39] E. G. Larsson, O. Edfors, F. Tufvesson, and T. L. Marzetta, "Massive mimo for next generation wireless systems," *IEEE Commun. Mag.*, vol. 52, no. 2, pp. 186–195, Feb. 2014.

[40] J. Chen, X. Chen, W. H. Gerstacker, and D. W. K. Ng, "Resource allocation for a massive mimo relay aided secure communication," *IEEE Trans. on Inf. Forensics and Security*, vol. 11, no. 8, pp. 1700–1711, Aug. 2016.

[41] Y. Wu, R. Schober, D. W. K. Ng, C. Xiao, and G. Caire, "Secure massive mimo transmission with an active eavesdropper," *IEEE Trans. Inf. Theory*, vol. 62, no. 7, pp. 3880–3900, Jul. 2016.

[42] J. Zhu, D. W. K. Ng, N. Wang, R. Schober, and V. K. Bhargava, "Analysis and design of secure massive mimo systems in the presence of hardware impairments," *IEEE Trans. Wireless Commun.*, vol. 16, no. 3, pp. 2001–2016, Mar. 2017.

[43] X. Wei, W. Peng, D. Chen, D. W. K. Ng, and T. Jiang, "Joint channel parameter estimation in multi-cell massive mimo system," *IEEE Trans. Commun.*, vol. 67, no. 5, pp. 3251–3264, May. 2019.

[44] C. E. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J.*, vol. 27, no. 1, pp. 379–423, Oct. 1948.

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

[46] Y. Kou, S. Lin, and M. P. C. Fossorier, "Low-density parity-check codes based on finite geometries: a rediscovery and new results," *IEEE Trans. Inf. Theory*, vol. 47, no. 7, pp. 2711–2736, Nov. 2001.

[47] Jinghu Chen, A. Dholakia, E. Eleftheriou, M. P. C. Fossorier, and Xiao-Yu Hu, "Reduced-complexity decoding of LDPC codes," *IEEE Trans. Commun.*, vol. 53, no. 8, pp. 1288–1299, Aug. 2005.

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

[49] J. Zhang, M. Fossorier, D. Gu, and J. Zhang, "Two-dimensional correction for min-sum decoding of irregular LDPC codes," *IEEE Commun. Lett.*, vol. 10, no. 3, pp. 180–182, Mar. 2006.

[50] V. Savin, "Self-corrected min-sum decoding of LDPC codes," in *Proc. IEEE Int. Sympos. on Inf. Theory*, Jul. 2008, pp. 146–150.

[51] S. Lin and D. J. Costello, *Error Control Coding, Second Edition.* Prentice-Hall, Inc., 2004.

[52] T. Richardson and R. Urbanke, *Modern coding theory.* Cambridge University Press, 2008.

[53] H. Ma and J. Wolf, "On tail biting convolutional codes," *IEEE Trans. Commun.*, vol. 34, no. 2, pp. 104–111, Feb. 1986.

[54] T. Moon, *Error Correction Coding: Mathematical Methods and Algorithms.* Wiley, 2005.

[55] R. Hamming, "Error Detecting and Error Correcting Codes," *Bell System Techincal Journal*, vol. 29, pp. 147–160, 1950.

[56] D. E. Muller, "Application of boolean algebra to switching circuit design and to error detection," *Transactions of the I.R.E. Professional Group on Electronic Computers*, vol. EC-3, no. 3, pp. 6–12, Sep. 1954.

[57] E. Prange, *Cyclic Error-correcting Codes in Two Symbols.* Air Force Cambridge Research Center, 1957.

[58] R. Bose and D. Ray-Chaudhuri, "On a class of error correcting binary group codes," *Inf. and Control*, vol. 3, no. 1, pp. 68 – 79, 1960.

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

[60] I. Reed and G. Solomon, "Polynomial codes over certain finite fields," *Journal of the Society for Industrial and Applied Mathematics*, vol. 8, no. 2, pp. 300–304, 1960.

[61] ETSI, “Digital video broadcasting; second generation framing structure, channel coding and modulation systems for broadcasting, interactive services, news gathering,” 3rd Generation Partnership Project (3GPP), EN 302 307 V1.1.1, Jun. 2017. [Online]. Available: http://www.etsi.org

[62] J. Bourne and D. Burstein, *DSL: A Wiley Tech Brief.* Wiley, 2002.

