DESIGN OF JOINT SOURCE-CHANNEL CODING SYSTEMS BASED ON DOUBLE P-LDPC CODES

ZHAN JIA

PhD

The Hong Kong Polytechnic University

2024

The Hong Kong Polytechnic University

Department of Electrical and Electronic Engineering

Design of Joint Source-Channel Coding Systems Based on Double P-LDPC Codes

Zhan Jia

A thesis submitted in partial fulfilment of the requirements for the degree of Doctor of Philosophy

June 2024

CERTIFICATE OF ORIGINALITY

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it reproduces no material previously published or written, nor material that has been accepted for the award of any other degree or diploma, except where due acknowledgement has been made in the text.

_____ (Signed)

_____Zhan Jia_____(Name of student)

Abstract

In a traditional joint source-channel coding system based on double protograph-based LDPC (DP-LDPC) codes, two P-LDPC codes are employed as the source code and the channel code, respectively. They are connected by a source-check-channel-variable (SCCV) linking matrix, which consists of an identity matrix and a zero matrix. This linking matrix connects check nodes (CNs) in the source P-LDPC code and variable nodes (VNs) in the channel P-LDPC code. A joint source-channel decoder that facilitates message exchange between the source decoder and the channel decoder is used to decode the source. Based on this traditional structure, this thesis proposes several novel joint source-channel coding systems, aiming at improving error performance further.

Firstly, a novel class of DP-LDPC codes is proposed for the joint source-channel coding, where the identity matrix in the SCCV linking matrix of traditional DP-LDPC codes is replaced with a lower or upper triangular matrix with "1"s on its diagonal. By doing this, the flexibility of code design is increased and the linear source compression is preserved. Theoretical and simulation results have shown the superior performance of the proposed DP-LDPC codes compared to the traditional ones.

Secondly, we propose a novel joint source-channel coding scheme based on spatially coupled DP-LDPC (SC-DP-LDPC) codes. It has been proved that a concatenated spatially coupled protograph-based LDPC (SC-P-LDPC) code can have better error performance than a traditional DP-LDPC block code. Motivated by this point, SC-DP-LDPC codes are proposed, where the source spatially coupled P-LDPC (SC-P-LDPC) code and the channel SC-P-LDPC code are linked by spatially coupled SCCV (SC-SCCV) connections. By doing this, source SC-P-LDPC

and channel SC-P-LDPC are not in a simple concatenated relationship. Like decoding the concatenated SC-P-LDPC code, we use a sliding window joint belief propagation (BP) decoding algorithm to decode the SC-DP-LDPC code. Theoretical and simulation results show that the proposed codes are superior to the concatenated SC-LDPC codes and state-of-the-art DP-LDPC block codes.

Thirdly, we propose two new types of joint source-channel coding systems for both low-entropy and high-entropy sources. Studies done by others have shown that for the traditional DP-LDPC codes, the source threshold can be improved by adding connections between VNs in source P-LDPC code and CNs in channel P-LDPC code, which is represented by a source-variable-channel-check (SVCC) linking (base) matrix. According to this finding and our first proposal, we propose a novel joint source-channel block code (JSC-BC) based on double P-LDPC block codes. Two P-LDPC block codes are connected not only by a source-variable-channel-check (SVCC) linking (base) matrix but also by a source-check-channel-variable (SCCV) linking (base) matrix, which consists of a zero matrix and a lower or upper triangular (base) matrix with "1"s on its diagonal. Also, we modify the traditional joint protograph-based extrinsic information transfer (JP-EXIT) algorithm to calculate the source threshold of a JSC-BC. The JP-EXIT algorithm uses the whole joint protomatrix to calculate the source threshold of a code. The new technique is called the untransmitted protograph-based EXIT (UP-EXIT) algorithm. Compared to the JP-EXIT algorithm, the proposed UP-EXIT algorithm is more efficient because a smaller protograph consisting of only the untransmitted VNs (i.e., the source VNs and the punctured channel VNs) and their connected check nodes need to be considered. We first search for the candidate codes with the proposed code structure and high source thresholds by using the UP-EXIT algorithm. Then we select those also with low channel thresholds among the candidate codes by using the JP-EXIT algorithm. Moreover, we construct new JSC-BCs whose decoding complexities are controlled by limiting their maximum row weights. Theoretical and simulation results show that the codes newly constructed outperform state-of-the-art DP-LDPC block codes. Next, we spatially couple the joint source-channel block code and obtain a spatially coupled joint source-channel code (SC-JSCC) for further error performance im-

provement. Theoretical analyses and simulation results show that even with a smaller window size and lower decoding complexity, the SC-JSCC with the spatially coupled structure for each sub-block (source protomatrix, channel protomatrix, SCCV linking matrix, and SVCC linking matrix) can have better error performance than existing spatially-coupled DP-LDPC codes.

Publications

Journal papers:

- J. Zhan and F. C. M. Lau, "Joint design of source-channel codes with linear source encoding complexity and good channel thresholds based on double-protograph LDPC codes," *IEEE Commun. Lett.*, vol. 27, no. 11, pp. 2909–2913, Sep. 2023.

- J. Zhan and F. C. M. Lau, "Design of joint source-channel coding scheme based on spatially-coupled DP-LDPC codes," *IEEE Commun. Lett.*, vol. 28, no. 4, pp. 749–753, Apr. 2024, doi:10.1109/LCOMM.2024.3368223.

- J. Zhan, W. M. Tam, and F. C. M. Lau, "Novel Double Protograph LDPC Codes for Joint Source-Channel Coding Systems," submitted.

Conference papers:

- L. Ma, C. W. Sham, J. Zhan, and F. C. M. Lau, "Implementation for JSCC Scheme Based on QC-LDPC Codes," *2022 IEEE 11th Global Conference on Consumer Electronics (GCCE)*, Osaka, Japan, Oct. 2022.

Acknowledgements

I would like to express my deepest gratitude and appreciation to all those who have contributed to the completion of this thesis.

First and foremost, I am immensely grateful to my supervisor, Prof. Francis C. M. Lau, for his unwavering guidance, support, and invaluable insights throughout the research process. His expertise, patience, and dedication have been instrumental in shaping the direction of this work and helping me overcome various challenges. I am truly fortunate to have had such a knowledgeable and inspiring supervisor.

I would like to acknowledge the support and assistance received from Dr. Wai-Man Tam and Dr. Sheng Jiang. Their collaboration, discussions, and shared experiences have been instrumental in shaping my ideas and enhancing the quality of this work. I am grateful for their camaraderie and intellectual exchange.

Lastly, I extend my sincere appreciation to my family for their unwavering support, encouragement, and understanding throughout this journey. Their belief in me, constant motivation, and words of encouragement have been a source of strength during challenging times.

Acronyms

AWGN Additive White Gaussian Noise

BP Belief Propagation

CN Check Node

CPM Circulant Permutation Matrix

DP-LDPC Double Protograph-based Low-Density Parity-Check

DE Differential Evolution

JP-EXIT Joint Protograph-based EXtrinsic Information Transfer

JSC-BC Joint Source-Channel Block Code

LDPC Low-Density Parity-Check

LLR Log-Likelihood-Ratio

P-LDPC Protograph-based Low-Density Parity-Check

P-EXIT Protograph-based EXtrinsic Information Transfer

SC-LDPC Spatially Coupled Low-Density Parity-Check

SC-P-LDPC Spatially Coupled Protograph-based Low-Density Parity-Check

SC-DP-LDPC Spatially Coupled Double Protograph-based Low-Density Parity-Check

SCCV Source-Check-Channel-Variable

SC-SCCV Spatially-Coupled Source-Check-Channel-Variable

SVCC Source-Variable-Channel-Check

SC-SVCC Spatially-Coupled Source-Variable-Channel-Check

SC-JSCC Spatially-Coupled Joint Source-Channel Code

SSER Source Symbol Error Rate

UP-EXIT Untransmitted Protograph-based EXtrinsic Information Transfer

VN Variable Node

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Significance and background

The rapid development of wireless communication technology has made communication services an indispensable part of people's daily lives. While enjoying the convenience brought by the communication service, people have higher and higher expectations of the communication service quality. Low latency, high-efficiency, and high-reliability communication systems have become the basic requirements. The architecture of a traditional digital communication system is shown in Fig. 1.1, whose design is based on Shannon's source-channel separation theorem [1]. The theorem tells that source coding and channel coding can be optimized separately to obtain optimal performance in the case of infinite code length and known channel statistics. Source coding is employed to reduce the redundancy in the original information by compression while channel coding is used to protect the compressed data during transmission by adding redundant information (additional parity-check bits). Source coding has been developed from the earliest Shannon-Fano-Elias codes [1] to Huffman code [2], which is still widely used in all aspects of data compression and transmission [3, 4], to Golomb-Rice coding [5] used for Gaussian distribution source compression, and to arithmetic coding [6], which has also aroused the interest of many researchers. In terms of the research on channel coding, it includes the earliest Hamming codes [7], Reed-Muller codes (referred to as RM codes) [8], convolutional codes [9], BCH

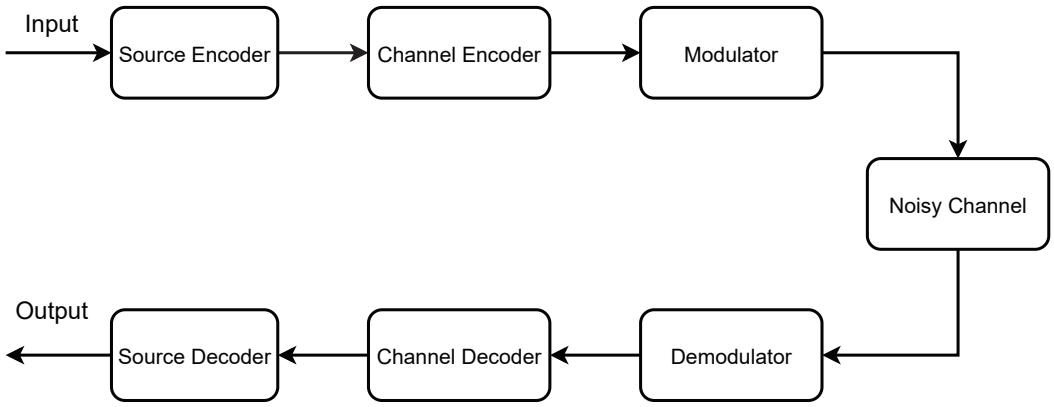


Figure 1.1: The architecture of a traditional digital communication system.

codes [10], and capacity-approaching codes such as turbo codes [11], LDPC [12], and polar codes [13], which have been widely used in the reliable transmission of various communication systems.