[63] P. Elias, “Coding for noisy channels,” *IRE Convetion Record*, vol. 4, pp. 37–46, 1955.

[64] A. Viterbi, “Error bounds for convolutional codes and an asymptotically optimum decoding algorithm,” *IEEE Trans. Inf. Theory*, vol. 13, no. 2, pp. 260–269, Apr. 1967.

[65] P. Elias, “Error-free coding,” *IEEE Trans. Inf. Theory*, vol. 4, no. 4, pp. 29–37, Sep. 1954.

[66] G. Forney, *Concatenated Codes.* M.I.T. Press, 1966.

[67] C. Berrou, A. Glavieux, and P. Thitimajshima, “Near shannon limit error-correcting coding and decoding: Turbo-codes. 1,” in *Proc. IEEE Int. Commun. Conf.*, vol. 2, May. 1993, pp. 1064–1070 vol.2.

[68] B. Vucetic and J. Yuan, *Turbo Codes: Principles and Applications.* Springer US, 2012.

[69] S. Johnson, *Iterative Error Correction: Turbo, Low-Density Parity-Check and Repeat-Accumulate Codes.* Cambridge University Press, 2010.

[70] R. Gallager, “Low-density parity-check codes,” *IEEE Trans. Inf. Theory*, vol. 8, no. 1, pp. 21–28, Jan. 1962.

[71] D. J. C. MacKay and R. M. Neal, “Near shannon limit performance of low density parity check codes,” *Electronics Letters*, vol. 33, no. 6, pp. 457–458, Mar. 1997.

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

[73] Sae-Young Chung, G. D. Forney, T. J. Richardson, and R. Urbanke, “On the design of low-density parity-check codes within 0.0045 db of the shannon limit,” *IEEE Commun. Lett.*, vol. 5, no. 2, pp. 58–60, Feb. 2001.

[74] W. Ryan and S. Lin, *Channel codes: classical and modern.* Cambridge University Press, 2009.

[75] P. Cappelletti, C. Golla, P. Olivo and E. Zanoni, *Flash memories, 1st Edition.* Kluwer Academic Publishers, 1999.

[76] H. Weingarten, "New strategies to overcome 3bpc challenges," in *Flash Memory Summit*, Aug. 2010.

[77] P. Du, H. Lue, Y. Shih, K. Hsieh, and C. Lu, "Overview of 3D NAND flash and progress of split-page 3D vertical gate (3DVG) NAND architecture," in *Proc. IEEE Int. Conf. on Solid-State and Integrated Circuit Technology*, Oct. 2014, pp. 1–4.

[78] S. Cho, D. Kim, J. Choi, and J. Ha, "Block-wise concatenated BCH codes for NAND flash memories," *IEEE Trans. Commun.*, vol. 62, no. 4, pp. 1164–1177, Apr. 2014.

[79] G. Yu and J. Moon, "Concatenated raptor codes in NAND flash memory," *IEEE J. Select. Areas Commun.*, vol. 32, no. 5, pp. 857–869, May. 2014.

[80] C. Yang, Y. Emre, and C. Chakrabarti, "Product code schemes for error correction in mlc nand flash memories," *IEEE Trans. on VLSI Systems*, vol. 20, no. 12, pp. 2302–2314, Dec. 2012.

[81] M. Qiu, L. Yang, Y. Xie, and J. Yuan, "Terminated staircase codes for nand flash memories," *IEEE Trans. Commun.*, vol. 66, no. 12, pp. 5861–5875, Dec. 2018.

[82] K. Haymaker and C. A. Kelley, "Structured bit-interleaved LDPC codes for MLC flash memory," *IEEE J. Select. Areas Commun.*, vol. 32, no. 5, pp. 870–879, May. 2014.

[83] Y. Lin, H. Li, M. Chung, and A. Wu, "Byte-reconfigurable LDPC codec design with application to high-performance ECC of NAND flash memory systems," *IEEE Trans. Circuits Syst. I*, vol. 62, no. 7, pp. 1794–1804, Jul. 2015.

[84] G. Dong, N. Xie, and T. Zhang, "On the use of soft-decision error-correction codes in NAND flash memory," *IEEE Trans. Circuits Syst. I*, vol. 58, no. 2, pp. 429–439, Feb. 2011.