Although a system designed separately for source and channel coding has simple and flexible design advantages, it cannot fully use the statistical characteristics of source and channel information. With the recent development of the Internet of Things, there has been a growing interest in combining source and channel coding to integrate the system and improve transmission efficiency and effectiveness. Therefore, the joint source-channel coding technology, which designs the source encoder/decoder and channel encoder/decoder in an interdependent way to make the overall system achieving optimal performance, has become an important direction in coding theory research.

The concept of joint source-channel coding was first conceived more than 70 years ago [1]. It has been further investigated since the 1990's [14–16]. The main idea is to allow the source statistics and channel information to be utilized in the joint source-channel decoding algorithms. Various joint source-channel coding schemes have been proposed. A typical image source coder, that is discrete cosine transform coding, and a convolutional channel code are used in a joint source-channel coding system [17]. Considerable coding gains are achieved by utilizing the image residual redundancy as a priori soft-output Viterbi decoding process. Due to the outstanding error-correction performance of turbo codes [18] and LDPC codes [19], joint source-channel coding systems using them as channel codes together with a source code have

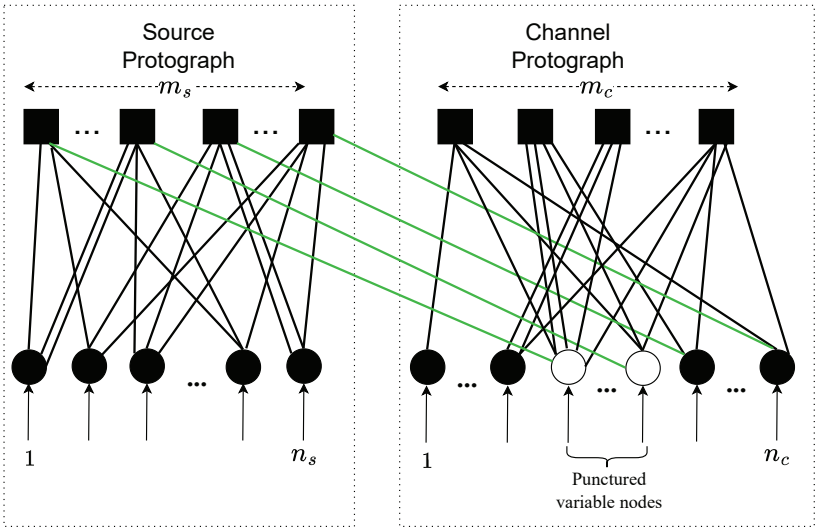


Figure 1.2: The protograph of a joint source-channel coding system using DP-LDPC codes [24]. Variable nodes and check nodes are represented by circles and squares, respectively. Punctured variable nodes are represented by blank circles.

been proposed [16, 20, 21]. The source code length is variable in these systems. Variable-length source codes increase the system implementation complexity and cannot match well with these channel codes with excellent performance. In [22, 23], a concatenated LDPC joint source-channel coding scheme is proposed. By using the LDPC code as the source code, a fixed-to-fixed length compressed source sequence is generated. Then the compressed information is protected by another LDPC code. An iterative source-channel BP decoding algorithm is used at the receiver and results in good error performance.

P-LDPC codes [25–27], which can be represented by small-size protographs consisting of VNs, CNs and parallel edges, are a sub-class of LDPC codes. They have parallel encoding and decoding structures and the linear minimum Hamming distance property. Besides, due to their small size, the theoretical performance can easily be evaluated by using the P-EXIT algorithm [28]. In [24], DP-LDPC codes replace double randomly constructed LDPC codes in the joint source-channel coding system, forming the DP-LDPC-based joint source-channel coding system. The protograph representation of this joint source-channel coding scheme is shown in Fig. 1.2. The source P-LDPC code and channel P-LDPC code are shown in the left dotted frame and the right dotted frame, respectively. Circles and squares represent VNs and CNs, respectively. In particular, punctured VNs in the channel protograph are represented by

blank circles. The green lines shown in Fig. 1.2 connecting CNs in the source protograph and VNs in the channel protograph in a one-to-one manner reflect the cascading relationship between the source encoder and the channel encoder. The protograph shown in Fig. 1.2 can be denoted by

$$\mathbf{B}_{\mathrm{J}_0} = \begin{pmatrix} \mathbf{B}_{\mathrm{s}} & \mathbf{B}_{\mathrm{sccv}} \\ \mathbf{0}_{m_c \times n_s} & \mathbf{B}_{\mathrm{c}} \end{pmatrix}, \tag{1.1}$$

where

$$\mathbf{B}_{\mathrm{sccv}} = \begin{pmatrix} \mathbf{0}_{m_s \times m_c} & \mathbf{I}_{m_s} \end{pmatrix}. \tag{1.2}$$

i) $\mathbf{B}_{\mathrm{s}}$ is the source protomatrix of size $m_s \times n_s$.

ii) $\mathbf{B}_{\mathrm{c}}$ is the channel protomatrix of size $m_c \times n_c$.

iii) $\mathbf{B}_{\mathrm{sccv}}$ is the SCCV linking protomatrix of size $m_s \times n_c$ [29], which consists of a zero matrix of size $m_s \times m_c$ and an identity matrix $\mathbf{I}_{m_s}$ of size $m_s \times m_s$. $\mathbf{I}_{m_s}$ is correponding to those green connections shown in Fig. 1.2.

Various optimization schemes for the DP-LDPC-based joint source-channel coding system have been proposed. An unequal error protection (UEP) technique [30] and an unequal power allocation technique [31] are respectively applied in the DP-LDPC-based joint source-channel coding system to improve the error performance. In [32], a source protograph-based extrinsic information transfer (SP-EXIT) algorithm is proposed to calculate the source thresholds of DP-LDPC codes, which provides theoretical guidance for matching source entropy and source encoding rate. In [33], a JP-EXIT algorithm is proposed for calculating the channel threshold of a DP-LDPC code. (Lower channel thresholds predict better waterfall region performance.) Then the channel protomatrix $\mathbf{B}_{\mathrm{c}}$ is redesigned based on the JP-EXIT algorithm results to obtain a performance improvement. In [34], the source protomatrix $\mathbf{B}_{\mathrm{s}}$ and the channel protomatrix $\mathbf{B}_{\mathrm{c}}$ are redesigned as a code pair to achieve coding gains. The optimal allocation of degree-2 VNs in the joint protomatrix $\mathbf{B}_{\mathrm{J}_0}$ (1.1) is studied in [35] and [36]. In [37], $\mathbf{B}_{\mathrm{J}_0}$ is redesigned as a whole and the resulting DP-LDPC codes have excellent error-correction capabilities. In [38],

the columns in the identity matrix $\mathbf{I}_{m_s}$ of $\mathbf{B}_{\text{sccv}}$ are exchanged to find the optimal connection between VNs in $\mathbf{B}_{\text{c}}$ and CNs in $\mathbf{B}_{\text{s}}$ so as to improve the error performance. In fact, swapping columns in $\mathbf{I}_{m_s}$ is equivalent to swapping rows in $\mathbf{B}_{\text{s}}$ or swapping columns in $\mathbf{B}_{\text{c}}$. This optimization scheme does not change the structure of DP-LDPC codes. All the above DP-LDPC codes are required to satisfy the common constraint defined in (1.2). Such a constraint allows easy encoding but largely restricts the design flexibility. Hence, the channel thresholds may not be optimized. In the traditional LDPC channel encoding, linear and fast encoding can be performed when the parity portion of the parity-check matrix possesses a lower or upper triangular structure [39,40]. Firstly, a new class of DP-LDPC codes which replaces the identity matrix $\mathbf{I}_{m_s}$ in the SCCV linking protomatrix $\mathbf{B}_{\text{sccv}}$ with a lower or upper triangular sub-base matrix with "1"s on its diagonal is proposed in this thesis. With the increased code design flexibility, we can design new DP-LDPC codes with lower channel thresholds while preserving the linear and fast source encoding property. The new class of DP-LDPC codes can outperform the original class of DP-LDPC codes in terms of theoretical channel thresholds and error rates while maintaining linear source encoding complexity.

It has been proved in [41–43] that compared with LDPC block codes, LDPC convolutional codes can obtain convolutional gain by using an iterative belief propagation algorithm. When LDPC convolutional codes are terminated, the corresponding codes are called SC-LDPC codes [44, 45]. In [46, 47], two spatially coupled regular LDPC codes are concatenated by an identity matrix in a joint source-channel coding system, and a sliding window joint source-channel decoder is exploited. These codes are shown to possess better error performance compared with DP-LDPC block codes used in [33]. Secondly, motivated by the result that spatially coupling the codes can improve the error performance, a joint source-channel coding system, where two spatially coupled SC-P-LDPC codes are spatially coupled is proposed in the thesis. Specifically, the parity-check matrices of two SC-P-LDPC codes are connected by a SC-SCCV linking matrix. By doing this, the flexibility of the code design is increased. Moreover, the correlation between codewords generated at different times is increased. The current compressed source symbols are not only related to the previous source symbols but also related to the previ-

ously generated channel codewords due to the spatially coupled structure of the SCCV linking protomatrix. We also set the first component base matrix for the SC-SCCV linking matrix with the same structure as the SCCV linking matrix in the new type of DP-LDPC codes to preserve the linear source compression. We call the proposed code "spatially-coupled double-protograph-based LDPC" (DP-LDPC).

The joint source-channel coding schemes mentioned above all focus on code optimization for low-entropy sources. There are only a few results for high-entropy sources. In [48], it is shown that error floors caused by the source compression can be lowered by adding connections between VNs in the source LDPC code and CNs in the channel LDPC code. In [49], similar connections, shown as the orange edges in Fig. 1.3, are added in the DP-LDPC codes. The relationship between the newly added connections and the error performance is investigated. In [50], some design rules for the new connections between VNs in the source P-LDPC and CNs in the channel P-LDPC are proposed for both low-entropy and high-entropy sources. In [29], the source protograph and the SVCC linking matrix are designed as a whole to obtain a high source threshold. Codes with higher source thresholds can be used to compress high-entropy sources without suffering from an error floor caused by the compression. Yet little research has been conducted in jointly designing the components of a code by considering both the source threshold and channel threshold, especially for high-entropy sources.

Thirdly, in this thesis, we propose a new type of joint source-channel block codes, whose SVCC linking matrix is a non-zero matrix for a high source threshold and SCCV linking matrix consists of a zero matrix and a lower or upper triangular matrix with "1"s on its diagonal for a low channel threshold. In other words, we add a non-zero SVCC linking base matrix to the new type of DP-LDPC codes to form a new type of joint source-channel block codes. We construct some new joint source-channel block codes for both low-entropy and high-entropy sources by considering both source and channel thresholds. Theoretical analyses and simulation results both show the proposed codes outperform state-of-the-art DP-LDPC codes for low-entropy and high-entropy sources. Moreover, we spatially couple the joint source-channel block codes and propose a new type of SC-JSCC. Compared with the structure of SC-DP-LDPC code, a SC-

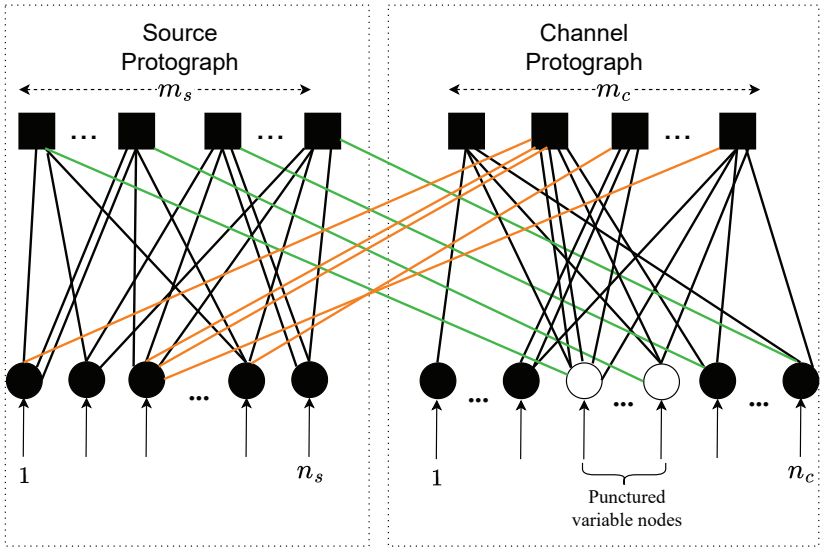


Figure 1.3: The protograph of a JSCC system using DP-LDPC codes [49] with connections between VNs in the source protograph and CNs in the channel protograph.

SVCC linking base matrix is newly added.

1.2 Outline of the thesis

The thesis is organized as follows. Chapter 1 shows the background and motivation of our research. The research results of DP-LDPC-based joint source-channel coding systems are reviewed. The shortcomings of the current DP-LDPC-based joint source-channel coding systems and our improved joint source-channel coding schemes are discussed. Chapter 2 introduces the basic concepts of P-LDPC codes, traditional DP-LDPC-based joint source-channel coding systems, SC-LDPC codes, and concatenated SC-LDPC codes, encoding and decoding methods and theoretical analysis methods. In Chapter 3, the new class of DP-LDPC-based joint source-channel coding system, where the identity matrix $\mathbf{I}_{m_s}$ in the SCCV linking protomatrix $\mathbf{B}_{\text{sccv}}$ is replaced with a lower or upper triangular sub-base matrix with "1"s on its diagonal, is introduced. Its encoder/decoder complexity and latency are also given. In Chapter 4, another new type of joint source-channel coding scheme, where the parity-check matrices of two SC-LDPC codes are connected by a SC-SCCV linking matrix, is illustrated. The first component base matrix for the SC-SCCV linking matrix has the same structure as the SCCV linking matrix

in the new type of DP-LDPC codes. It is called the SC-DP-LDPC based joint source-channel coding scheme. Theoretical thresholds and error performance show that SC-DP-LDPC codes constructed have better error performance compared with the concatenated SC-LDPC codes and the existing DP-LDPC block codes. In Chapter 5, the structure, the encoding method, the protograph-based analysis method, and the design method of the proposed JSC-BC are first introduced. Joint source-channel block codes consist of source protomatrix, channel protomatrix, a non-zero SVCC linking matrix and a SCCV linking matrix consists of a zero matrix and a lower or upper triangular matrix with "1"s on its diagonal. The source and channel thresholds and simulation results of the newly constructed JSC-BCs are also presented. Then, we introduce the structure and encoding method of the proposed SC-JSCC. Based on the JSC-BCs, we construct some SC-JSCCs. Each sub-block in the SC-JSCC (source protomatrix, channel protomatrix, SCCV linking base matrix, and SVCC linking base matrix) has a spatially coupled structure. Some error performance comparison results are also given. Chapter 6 gives the conclusion of the thesis.

Chapter 2

Related technologies

2.1 LDPC codes

In 1962, Gallager proposed the LDPC code [12], which is a linear block code with excellent error-correction performance. LDPC codes can be represented by a parity-check matrix or Tanner graph [51]. For example, a (N, K) LDPC code is considered. N and K denote the codeword length and the information length, respectively. The length of the parity-check sequence is $M = N - K$. The size of the parity-check matrix $\mathbf{H}$ is $M \times N$. The row weight denotes the number of "1"s in the row of $\mathbf{H}$. Similarly, the number of "1"s in the column of $\mathbf{H}$ is called the column weight. If the row weight and column weight are fixed, the LDPC code is called a regular LDPC code; otherwise, it is called an irregular LDPC code. Since the number of "1"s in $\mathbf{H}$ is very small, $\mathbf{H}$ is a sparse matrix with a low density of "1"s. Because of this feature, LDPC codes offer low implementation complexity.

For example, $\mathbf{H}$ in (2.1) denotes the parity-check matrix of a regular $(10, 5)$ LDPC code, whose row weight and column weight are, respectively, 4 and 2. It can also be represented by the Tanner graph shown in Fig. 2.1. The squares and circles represent check nodes ($\{C_1, C_2, ..., C_M\}$) and variable nodes (VNs) ($\{V_1, V_2, ..., V_N\}$), respectively. When the entry in the ith row and jth column of $\mathbf{H}$ is "1", there is an edge in Fig. 2.1 connecting the ith CN with the jth VN. The number of edges connecting to a VN is called its degree. If a node goes back to itself through a

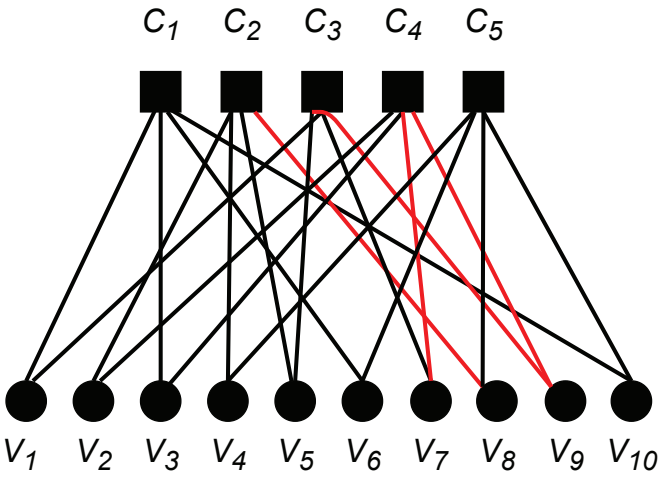
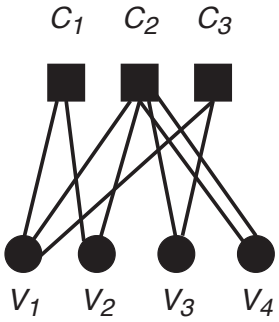


Figure 2.1: The Tanner graph of the parity-check matrix $\mathbf{H}$ in (2.1).

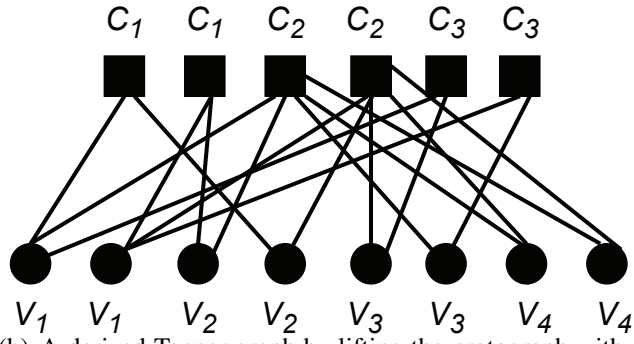
$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}. \tag{2.1}$$

path, forming a closed loop, the number of edges contained in this closed loop denotes the size of this cycle. The size of the shortest cycle in the Tanner graph is called the girth and is denoted by g. For example, there is a cycle of size 4 and the path is $V_7 \rightarrow C_4 \rightarrow V_9 \rightarrow C_3 \rightarrow V_7$ marked in red in Fig. 2.1. Short cycles have a negative influence on error performance. It is crucial to avoid them.

In 2003, J. Thorpe proposed the protograph-based LDPC (P-LDPC) codes [52], a sub-category of multi-edge type LDPC codes [53]. They have outstanding error-correction capability and low decoding complexity [54–56]. P-LDPC codes can be represented by a protomatrix or a protograph. We use $\mathbf{B}$ of size $m \times n$ to denote a protomatrix and use b_{ij} to indicate the entry in the ith row and jth column of $\mathbf{B}$, where $i = 1, 2, ..., m; j = 1, 2, ..., n$. b_{ij} can be any non-negative integer. For example, $\mathbf{B}$ is shown in (2.2), where $m = 3$ and $n = 4$. Its code rate is $R = (n - m)/n = 1/4 = 0.25$. Its protograph is shown in Fig. 2.2(a). We use $\boldsymbol{G} = (\boldsymbol{N}_v, \boldsymbol{N}_c, \boldsymbol{E})$ to represent the protograph. $\boldsymbol{N}_v$ and $\boldsymbol{N}_c$, respectively, denote the set of VNs



(a) The protograph of $\mathbf{B}$ in (2.2).