[85] D. J. Costello, L. Dolecek, T. E. Fuja, J. Kliewer, D. G. M. Mitchell, and R. Smarandache, "Spatially coupled sparse codes on graphs: theory and practice," *IEEE Commun. Mag.*, vol. 52, no. 7, pp. 168–176, Jul. 2014.

[86] A. J. Felstrom and K. S. Zigangirov, "Time-varying periodic convolutional codes with low-density parity-check matrix," *IEEE Trans. Inf. Theory*, vol. 45, no. 6, pp. 2181–2191, Sep. 1999.

[87] W. Nitzold, M. Lentmaier, and G. P. Fettweis, "Spatially coupled protograph-based LDPC codes for incremental redundancy," in *Proc. Int. Sympos. Turbo Codes Iterative Inf. Process*, Aug. 2012, pp. 155–159.

[88] K. Huang, D. G. M. Mitchell, L. Wei, X. Ma, and D. J. Costello, "Performance comparison of LDPC block and spatially coupled codes over GF(q)," *IEEE Trans. Commun.*, vol. 63, no. 3, pp. 592–604, Mar. 2015.

[89] D. G. M. Mitchell, M. Lentmaier, and D. J. Costello, "Spatially coupled LDPC codes constructed from protographs," *IEEE Trans. Inf. Theory*, vol. 61, no. 9, pp. 4866–4889, Sep. 2015.

[90] L. Wei, D. G. M. Mitchell, T. E. Fuja, and D. J. Costello, "Design of spatially coupled LDPC codes over GF(q) for windowed decoding," *IEEE Trans. Inf. Theory*, vol. 62, no. 9, pp. 4781–4800, Sep. 2016.

[91] Y. Xie, L. Yang, P. Kang, and J. Yuan, "Euclidean geometry-based spatially coupled LDPC codes for storage," *IEEE J. Select. Areas Commun.*, vol. 34, no. 9, pp. 2498–2509, Sep. 2016.

[92] J. Zhang, B. Bai, D. Deng, M. Zhu, H. Xu, and M. Guan, "Non-uniform spatially-coupled LDPC codes over GF(2^m)," in *Proc. IEEE Int. Sympos. on Inf. Theory*, Jun. 2018, pp. 816–820.

[93] A. R. Iyengar, M. Papaleo, P. H. Siegel, J. K. Wolf, A. Vanelli-Coralli, and G. E. Corazza, "Windowed decoding of protograph-based LDPC convolutional codes over erasure channels," *IEEE Trans. Inf. Theory*, vol. 58, no. 4, pp. 2303–2320, Apr. 2012.

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

[95] I. Ali, J. H. Kim, S. H. Kim, H. Kwak, and J. S. No, "Improving windowed decoding of SC LDPC codes by effective decoding termination, message reuse, and amplification," *IEEE Access*, Feb. 2017.

[96] N. U. Hassan, A. E. Pusane, M. Lentmaier, G. P. Fettweis, and D. J. Costello, "Non-uniform window decoding schedules for spatially coupled LDPC codes," *IEEE Trans. Commun.*, vol. 65, no. 2, pp. 501–510, Feb. 2017.

[97] M. Lentmaier, M. M. Prenda, and G. P. Fettweis, "Efficient message passing scheduling for terminated LDPC convolutional codes," in *Proc. IEEE Int. Sympos. on Inf. Theory*, Jul. 2011, pp. 1826–1830.

[98] T. Mohsenin, D. N. Truong, and B. M. Baas, "A low-complexity message-passing algorithm for reduced routing congestion in LDPC decoders," *IEEE Trans. Circuits Syst. I*, vol. 57, no. 5, pp. 1048–1061, May 2010.

[99] G. Dong, N. Xie, and T. Zhang, "On the use of soft-decision error-correction codes in nand flash memory," *IEEE Trans. Circuits Syst. I*, vol. 58, no. 2, pp. 429–439, Feb 2011.

[100] N. Q. Nhan, T. M. N. Ngatched, O. A. Dobre, P. Rostaing, K. Amis, and E. Radoi, "Multiple-votes parallel symbol-flipping decoding algorithm for non-binary LDPC codes," *IEEE Commun. Lett.*, vol. 19, no. 6, pp. 905–908, Jun. 2015.

[101] Z. Liu and D. A. Pados, "A decoding algorithm for finite-geometry LDPC codes," *IEEE Trans. Commun.*, vol. 53, no. 3, pp. 415–421, Mar. 2005.