(b) A derived Tanner graph by lifting the protograph with a small lifting factor

Figure 2.2: The process of the first lifting with $z_1 = 2$.

$$\mathbf{B} = \begin{bmatrix} 1\,1\,0\,0 \\ 1\,1\,1\,2 \\ 1\,0\,1\,0 \end{bmatrix} \xrightarrow{z_1=2} \mathbf{B}_{z_1} = \begin{bmatrix} 1\,0\,0\,1\,0\,0\,0\,0 \\ 0\,1\,1\,0\,0\,0\,0\,0 \\ 1\,0\,1\,0\,0\,1\,1\,1 \\ 0\,1\,0\,1\,1\,0\,1\,1 \\ 1\,0\,0\,0\,1\,0\,0\,0 \\ 0\,1\,0\,0\,0\,1\,0\,0 \end{bmatrix} \xrightarrow{z_2} \mathbf{H}_{\text{QC}} = \begin{bmatrix} \mathbf{I}_{z_2}^0 & \mathbf{0}_{z_2} & \mathbf{0}_{z_2} & \mathbf{I}_{z_2}^0 & \mathbf{0}_{z_2} & \mathbf{0}_{z_2} & \mathbf{0}_{z_2} & \mathbf{0}_{z_2} \\ \mathbf{0}_{z_2} & \mathbf{I}_{z_2}^1 & \mathbf{I}_{z_2}^2 & \mathbf{0}_{z_2} & \mathbf{0}_{z_2} & \mathbf{0}_{z_2} & \mathbf{0}_{z_2} & \mathbf{0}_{z_2} \\ \mathbf{I}_{z_2}^0 & \mathbf{0}_{z_2} & \mathbf{I}_{z_2}^3 & \mathbf{0}_{z_2} & \mathbf{0}_{z_2} & \mathbf{I}_{z_2}^3 & \mathbf{I}_{z_2}^0 & \mathbf{I}_{z_2}^1 \\ \mathbf{0}_{z_2} & \mathbf{I}_{z_2}^2 & \mathbf{0}_{z_2} & \mathbf{I}_{z_2}^9 & \mathbf{I}_{z_2}^2 & \mathbf{0}_{z_2} & \mathbf{I}_{z_2}^7 & \mathbf{I}_{z_2}^3 \\ \mathbf{I}_{z_2}^0 & \mathbf{0}_{z_2} & \mathbf{0}_{z_2} & \mathbf{0}_{z_2} & \mathbf{I}_{z_2}^6 & \mathbf{0}_{z_2} & \mathbf{0}_{z_2} & \mathbf{0}_{z_2} \\ \mathbf{0}_{z_2} & \mathbf{I}_{z_2}^1 & \mathbf{0}_{z_2} & \mathbf{0}_{z_2} & \mathbf{0}_{z_2} & \mathbf{I}_{z_2}^8 & \mathbf{0}_{z_2} & \mathbf{0}_{z_2} \end{bmatrix} \tag{2.2}$$

and CNs. $\boldsymbol{E}$ represents the set of edges. Unlike the Tanner graph, there may be parallel edges between VNs and CNs in the protograph.

We can use two steps to lift the protograph to form a large low-density parity-check matrix with a quasi-cyclic structure [57–59]. Firstly, we lift the protomatrix $\mathbf{B}$ with a small lifting factor z_1 using the progressive-edge-growth (PEG) algorithm [60]. The objective of this lifting is to eliminate all entries with values larger than 1, thereby obtaining a matrix with only 0's and

1's. $\mathbf{B}_{z_1}$ shown in (2.2) is the protomatrix obtained by lifting $\mathbf{B}$ when $z_1 = 2$. Figure 2.2(b) is the Tanner graph obtained by lifting the protograph when $z_1 = 2$.

Secondly, we lift $\mathbf{B}_{z_1}$ with a lifting factor of z_2, forming a large parity-check matrix with the quasi-cyclic structure of appropriate size [61]. We use $\mathbf{I}_{z_2}^{h_{i,j}}$ $(0 \leq h_{i,j} < z_2)$ to replace each non-zero entry in $\mathbf{B}_{z_1}$. $\mathbf{I}_{z_2}^{h_{i,j}}$ represents a CPM with size $z_2 \times z_2$ obtained by cyclically right-shifting the identity matrix $\mathbf{I}_{z_2}$ by $h_{i,j}$ columns. For example, when $h_{i,j} = 3$, the corresponding CPM with $z_2 = 10$ is shown in (2.3). Moreover, we replace each zero in $\mathbf{B}_{z_1}$ with a zero matrix $\mathbf{0}_{z_2}$ of size $z_2 \times z_2$.

A toy example of the second lifting is shown in (2.2). $\mathbf{H}_{\text{QC}}$ denotes the resultant parity-check matrix by lifting $\mathbf{B}_{z_1}$ with $z_2 = 10$. Note that we try to maximize the girth of the resultant QC-LDPC matrix during the second lifting process.

$$\begin{bmatrix}
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix} \tag{2.3}$$

2.2 JSCC based DP-LDPC codes

Figure 2.3 depicts the protograph of a joint source-channel coding system using DP-LDPC codes [24]. VNs and CNs are denoted by circles (blank circles denote punctured VNs) and squares, respectively. The left dotted box of the figure includes a source P-LDPC code $\mathbf{B}_{\text{s}}$ of size $m_s \times n_s$ and the right one shows a channel P-LDPC code $\mathbf{B}_{\text{c}}$ of size $m_c \times n_c$. In the source

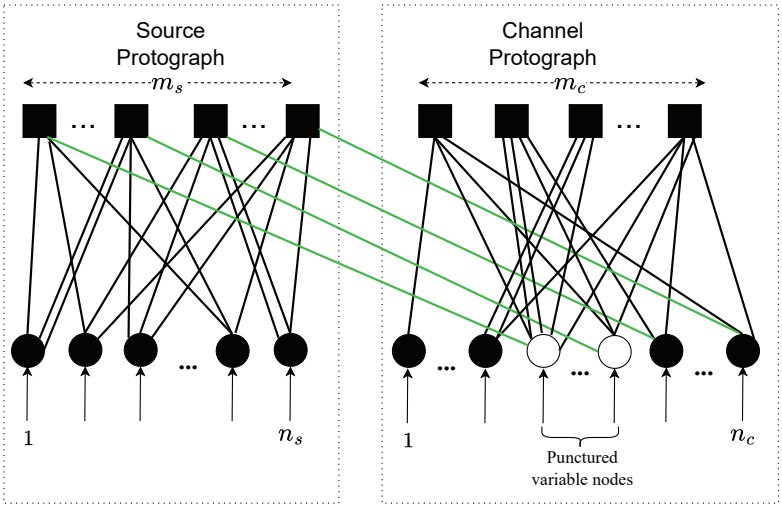


Figure 2.3: The protograph of the joint source-channel coding system based on double protograph-based LDPC codes.

protograph, VNs and CNs correspond to source symbols and compressed symbols, respectively. In the channel protograph, VNs correspond to the compressed symbols and parity-check bits, and CNs represent a series of parity-check equations. Punctured VNs are not transmitted. The source protograph and channel protograph are connected in a one-to-one manner, as shown by the green connections in Fig. 2.3. The DP-LDPC code can also be denoted by

$$\mathbf{B}_{\mathrm{J}_0} = \begin{pmatrix} \mathbf{B}_{\mathrm{s}} & \mathbf{B}_{\mathrm{sccv}} \\ \mathbf{0}_{m_c \times n_s} & \mathbf{B}_{\mathrm{c}} \end{pmatrix}, \tag{2.4}$$

where

$$\mathbf{B}_{\mathrm{sccv}} = \begin{pmatrix} \mathbf{0}_{m_s \times m_c} & \mathbf{I}_{m_s} \end{pmatrix}. \tag{2.5}$$

$\mathbf{B}_{\mathrm{sccv}}$ of size $m_s \times n_c$ is called the source-check-channel-variable (SCCV) linking protomatrix [29], which consists of a zero matrix of size $m_s \times (n_c - m_s) = m_c$ and an identity matrix $\mathbf{I}_{m_s}$ of size $m_s \times m_s$. $\mathbf{I}_{m_s}$ is corresponding to those green connections shown in Fig. 2.3. The overall code rate of the DP-LDPC code is $R = n_s/(n_c - n_p)$, where n_p denotes the number of punctured VNs in the channel P-LDPC code.

A large parity-check matrix can be obtained by lifting $\mathbf{B}_{\mathrm{J}_0}$ twice, resulting in

$$\mathbf{H}_{\mathrm{J}_0} = \begin{pmatrix} \mathbf{H}_{\mathrm{s}} & \mathbf{H}_{\mathrm{sccv}} \\ \mathbf{0}_{m_c z_1 z_2 \times n_s z_1 z_2} & \mathbf{H}_{\mathrm{c}} \end{pmatrix}, \tag{2.6}$$

where $\mathbf{H}_{\mathrm{s}}$, $\mathbf{H}_{\mathrm{c}}$, and $\mathbf{H}_{\mathrm{sccv}}$ are, respectively, the parity-check matrices obtained by lifting $\mathbf{B}_{\mathrm{s}}$, $\mathbf{B}_{\mathrm{c}}$, and $\mathbf{B}_{\mathrm{sccv}}$ twice.

2.2.1 Encoder

We use $\boldsymbol{s} = \{s_1, s_2, ..., s_{N_s}\} \in \{0,1\}^{1\times N_s}$ to represent the binary independent and identically distributed source sequence. $N_s = z_1 z_2 n_s$ denotes the source symbol length. We define $z = z_1 z_2$ as the overall lifting factor. The probability of "1" in $\boldsymbol{s}$ is denoted by p_1 ($p_1 \neq 0.5$). The probability of "0" in $\boldsymbol{s}$ equals $1 - p_1$. Therefore, the source entropy is calculated by

$$H = -p_1 \log_2 p_1 - (1-p_1)\log_2(1-p_1). \tag{2.7}$$

The encoding algorithm of a DP-LDPC code is introduced in detail below.

Step(a) Firstly, we use $\boldsymbol{c}$ to denote the compressed source sequence and it can be obtained by

$$\boldsymbol{c} = \boldsymbol{s}(\mathbf{H}_{\mathrm{s}})^{\mathrm{T}}, \tag{2.8}$$

where $(\mathbf{H}_{\mathrm{s}})^{\mathrm{T}}$ denotes the transpose of $\mathbf{H}_{\mathrm{s}}$.

Step(b) Secondly, $\boldsymbol{c}$ is the input for the channel encoder. We use the Gaussian elimination to transform $\mathbf{H}_{\mathrm{c}}$ to $\mathbf{H}'_{\mathrm{c}} = \begin{pmatrix} \mathbf{I}_{m_c z \times m_c z} & \mathbf{P}_{m_c z \times m_s z} \end{pmatrix}$, which has a systematic form. $\mathbf{I}_{m_c z \times m_c z}$ is an identity matrix of size $m_c z \times m_c z$. Next, we obtain the generating matrix $\mathbf{G}_{\mathrm{c}} = \begin{pmatrix} (\mathbf{P}_{m_c z \times m_s z})^{\mathrm{T}} & \mathbf{I}_{m_s z \times m_s z} \end{pmatrix}$. The channel codeword sequence $\boldsymbol{v}$ can be obtained by

$$\boldsymbol{v} = \boldsymbol{c}\mathbf{G}_{\mathrm{c}}. \tag{2.9}$$

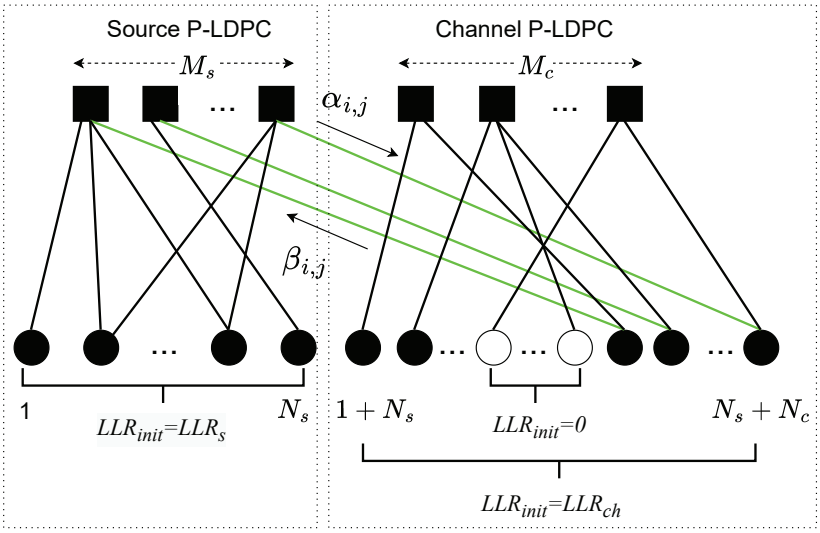


Figure 2.4: The Tanner graph shows the joint BP decoder.

Finally, the codeword sequence $\boldsymbol{v}$ excluding the punctured bits is modulated by binary-phase-shift-keying (BPSK) ("0" and "1" are mapped to "$+1$" and "-1", respectively), and transmitted over an additive white Gaussian noise (AWGN) channel.

2.2.2 Decoder

A joint source-channel belief propagation (BP) algorithm [62] is used to decode the DP-LDPC codes. The Tanner graph in Fig. 2.4 shows the iterative decoding process.

We assume the following parameters.

- I_{max} is the maximum number of decoding iterations.
- M_s and M_c, respectively, denote the number of check nodes in the source parity-check matrix and in the channel parity-check matrix. N_s and N_c, respectively, denote the number of source symbols and the number of variable nodes in the channel parity-check matrix.
- The noise variance of the AWGN channel is given by σ^2.
- $LLR_{init}(j)$ represents the initial LLR of the jth VN ($j = 1, 2, ..., N_s + N_c$).

- α_{ij} represents the LLR message sent from the ith CN to the jth VN. β_{ij} denotes the LLR message sent from the jth VN to the ith CN ($i = 1, 2, ..., M_s + M_c$ and $j = 1, 2, ..., N_s + N_c$).

- $L_{APP,j}$ denotes the a posteriori LLR of the jth VN.

- $\mathcal{M}(j)$ and $\mathcal{N}(i)$ represent the set of all CNs connected to the jth VN and the set of all VNs connected to the ith CN, respectively. $\mathcal{M}(j)\backslash i$ denotes the set of all CNs connected to the jth VN excluding the ith CN. $\mathcal{N}(i)\backslash j$ denotes the set of all VNs connected to the ith CN excluding the jth VN.

The following shows the decoding process.

Initialization:

- Set the iteration counter $t = 0$.

- Set $\alpha_{ij} = \beta_{ij} = 0 \; \forall i, j$.

- As shown in Fig. 2.4, LLR_{init} is generated by

$$LLR_{init}(j) = \begin{cases} LLR_s(j), & j = 1, 2, ..., N_s, \\ LLR_{ch}(j), & j = N_s + 1, N_s + 2, ..., N_s + N_c, \end{cases} \tag{2.10}$$

  where the channel initial LLR information $LLR_{ch}(j) = \log \frac{P(v_j=0)}{P(v_j=1)} + \log \frac{P(y_j|v_j=0)}{P(y_j|v_j=1)} = 2y_j/\sigma^2$ for an AWGN channel with BPSK modulation. v_j denotes the jth channel bit. The jth received signal $y_j = \pm 1 + \eta_j$, where $\eta_j \sim N(0, \sigma^2)$ denotes AGWN samples. For punctured bits, $LLR_{ch}(j) = 0$. The source a priori information is given $LLR_s(j) = \ln((1 - p_1)/p_1)$.

Iterative process:

- Start: Set the iteration counter $t = 1$.

- Step One: Updating LLRs from VNs to CNs by

$$\beta_{ij} = LLR_{init}(j) + \sum_{i' \in \mathcal{M}(j) \backslash i} \alpha_{i'j}, \ \forall i, j. \tag{2.11}$$

- Step Two: Updating LLRs from CNs to VNs by

$$\alpha_{ij} = 2 \tanh^{-1} \left(\prod_{j' \in \mathcal{N}(i) \backslash j} \tanh(\beta_{ij'}/2) \right), \ \forall i, j. \tag{2.12}$$

- Step Three: Calculating the posterior LLRs by:

$$L_{APP,j} = LLR_{init}(j) + \sum_{i \in \mathcal{M}(j)} \alpha_{ij}, \ \forall j. \tag{2.13}$$

- Step Four: The reconstructed codeword is given by:

$$\hat{v}_j = 0 \text{ if } L_{APP,j} \geq 0, \text{ otherwise } \hat{v}_j = 1, \ \forall j. \tag{2.14}$$

  The reconstructed codeword should meet the condition:

$$\hat{\boldsymbol{v}} \cdot (\mathbf{H}_{\mathrm{J}_0})^{\mathrm{T}} = \mathbf{0}, \tag{2.15}$$

  where $\hat{\boldsymbol{v}} = \{\hat{v}_1, \hat{v}_2, ..., \hat{v}_{N_s+N_c}\}$ denotes the reconstructed codeword sequence. If (2.15) is satisfied or $t = I_{max}$, the iteration process will stop. Otherwise the iteration counter t is increased by 1, and Step One to Step Four are repeated.

2.2.3 JP-EXIT algorithm

For DP-LDPC codes, we can use the joint protograph-based extrinsic information transfer (JP-EXIT) algorithm [33] to calculate the channel threshold $(E_s/N_0)_{\mathrm{th}}$. $(E_s/N_0)_{\mathrm{th}}$ is the smallest E_s/N_0 value (in dB) which can make the mutual information (MI) between the a posterior log-

likelihood-ratios (APP-LLRs) evaluated by variable nodes and their corresponding symbols converging to "1" for a given p_1. E_s and N_0 denote the average transmitted energy per source symbol and the AWGN noise power spectral density, respectively. A small $(E_s/N_0)_{\text{th}}$ predicts good waterfall error performance, i.e., good performance in the small E_s/N_0 region.

We first define the following parameters.

- $I_{\text{A\_VC}}(i,j)$ and $I_{\text{E\_VC}}(i,j)$ denote the a priori mutual information (AMI) and the extrinsic mutual information (EMI) from the jth VN to the ith CN, respectively.
- $I_{\text{A\_CV}}(i,j)$ and $I_{\text{E\_CV}}(i,j)$ denote the AMI and the EMI from the ith CN to the jth VN, respectively.
- $I_{\text{APP}}(j)$ denotes the mutual information (MI) between the a posteriori log-likelihood-ratio (APP-LLR) of the jth VN and its corresponding symbol.
- We use $F_{\max}$ to denote the maximum number of iterations in the JP-EXIT algorithm. Δ and θ denote the step size and tolerance value, respectively.

The definitions $J_{\text{BSC}}(\cdot)$, $J(\cdot)$, and $J^{-1}(\cdot)$ [34] are given here.

$$J(\sigma_{ch}) = 1 - \int_{-\infty}^{\infty} \frac{e^{-(x-\sigma_{ch}^2/2)^2/(2\sigma_{ch}^2)}}{\sqrt{2\pi\sigma_{ch}^2}} \log_2[1-e^{-x}]dx, \tag{2.16}$$

where σ_{ch}^2 denotes the variance of LLR obtained by a transmitted VN.

The inverse $J(\cdot)$ function is given by

$$J^{-1}(I) = \begin{cases} a_1 I^2 + b_1 I + c_1\sqrt{I}, \ 0 \le I \le 0.3646, \\ -a_2 \ln[b_2(1-I)] - c_2 I, \ 0.3646 < I < 1, \end{cases} \tag{2.17}$$

where $a_1 = 1.09542$, $b_1 = 0.214117$, $c_1 = 2.33727$, $a_2 = 0.706692$, $b_2 = 0.386013$, and $c_2 = -1.75017$.

$$J_{\text{BSC}}(u, p_1) = (1-p_1) \times I(V_s; \xi^{1-p_1}) + p_1 \times I(V_s; \xi_1^p), \tag{2.18}$$

where u denotes the average LLR value obtained by the source variable node V_s. $I(V_s;\xi_1^p/\xi^{1-p_1})$ denotes the mutual information between V_s and ξ_1^p/ξ^{1-p_1}. $\xi^{1-p_1} \sim N(u+LLR_s,2u)$ and $\xi^{p_1} \sim N(u-LLR_s,2u)$.

Algorithm 1 shows the process of obtaining $(E_s/N_0)_{\text{th}}$.

Algorithm 1 JP-EXIT algorithm.

Given a protomatrix $\mathbf{B}_{\text{J}_0}$ and set $F_{\text{flag}} =$ true.

Set a sufficiently small E_s/N_0.

while F_{flag} **do**

For $\forall i,j$, set $I_{\text{A\_VC}}(i,j) = I_{\text{E\_VC}}(i,j) = I_{\text{A\_CV}}(i,j) = I_{\text{E\_VC}}(i,j) = 0$ and $I_{\text{APP}}(j) = 0$, and set the iteration counter $f = 1$.

while $f \le F_{max}$ **do**

Update MI from VNs to CNs:

for $\forall i,j$ **do**

Calculate $I_{\text{E\_VC}}(i,j) =$

$$\begin{cases} \psi(e_{i,j})J_{\text{BSC}}\left(\sum\limits_{i'\neq i} e_{i',j}[J^{-1}(I_{\text{A\_CV}}(i',j))]^2 + (e_{i,j}-1)[J^{-1}(I_{\text{A\_CV}}(i,j))]^2, p_1\right), \\ \qquad\qquad 1 \le j \le n_s, \\ \psi(e_{i,j})J\left(\sqrt{\sum\limits_{i'\neq i} e_{i',j}[J^{-1}(I_{\text{A\_CV}}(i',j))]^2 + (e_{i,j}-1)[J^{-1}(I_{\text{A\_CV}}(i,j))]^2 + \sigma_{ch,j}^2}\right), \\ \qquad\qquad n_s < j \le (n_s+n_c), \end{cases} \tag{2.19}$$

where $e_{i,j}$ denote the entry in row i and column j of $\mathbf{B}_{\text{J}_0}$. $\psi(e_{i,j}) = 0$ when $e_{i,j} = 0$, otherwise $\psi(e_{i,j}) = 1$. $\sigma_{ch,j} = 8R(E_s/N_0)$ when the jth VN is not punctured. $\sigma_{ch,j} = 0$ when the jth VN is punctured.

end for

Set $I_{\text{A\_VC}}(i,j) = I_{\text{E\_VC}}(i,j),\ \forall i,j$.