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

[103] J. Chen, R. M. Tanner, C. Jones, and Y. Li, "Improved min-sum decoding algorithms for irregular LDPC codes," in *Proc. IEEE Int. Sympos. on Inf. Theory*, Sep. 2005, pp. 449–453.

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

[105] S. V. S. Ranganathan, R. D. Wesel, and D. Divsalar, "Linear rate-compatible codes with degree-1 extending variable nodes under iterative decoding," in *Proc. IEEE Int. Sympos. on Inf. Theory*, Jun. 2018, pp. 1166–1170.

[106] T. Etzion, A. Trachtenberg, and A. Vardy, "Which codes have cycle-free tanner graphs," *IEEE Trans. Inf. Theory*, vol. 45, no. 6, pp. 2173–2181, Sep. 1999.

[107] M. P. C. Fossorier, "Iterative reliability-based decoding of low-density parity check codes," *IEEE J. Select. Areas Commun.*, vol. 19, no. 5, pp. 908–917, May. 2001.

[108] N. Varnica, M. P. C. Fossorier, and A. Kavcic, "Augmented belief propagation decoding of low-density parity check codes," *IEEE Trans. Commun.*, vol. 55, no. 7, pp. 1308–1317, Jul. 2007.

[109] S. Scholl, P. Schläfer, and N. Wehn, "Saturated min-sum decoding: An afterburner for LDPC decoder hardware," in *2016 Design, Automation Test in Europe Conference Exhibition (DATE)*, Mar. 2016, pp. 1219–1224.

[110] M. Ostojic and H. Loeliger, "Multitree decoding and multitree-aided LDPC decoding," in *Proc. IEEE Int. Sympos. on Inf. Theory*, Jun. 2010, pp. 779–783.

[111] Xiao-Yu Hu, E. Eleftheriou, and D. M. Arnold, "Regular and irregular progressive edge-growth tanner graphs," *IEEE Trans. Inf. Theory*, vol. 51, no. 1, pp. 386–398, Jan. 2005.

[112] Tao Tian, C. R. Jones, J. D. Villasenor, and R. D. Wesel, "Selective avoidance of cycles in irregular ldpc code construction," *IEEE Trans. Commun.*, vol. 52, no. 8, pp. 1242–1247, Aug. 2004.

[113] J. Thorpe, "Low-density parity-check (ldpc) codes constructed from protographs," *IPN Progress Report*, vol. 42, no. 154, pp. 42–154, Aug. 2003.

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

[115] D. Divsalar, S. Dolinar, C. R. Jones, and K. Andrews, "Capacity-approaching protograph codes," *IEEE J. Select. Areas Commun.*, vol. 27, no. 6, pp. 876–888, Aug. 2009.

[116] S. Kudekar, T. Richardson, and R. L. Urbanke, "Spatially coupled ensembles universally achieve capacity under belief propagation," *IEEE Trans. Inf. Theory*, vol. 59, no. 12, pp. 7761–7813, Dec. 2013.

[117] M. Stinner and P. M. Olmos, "On the waterfall performance of finite-length SC-LDPC codes constructed from protographs," *IEEE J. Select. Areas Commun.*, vol. 34, no. 2, pp. 345–361, Feb. 2016.

[118] S. Kudekar, T. J. Richardson, and R. L. Urbanke, "Threshold saturation via spatial coupling: Why convolutional LDPC ensembles perform so well over the BEC," *IEEE Trans. Inf. Theory*, vol. 57, no. 2, pp. 803–834, Feb. 2011.

[119] A. Yedla, Y. Y. Jian, P. S. Nguyen, and H. D. Pfister, "A simple proof of threshold saturation for coupled scalar recursions," in *Proc. Int. Sympos. Turbo Codes Iterative Inf. Process*, Aug. 2012, pp. 51–55.

[120] S. Kudekar, C. Méassony, T. Richardsony, and R. Urbankez, "Threshold saturation on BMS channels via spatial coupling," in *Proc. Int. Sympos. Turbo Codes Iterative Inf. Process*, Sep. 2010, pp. 309–313.

[121] D. Donoho, "Message-passing algorithms for compressed sensing," *Proc. of the National Academy of Sciences*, vol. 106, no. 45, pp. 18 914–18 919, Nov. 2009.