Update MI from CNs to VNs:

for $\forall i,j$ **do**

Calculate $I_{\text{E\_CV}}(i,j) =$

$$\psi(e_{i,j})\left(1-J\left(\sqrt{\sum_{j'\neq j} e_{i,j'}[J^{-1}(1-I_{\text{A\_VC}}(i,j'))]^2+(e_{i,j}-1)[J^{-1}(1-I_{\text{A\_VC}}(i,j))]^2}\right)\right). \tag{2.20}$$

end for

Set $I_{\text{A\_CV}}(i,j) = I_{\text{E\_CV}}(i,j),\ \forall i,j.$

Calculate

$$I_{\text{APP}}(j) = \begin{cases} J_{\text{BSC}}\left(\sum_i e_{i,j}[J^{-1}(I_{\text{A\_CV}}(i,j))]^2, p_1\right), \\ \qquad\qquad j = 1,2,...,n_s, \\ J\left(\sqrt{\sum_i e_{i,j}[J^{-1}(I_{\text{A\_CV}}(i,j))]^2+\sigma_{ch,j}^2}\right), \\ \qquad\qquad j = n_s+1,...,n_s+n_c. \end{cases} \tag{2.21}$$

if $\sum_{j=1}^{n_s+n_c}(1-I_{\text{APP}}(j)) < \theta$ **then**

Set $(E_s/N_0)_{\text{th}} = E_s/N_0$ and $F_{\text{flag}} =$ false

break

else

Set $f = f+1$.

end if

end while

Set $E_s/N_0 = E_s/N_0 + \Delta$

end while

A source threshold p_{th} is defined as the largest value of p_1 which makes $I_{\text{APP}}(j)$ for $j = 1,2,\ldots,n_s$ reach "1" when the channel information is perfect, i.e., E_s/N_0 is large enough. A high source threshold means the code can adapt to high-entropy sources.

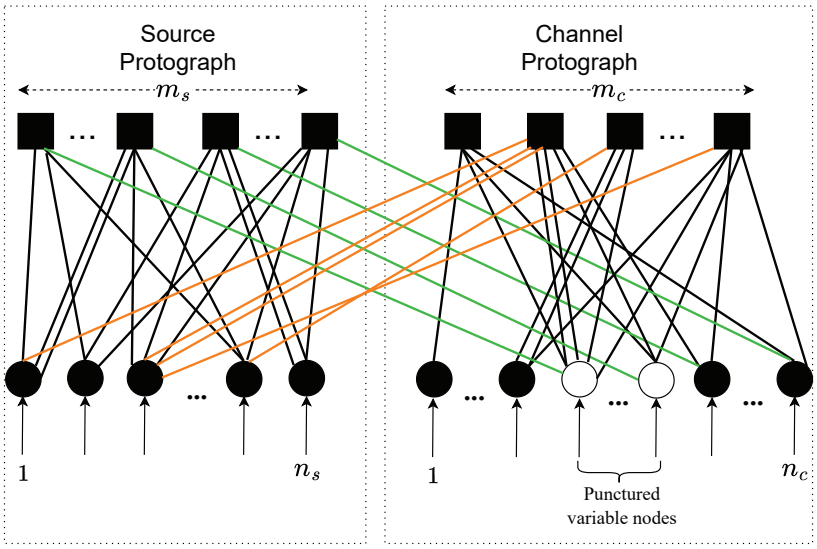


Figure 2.5: The protograph of DP-LDPC codes with a SVCC linking base matrix.

2.2.4 DP-LDPC codes with a SVCC linking matrix

It has been proved in [29, 48–50] that adding connections between VNs in the source P-LDPC code and CNs in the channel P-LDPC code can obtain DP-LDPC codes with high source thresholds. Those connections shown as orange lines in Fig. 2.5 form the source-variable-channel-check (SVCC) linking base matrix $\mathbf{B}_{\text{svcc}}$. This type of DP-LDPC codes is denoted by

$$\mathbf{B}_{\text{J}}^{*} = \begin{pmatrix} \mathbf{B}_{\text{s}} & \mathbf{B}_{\text{sccv}} \\ \mathbf{B}_{\text{svcc}} & \mathbf{B}_{\text{c}} \end{pmatrix}. \tag{2.22}$$

The parity-check matrix obtained by lifting $\mathbf{B}_{\text{J}}^{*}$ is denoted by

$$\mathbf{H}_{\text{J}}^{*} = \begin{pmatrix} \mathbf{H}_{\text{s}} & \mathbf{H}_{\text{sccv}} \\ \mathbf{H}_{\text{svcc}} & \mathbf{H}_{\text{c}} \end{pmatrix}. \tag{2.23}$$

There is a small difference in the encoding methods of the DP-LDPC codes with and without the SVCC linking matrix. Firstly, we obtain the compressed source sequence $\boldsymbol{c}$ using the same method shown in (2.8). Next, we combine $\boldsymbol{s}$ and $\boldsymbol{c}$ as the input for the channel encoder. Channel encoding is performed based on the parity-check matrix $[\mathbf{H}_{\text{svcc}}\ \mathbf{H}_{\text{c}}]$ to obtain the parity-check bit sequence $\boldsymbol{p}$. $\boldsymbol{s}$ and punctured bits in the channel codeword $\boldsymbol{v} = \{\boldsymbol{p}, \boldsymbol{c}\}$ are not transmitted.

$\mathbf{H}_{\mathrm{J}}^{*}$ has the similar decoding method and threshold calculation method as $\mathbf{H}_{\mathrm{J}_0}$, i.e., the joint BP decoder and JP-EXIT algorithm.

2.3 SC-LDPC codes

In 1999, Alberto Jiménez Felström and Kamil Sh. Zigangiro proposed the concept of LDPC convolutional code based on LDPC code and also gave its encoding and decoding algorithm [63]. Convolutional LDPC codes combine the advantages of LDPC block codes and convolutional codes [64, 65]. They not only have excellent error correction performance, but also can meet the high-speed encoding and decoding requirements. In 2011, Michael Lentmaier et al. proposed LDPC convolutional terminated codes [66], and proved these codes have threshold saturation properties when the coupling chain is long enough. This means their BP decoding thresholds can approximately reach the maximum a posteriori probability decoding thresholds when the code length is large enough. In the same year, Shrivinivas Kudekar et al. [67] proved that the characteristics of threshold saturation allow the codes to approach the Shannon limit in theory. The LDPC convolutional terminated codes are also called spatially coupled LDPC (SC-LDPC) codes [68]. The SC-LDPC code can be constructed based on the P-LDPC block codes by using the edge spreading technique [69, 70]. When the SC-LDPC code length is too long, a good decoding method needs to be used to decrease the decoding complexity and decoding latency. A sliding window belief propagation (BP) decoding method [41, 43, 71, 72] is usually used to decode the SC-LDPC codes.

We present the structure of SC-LDPC codes. Figure 2.6 shows an example of the construction process of the SC-LDPC code based on an LDPC block code. Circles and squares, respectively, denote check nodes (CNs) and variable nodes (VNs). Figure 2.6(a) shows the protograph of an LDPC block code. It is a $(3, 6)$-regular LDPC block code, where 3 and 6 denote the degrees of VN and CN, respectively. Figure 2.6(b) shows the protograph of an LDPC convolutional code. The protograph shown in Fig. 2.6(a) is repeated infinitely and edges are rearranged. By coupling repeated protographs of the LDPC block code, a VN is connected with

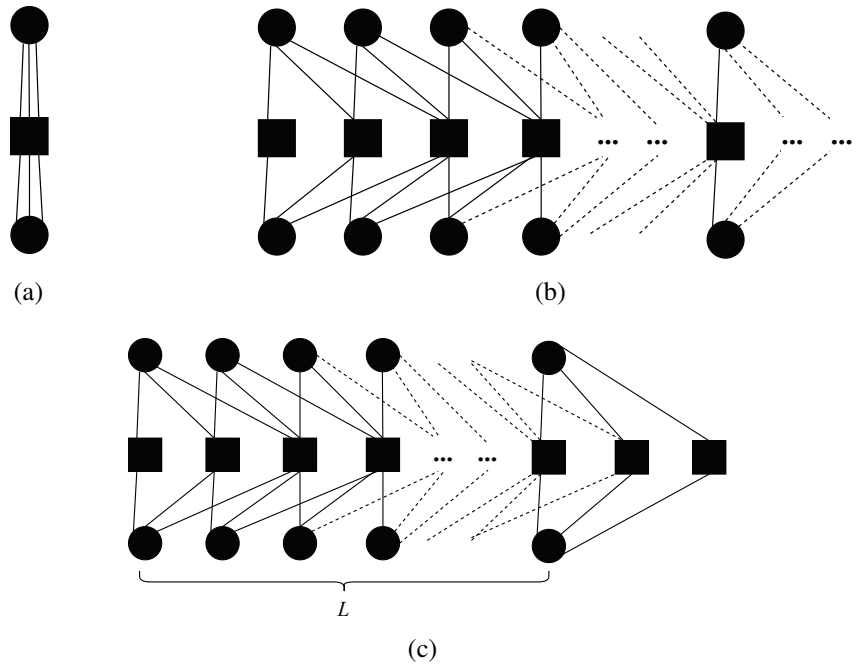


Figure 2.6: (a) The protograph of an LDPC block code. (b) The protograph of an LDPC convolutional code. (c) The protograph of an SC-LDPC code.

$m+1$ CNs adjacent to it. m denotes the syndrome former memory. When we terminate the LDPC convolutional code, the protograph of an SC-LDPC code is shown in Fig. 2.6(c). L is the coupling length, i.e., the number of repeated time of the LDPC block code.