[122] R. Blahut, *Algebraic Codes for Data Transmission.* Cambridge University Press, 2003.

[123] J. Proakis, *Digital Communications.* McGraw-Hill, 2001.

[124] R. Gallager, *Principles of Digital Communication.* Cambridge University Press, 2008.

[125] D. Tse and P. Viswanath, *Fundamentals of Wireless Communication.* Cambridge University Press, 2005.

[126] A. Goldsmith and K. (Firm), *Wireless Communications.* Cambridge University Press, 2005.

[127] R. Gallager, "The random coding bound is tight for the average code (corresp.)," *IEEE Trans. Inf. Theory*, vol. 19, no. 2, pp. 244–246, Mar. 1973.

[128] C. Shannon, R. Gallager, and E. Berlekamp, "Lower bounds to error probability for coding on discrete memoryless channels. i," *Inf. and Control*, vol. 10, no. 1, pp. 65 – 103, Jan. 1967.

[129] Y. Polyanskiy, H. V. Poor, and S. Verdu, "Channel coding rate in the finite blocklength regime," *IEEE Trans. Inf. Theory*, vol. 56, no. 5, pp. 2307–2359, May. 2010.

[130] T. Erseghe, "On the evaluation of the polyanskiy-poor-verd converse bound for finite block-length coding in awgn," *IEEE Trans. Inf. Theory*, vol. 61, no. 12, pp. 6578–6590, Dec. 2015.

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

[132] Y. Kou, S. Lin, and M. P. C. Fossorier, "Low density parity check codes: construction based on finite geometries," in *Proc. IEEE Global Commun. Conf.*, vol. 2, Nov. 2000, pp. 825–829 vol.2.

[133] T. Chen, K. Vakilinia, D. Divsalar, and R. D. Wesel, "Protograph-based raptor-like LDPC codes," *IEEE Trans. Commun.*, vol. 63, no. 5, pp. 1522–1532, May. 2015.

[134] 3GPP, "Final minutes report RAN1 AH1 NR," 3rd Generation Partnership Project (3GPP), TSG RAN WG1 Meeting, Jan. 2017. [Online]. Available: https://www.3gpp.org/ftp/tsg_ran/WG1_RL1/TSGR1_88/Docs/

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

[136] E. Yeo, P. Pakzad, B. Nikolic, and V. Anantharam, "High throughput low-density parity-check decoder architectures," in *Proc. IEEE Global Commun. Conf.*, vol. 5, Dec. 2001, pp. 3019–3024.

[137] M. M. Mansour and N. R. Shanbhag, "High-throughput ldpc decoders," *IEEE Trans. on VLSI Systems*, vol. 11, no. 6, pp. 976–996, Dec. 2003.

[138] J. Zhang and M. P. C. Fossorier, "Shuffled iterative decoding," *IEEE Trans. Commun.*, vol. 53, no. 2, pp. 209–213, Feb. 2005.

[139] E. Sharon, S. Litsyn, and J. Goldberger, "Efficient serial message-passing schedules for LDPC decoding," *IEEE Trans. Inf. Theory*, vol. 53, no. 11, pp. 4076–4091, Nov. 2007.

[140] A. I. V. Casado, M. Griot, and R. D. Wesel, "LDPC decoders with informed dynamic scheduling," *IEEE Trans. Commun.*, vol. 58, no. 12, pp. 3470–3479, Dec. 2010.

[141] G. Elidan, I. McGraw, and D. Koller, "Residual belief propagation: Informed scheduling for asynchronous message passing," in *Proc. of the Twenty-Second Conf. on Uncertainty in Artificial Intelligence*, vol. 9, Jul. 2006, pp. 165–173.

[142] Y. Gong, X. Liu, W. Ye, and G. Han, "Effective informed dynamic scheduling for belief propagation decoding of ldpc codes," *IEEE Trans. Commun.*, vol. 59, no. 10, pp. 2683–2691, Oct. 2011.

[143] H. C. Lee, Y. L. Ueng, S. M. Yeh, and W. Y. Weng, "Two informed dynamic scheduling strategies for iterative ldpc decoders," *IEEE Trans. Commun.*, vol. 61, no. 3, pp. 886–896, Mar. 2013.

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

[145] G. Liva and M. Chiani, "Protograph LDPC Codes Design Based on EXIT Analysis," in *Proc. IEEE Global Commun. Conf.*, Nov. 2007, pp. 3250–3254.