We can use the protomatrix $\mathbf{B} = \begin{pmatrix} 3 & 3 \end{pmatrix}$ to denote the protograph in Fig. 2.6(a). We can use protomatrices $\mathbf{B}_{\text{conv}}$ and $\mathbf{B}_{\text{sc}}$ shown in Fig. 2.7(a) and Fig. 2.7(b) to denote the protographs in Fig. 2.6(b) and Fig. 2.6(c), respectively. When we use the edge spreading technique [69, 70] to construct an LDPC convolutional code or an SC-LDPC code based on an LDPC block code, the sum of $\mathbf{B}_0$ to $\mathbf{B}_m$ in $\mathbf{B}_{\text{conv}}$ or $\mathbf{B}_{\text{sc}}$ should be equal to $\mathbf{B}$. For $\mathbf{B}_{\text{conv}}$ or $\mathbf{B}_{\text{sc}}$ constructed based on $\mathbf{B} = \begin{pmatrix} 3 & 3 \end{pmatrix}$, $m = 2$ and $\mathbf{B}_0 = \mathbf{B}_1 = \mathbf{B}_2 = \begin{pmatrix} 1 & 1 \end{pmatrix}$. We use b_{c} and $b_{\boldsymbol{v}}$ to denote the number of CNs and VNs in $\mathbf{B}$, respectively. For Fig. 2.6(a), $b_{\text{c}} = 1$ and $b_{\text{v}} = 2$. Its code rate $R = (b_{\text{v}} - b_{\text{c}})/b_{\text{v}} = 0.5$. The code rate of an LDPC convolutional code is the same as R. The code rate of an SC-LDPC code is $R_{\text{sc}} = R - \frac{m}{L}\frac{b_{\text{c}}}{b_{\text{v}}}$.

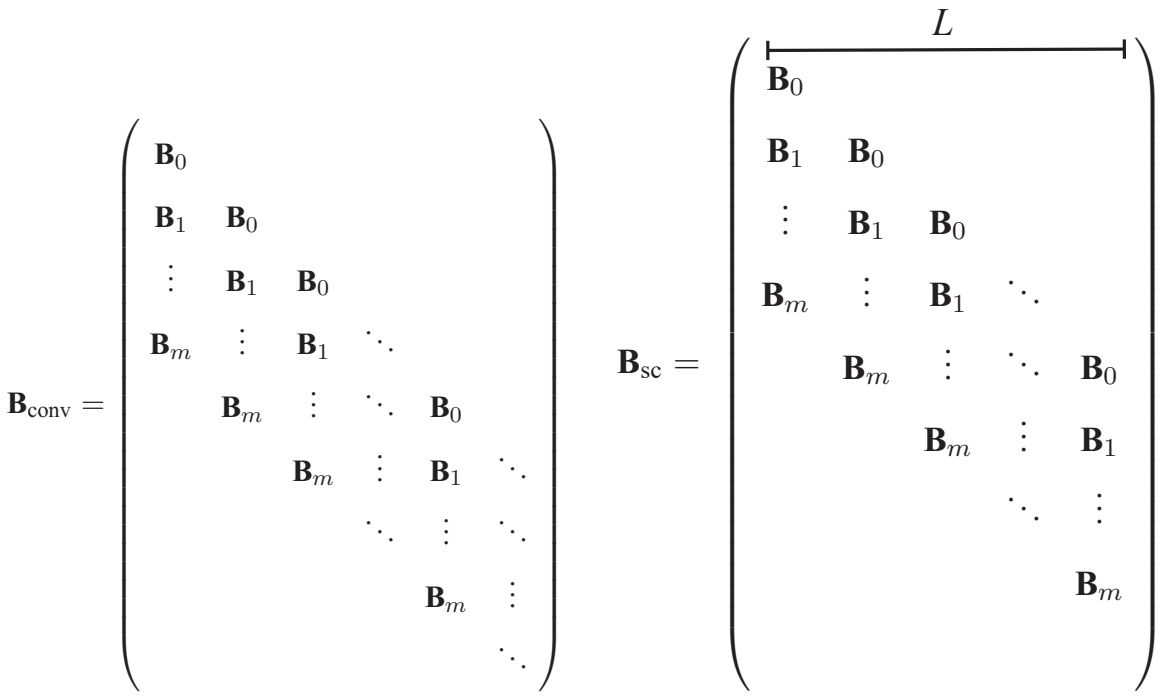


Figure 2.7: (a)The protomatrix of an LDPC convolutional code. (c) The protomatrix of an SC-LDPC code. L is the coupling length. For protographs in Fig. 2.6, $m = 2$ and $\mathbf{B}_0 = \mathbf{B}_1 = \mathbf{B}_2 = \begin{pmatrix} 1 & 1 \end{pmatrix}$.

2.4 Concatenated SC-LDPC codes for joint source-channel coding

Based on the excellent error-correction capability of SC-LDPC codes, Ahmad Golmohammadi and David G. M. Mitchell proposed a concatenated SC-LDPC joint source-channel coding scheme in [46, 47]. We take an example to introduce the structure of a concatenated SC-LDPC code and its encoding and decoding methods. A source SC-LDPC code is constructed based on a $(3, 12)$-regular LDPC block code, i.e., $\mathbf{B}_{\text{s}} = \begin{pmatrix} 3 & 3 & 3 & 3 \end{pmatrix}$. A channel SC-LDPC code is constructed based on a $(3, 6)$-regular LDPC block code, i.e., $\mathbf{B}_{\text{c}} = \begin{pmatrix} 3 & 3 \end{pmatrix}$. The syndrome former memories of source SC-LDPC code and channel SC-LDPC code are, respectively, $m_0 = 2$ and $m_1 = 2$. The sub-block protomatrices of source SC-LDPC code are $\mathbf{B}_{\text{s}_0} = \mathbf{B}_{\text{s}_1} = \mathbf{B}_{\text{s}_2} = \begin{pmatrix} 1 & 1 & 1 & 1 \end{pmatrix}$. They have $m_s = 1$ CNs and $n_s = 4$ VNs. The sub-block protomatrices of channel SC-LDPC code are $\mathbf{B}_{\text{c}_0} = \mathbf{B}_{\text{c}_1} = \mathbf{B}_{\text{c}_2} = \begin{pmatrix} 1 & 1 \end{pmatrix}$. They have $m_c = 1$

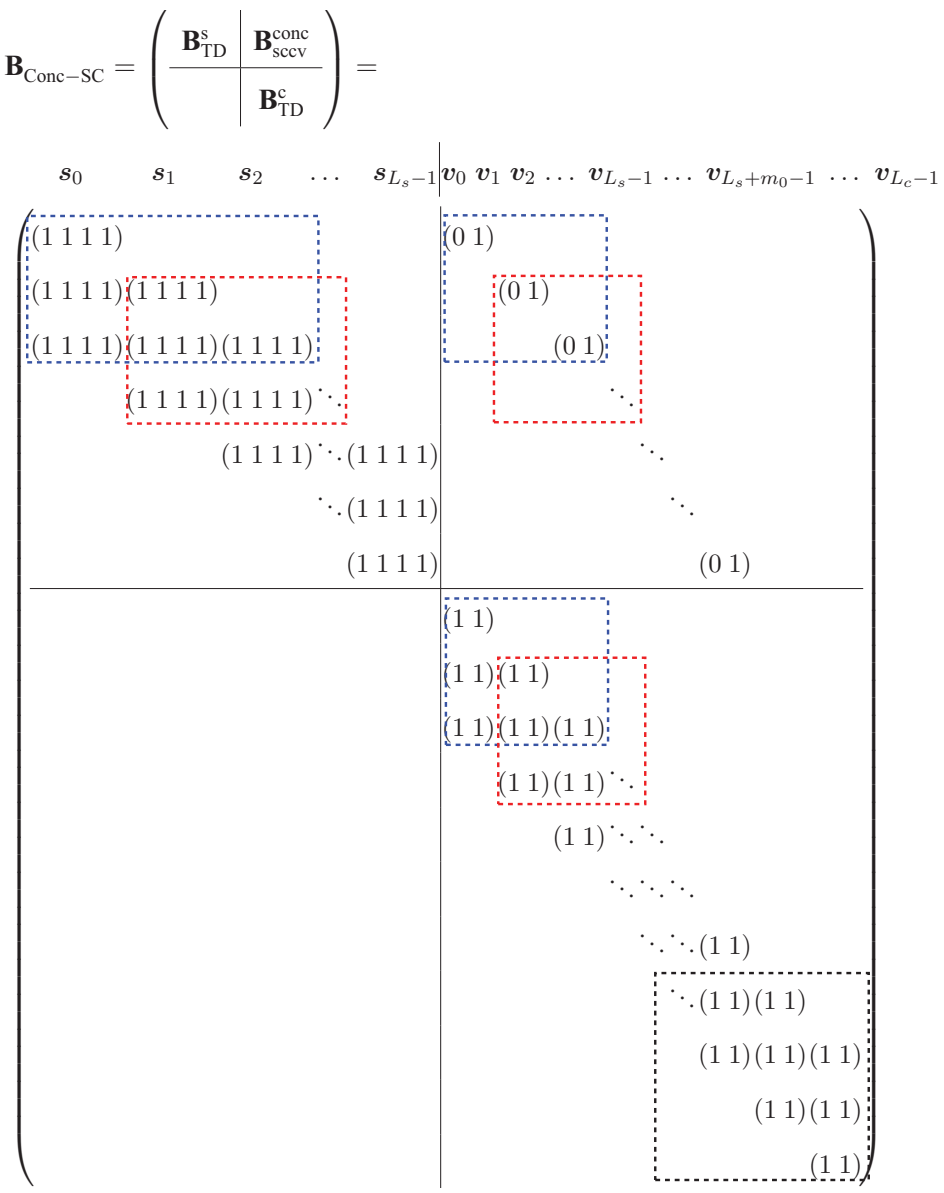


Figure 2.8: The protomatrix of a concatenated SC-LDPC code constructed based on a source $(3, 12)$-regular LDPC block code and a channel $(3, 6)$-regular LDPC block code.

CNs and $n_c = 2$ VNs. The source SC-LDPC code and channel SC-LDPC code are concatenated by using an identity matrix of size $m_s \times m_s$, i.e., 1 in $\begin{pmatrix} 0 & 1 \end{pmatrix}$ shown in Fig. 2.8. L_s and L_c shown in Fig. 2.8 are the coupling lengths of source SC-LDPC and channel SC-LDPC codes, respectively.