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

[147] Sae-Young Chung, R. Urbanke, and T. J. Richardson, "Gaussian approximation for sum-product decoding of low-density parity-check codes," in *Proc. IEEE Int. Sympos. on Inf. Theory*, June 2000, p. 318.

[148] F. Lehmann and G. M. Maggio, "Analysis of the iterative decoding of ldpc and product codes using the gaussian approximation," *IEEE Trans. Inf. Theory*, vol. 49, no. 11, pp. 2993–3000, Nov. 2003.

[149] E. Sharon, A. Ashikhmin, and S. Litsyn, "Analysis of low-density parity-check codes based on exit functions," *IEEE Trans. Commun.*, vol. 54, no. 8, pp. 1407–1414, Aug. 2006.

[150] Lei Chen, Jun Xu, I. Djurdjevic, and S. Lin, "Near-shannon-limit quasi-cyclic low-density parity-check codes," *IEEE Trans. Commun.*, vol. 52, no. 7, pp. 1038–1042, Jul. 2004.

[151] B. Vasic and O. Milenkovic, "Combinatorial constructions of low-density parity-check codes for iterative decoding," *IEEE Trans. Inf. Theory*, vol. 50, no. 6, pp. 1156–1176, Jun. 2004.

[152] Heng Tang, Jun Xu, S. Lin, and K. A. S. Abdel-Ghaffar, "Codes on finite geometries," *IEEE Trans. Inf. Theory*, vol. 51, no. 2, pp. 572–596, Feb. 2005.

[153] M. Shan, C. M. Zhao, and M. Jiang, "Improved weighted bit-flipping algorithm for decoding LDPC codes," *IEE Proc. - Commun.*, vol. 152, no. 6, pp. 919–922, Dec. 2005.

[154] L. Zhang, Q. Huang, and S. Lin, "Iterative algorithms for decoding a class of two-step majority-logic decodable cyclic codes," *IEEE Trans. Commun.*, vol. 59, no. 2, pp. 416–427, Feb. 2011.

[155] K. Liu, S. Lin, and K. Abdel-Ghaffar, "A revolving iterative algorithm for decoding algebraic cyclic and quasi-cyclic LDPC codes," *IEEE Trans. Commun.*, vol. 61, no. 12, pp. 4816–4827, Dec. 2013.

[156] M. Zhu, D. G. M. Mitchell, M. Lentmaier, D. J. Costello, and B. Bai, "Braided convolutional codes with sliding window decoding," *IEEE Trans. Commun.*, vol. 65, no. 9, pp. 3645–3658, Sep. 2017.

[157] Q. Huang, Q. Diao, S. Lin, and K. Abdel-Ghaffar, "Cyclic and quasi-cyclic ldpc codes on constrained parity-check matrices and their trapping sets," *IEEE Trans. Inf. Theory*, vol. 58, no. 5, pp. 2648–2671, May. 2012.

[158] Z. Li, L. Chen, L. Zeng, S. Lin, and W. Fong, "Efficient encoding of quasi-cyclic low-density parity-check codes," *IEEE Trans. Commun.*, vol. 53, no. 11, pp. 1973–1973, Nov. 2005.

[159] N. Hamada, "The rank of the incidence matrix of points and d-flats in finite geometries," *J. Sci. Hiroshima Univ. Ser. A-I Math.*, vol. 32, no. 2, pp. 381–396, 1968.

[160] Shu Lin, "Number of information symbols in polynomial codes," *IEEE Trans. Inf. Theory*, vol. 18, no. 6, pp. 785–794, Nov. 1972.

[161] Y. Xie, J. C. Mu, and J. Yuan, "Design of rate-compatible protograph-based LDPC codes with mixed circulants," in *Proc. Int. Sympos. Turbo Codes Iterative Inf. Process*, Sep. 2010, pp. 434–438.

[162] M. Zhang, Z. Wang, Q. Huang, and S. Wang, "Time-invariant quasi-cyclic spatially coupled LDPC codes based on packings," *IEEE Trans. Commun.*, vol. 64, no. 12, pp. 4936–4945, Dec. 2016.

[163] L. Yang, Y. Xie, J. Yuan, X. Cheng, and L. Wan, "Chained LDPC codes for future communication systems," *IEEE Commun. Lett.*, vol. 22, no. 5, pp. 898–901, May. 2018.

[164] H. Li, B. Bai, X. Mu, J. Zhang, and H. Xu, "Algebra-assisted construction of quasi-cyclic LDPC codes for 5G New Radio," *IEEE Access*, vol. 6, pp. 50 229–50 244, 2018.

[165] M. Shirvanimoghaddam, M. S. Mohammadi, R. Abbas, A. Minja, C. Yue, B. Matuz, G. Han, Z. Lin, W. Liu, Y. Li, S. Johnson, and B. Vucetic, "Short block-length codes for ultra-reliable low latency communications," *IEEE Commun. Mag.*, vol. 57, no. 2, pp. 130–137, Feb. 2019.

[166] X. Wu, L. Yang, J. Yuan, X. Cheng, and L. Wan, "Rate-compatible tail-biting convolutional codes for M2M communications," *IEEE Commun. Lett.*, vol. 22, no. 3, pp. 482–485, Mar. 2018.

[167] S. Zhao, X. Ma, Q. Huang, and B. Bai, "Recursive block markov superposition transmission of short codes: construction, analysis, and applications," *IEEE Trans. Commun.*, vol. 66, no. 7, pp. 2784–2796, Jul. 2018.

[168] T. Erseghe, "Coding in the finite-blocklength regime: Bounds based on laplace integrals and their asymptotic approximations," *IEEE Trans. Inf. Theory*, vol. 62, no. 12, pp. 6854–6883, Dec. 2016.

[169] P. Kang, Y. Xie, L. Yang, and J. Yuan, "Reliability-based windowed decoding for spatially coupled ldpc codes," *IEEE Commun. Lett.*, vol. 22, no. 7, pp. 1322–1325, Jul. 2018.

[170] G. Miller and D. Burshtein, "Bounds on the maximum-likelihood decoding error probability of low-density parity-check codes," *IEEE Trans. Inf. Theory*, vol. 47, no. 7, pp. 2696–2710, Nov. 2001.

[171] A. Barg and G. D. Forney, "Random codes: minimum distances and error exponents," *IEEE Trans. Inf. Theory*, vol. 48, no. 9, pp. 2568–2573, Sep. 2002.

[172] A. Cohen and N. Merhav, "Lower bounds on the error probability of block codes based on improvements on de Caen's inequality," *IEEE Trans. Inf. Theory*, vol. 50, no. 2, pp. 290–310, Feb. 2004.

[173] X. Ma, K. Huang, and B. Bai, "Systematic block markov superposition transmission of repetition codes," *IEEE Trans. Inf. Theory*, vol. 64, no. 3, pp. 1604–1620, Mar. 2018.

[174] X. Hu, M. P. C. Fossorier, and E. Eleftheriou, "On the computation of the minimum distance of low-density parity-check codes," in *Proc. IEEE Int. Commun. Conf.*, vol. 2, Jun. 2004, pp. 767–771 Vol.2.

[175] D. L. Donoho, A. Maleki, and A. Montanari, "Message passing algorithms for compressed sensing: I. motivation and construction," in *Proc. IEEE Inf. Theory Workshop*, Jan. 2010, pp. 1–5.

[176] D. L. Donoho, and A. Maleki, and A. Montanari, "Message passing algorithms for compressed sensing: II. analysis and validation," in *Proc. IEEE Inf. Theory Workshop*, Jan. 2010, pp. 1–5.

[177] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of The Royal Statistical Society, Series B*, vol. 58, pp. 267–288, 1994.

[178] E. Candès, "The restricted isometry property and its implications for compressed sensing," *Comptes Rendus Mathematique*, vol. 346, no. 9-10, pp. 589–592, May. 2008.

[179] A. Montanari, *Graphical models concepts in compressed sensing.* Cambridge University Press, 2012, pp. 394–438.

[180] Z. Chen, F. Sohrabi, and W. Yu, "Sparse activity detection for massive connectivity," *IEEE Trans. Signal Process.*, vol. 66, no. 7, pp. 1890–1904, Apr. 2018.

[181] Z. Sun, Z. Wei, L. Yang, J. Yuan, X. Cheng, and L. Wan, "Joint user identification and channel estimation in massive connectivity with transmission control," in *Proc. Int. Sympos. Turbo Codes Iterative Inf. Process*, Dec. 2018, pp. 1–5.