2.4.1 Encoder

At time t ($= 0, 1, 2, \ldots, L_s - 1$), the source sequence of size $1 \times n_s z$ (z is the overall lifting factor) is denoted by $\boldsymbol{s}_t$. Entries in $\boldsymbol{s}_t$ are "0" or "1". The probability of "1" in $\boldsymbol{s}_t$ is denoted by p_1 and the probability of "0" given by $1 - p_1$. The distribution of "0" and "1" in $\boldsymbol{s}_t$ follows a Bernoulli distribution. The compressed source sequence of size $1 \times m_s z$ is denoted by $\boldsymbol{c}_t$ at time t. The parity-check bit sequence of size $1 \times m_c z$ for channel encoder is denoted by $\boldsymbol{p}_t$ at time t. The channel codeword sequence of size $1 \times n_c z$ is denoted by $\boldsymbol{v}_t$ at time t. $\boldsymbol{v}_t$ consists of $\boldsymbol{c}_t$ and $\boldsymbol{p}_t$, i.e., $\boldsymbol{v}_t = [\boldsymbol{p}_t, \boldsymbol{c}_t]$.

1. At time $t = 0$, we firstly generate the compressed sequence $\boldsymbol{c}_0$ by calculating the syndrome based on the input $\boldsymbol{s}_0$ and the quasi-cyclic (QC) parity-check matrix obtained by lifting $\mathbf{B}_{\mathrm{s}_0} = \begin{pmatrix} 1 & 1 & 1 & 1 \end{pmatrix}$ on the first row of $\mathbf{B}^{\mathrm{s}}_{\mathrm{TD}}$. Next, $\boldsymbol{c}_0$ is regarded as the input for the channel encoder at time $t = 0$ and we generate $\boldsymbol{p}_0$ based on the QC parity-check matrix obtained by lifting $\mathbf{B}_{\mathrm{c}_0} = \begin{pmatrix} 1 & 1 \end{pmatrix}$ on the first row of $\mathbf{B}^{\mathrm{c}}_{\mathrm{TD}}$. Based on the SCCV linking base matrix shown in Fig. 2.8, i.e., $\begin{pmatrix} 0 & 1 \end{pmatrix}$, the left VN in $\mathbf{B}_{\mathrm{c}_0}$ is corresponding to $\boldsymbol{p}_0$ and the right VN in $\mathbf{B}_{\mathrm{c}_0}$ is corresponding to $\boldsymbol{c}_0$.

2. At time $t = 1$, we firstly generate the compressed sequence $\boldsymbol{c}_1$ by calculating the syndrome based on $[\boldsymbol{s}_0, \boldsymbol{s}_1]$ and the QC parity-check matrix obtained by lifting $\begin{pmatrix} \mathbf{B}_{\mathrm{s}_1} & \mathbf{B}_{\mathrm{s}_0} \end{pmatrix}$ on the second row of $\mathbf{B}^{\mathrm{s}}_{\mathrm{TD}}$. Next, $\boldsymbol{c}_1$ together with $\boldsymbol{v}_0$ is regarded as the input for the channel encoder at time $t = 1$ and we generate $\boldsymbol{p}_1$ based on the QC parity-check matrix obtained by lifting $\begin{pmatrix} \mathbf{B}_{\mathrm{c}_1} & \mathbf{B}_{\mathrm{c}_0} \end{pmatrix} = \begin{pmatrix} (1\ 1) & (1\ 1) \end{pmatrix}$ on the second row of $\mathbf{B}^{\mathrm{c}}_{\mathrm{TD}}$.

3. At time $2 \leq t < L_s$, we firstly generate the compressed sequence $\boldsymbol{c}_t$ by calculating the syndrome based on $[\boldsymbol{s}_{t-2}, \boldsymbol{s}_{t-1}, \boldsymbol{s}_t]$ and the QC parity-check matrix obtained by lifting $\begin{pmatrix} \mathbf{B}_{\mathrm{s}_2} & \mathbf{B}_{\mathrm{s}_1} & \mathbf{B}_{\mathrm{s}_0} \end{pmatrix}$ on the $(t+1)$-th row of $\mathbf{B}^{\mathrm{s}}_{\mathrm{TD}}$. Next, $\boldsymbol{c}_t$ together with $[\boldsymbol{v}_{t-2}, \boldsymbol{v}_{t-1}]$ is regarded as the input for the channel encoder at time t and we generate $\boldsymbol{p}_t$ based on the QC parity-check matrix obtained by lifting $\begin{pmatrix} \mathbf{B}_{\mathrm{c}_2} & \mathbf{B}_{\mathrm{c}_1} & \mathbf{B}_{\mathrm{c}_0} \end{pmatrix} = \begin{pmatrix} (1\ 1) & (1\ 1) & (1\ 1) \end{pmatrix}$ on the $(t+1)$-th row of $\mathbf{B}^{\mathrm{c}}_{\mathrm{TD}}$.

4. There are no new inputs to generate $\boldsymbol{v}_{L_s}, \ldots, \boldsymbol{v}_{L_s+m_0-1}$ ($m_0 = 2$ here). For $L_s \leq t \leq L_s + m_0 - 1$, we firstly generate the compressed sequence $\boldsymbol{c}_t$ by calculating the syndrome based on $[\boldsymbol{s}_{t-2}, \ldots, \boldsymbol{s}_{L_s-1}]$ and the QC parity-check matrix obtained by lifting $\begin{pmatrix} \mathbf{B}_{\mathrm{s}_{L_s+1-t}} & \cdots & \mathbf{B}_{\mathrm{s}_0} \end{pmatrix}$ on the $(t+1)$-th row of $\mathbf{B}^{\mathrm{s}}_{\mathrm{TD}}$. Next, $\boldsymbol{c}_t$ together with $[\boldsymbol{v}_{t-2}, \boldsymbol{v}_{t-1}]$ is regarded as the input for the channel encoder at time t and we generate $\boldsymbol{p}_t$ based on the QC parity-check matrix obtained by lifting $\begin{pmatrix} \mathbf{B}_{\mathrm{c}_2} & \mathbf{B}_{\mathrm{c}_1} & \mathbf{B}_{\mathrm{c}_0} \end{pmatrix} = \begin{pmatrix} (1\ 1) & (1\ 1) & (1\ 1) \end{pmatrix}$ on the $(t+1)$-th row of $\mathbf{B}^{\mathrm{c}}_{\mathrm{TD}}$.

5. At time $t > L_s + m_0 - 1$ ($m_0 = 2$ here), extra channel codeword sequences consisting of parity-check bits need to be added to check the left $m_1 = 2$ block rows of $\mathbf{B}^{\mathrm{c}}_{\mathrm{TD}}$, shown in the first two block rows of the sub-protomatrix in the black frames of Fig. 2.8. We use $\mathbf{B}_{\mathrm{remaining}}$ (2.24) to denote the sub-protomatrix in the black frame of Fig. 2.8.

$$\mathbf{B}_{\mathrm{remaining}} = \begin{array}{c} \begin{array}{cc|cc} \boldsymbol{v}_{L_s} & \boldsymbol{v}_{L_s+1} & \boldsymbol{v}_{L_s+2} & \boldsymbol{v}_{L_s+3} \end{array} \\ \left(\begin{array}{cc|cc} (1\ 1) & (1\ 1) & (1\ 1) & \\ & (1\ 1) & (1\ 1) & (1\ 1) \\ \hline & & (1\ 1) & (1\ 1) \\ & & & (1\ 1) \end{array}\right) \end{array}. \tag{2.24}$$

To ensure that encoding could be performed based on $\mathbf{B}_{\mathrm{remaining}}$ and the known $\boldsymbol{v}_{L_s}$ (i.e., $\boldsymbol{v}_{L_s+m_0-m_1}$) and $\boldsymbol{v}_{L_s+1}$ (i.e., $\boldsymbol{v}_{L_s+m_0-1}$), the number of parity-check equations in $\mathbf{B}_{\mathrm{remaining}}$ should be no more than the number of variable nodes in $\boldsymbol{v}_{L_s+2}$ (i.e., $\boldsymbol{v}_{L_s+m_0}$), $\cdots$, $\boldsymbol{v}_{L_c-1}$, i.e.,

$$L_{\mathrm{extra}} + m_1 \leq n_c L_{\mathrm{extra}} \rightarrow L_{\mathrm{extra}} + 2 \leq 2L_{\mathrm{extra}}, \tag{2.25}$$

where L_{extra} denotes the number of block columns added after the block column corresponding to $\boldsymbol{v}_{L_s+1}$ (i.e., after the (L_s+2)-th block column of $\mathbf{B}^{\mathrm{c}}_{\mathrm{TD}}$). Therefore, we obtain $L_{\mathrm{extra}} = 2$, which is the smallest integer selected such that (2.25) is satisfied. Next, we can calculate the channel coupling length by $L_c = L_s + m_0 + L_{\mathrm{extra}} = L_s + 4$.

2.4.2 A joint sliding window-based decoder

A joint sliding window-based belief propagation (BP) algorithm is proposed in [47] to decode concatenated SC-LDPC codes. w denotes the window size. We use the protomatrix of concatenated SC-LDPC codes shown in Fig. 2.8 to illustrate the decoding method. A window with size w includes w sub-block rows and w sub-block columns of the source SC-LDPC protomatrix, the channel SC-LDPC protomatrix, and the SCCV linking protomatrix. The blue dashed frame in Fig. 2.8 includes the sub-protomatrices of the concatenated SC-LDPC codes in a window of size $w = 3$. In each window, the first $n_s = 4$ VNs of the source sub-protograph and the first $n_c = 2$ VNs of the channel protograph can be decoded by using the joint BP algorithm introduced in Section 2.2.2.

In the next decoding timeslot, the window in Fig. 2.8 will slide to the right and downwards, i.e., move to the red dashed frame. Moreover, all updated log-likelihood ratio (LLR) messages and previously decoded source symbols and channel codewords may be used to facilitate decoding the source symbols and channel codewords in this window.

The detailed decoding process is described in **Algorithm 2**. We first define the parameters as follows.

- $I_{\max}$ is the maximum number of decoding iterations.
- Binary phase-shift-keying (BPSK) modulation is used where bit "1" and "0" are mapped to "-1" and "$+1$", respectively.
- The channel used in this thesis is the AWGN channel and the noise variance is denoted by σ^2.
- i and j respectively denote the indices of CNs and VNs in the window.
- $\mathcal{M}(j)$ and $\mathcal{N}(i)$ represent the neighbors of the jth VN and the neighbors of the ith CN, respectively. Neighbors of a node represent all nodes connected to that node.
- $\mathcal{M}(j)\backslash i$ denotes the neighbors of the jth VN excluding the ith CN; and $\mathcal{N}(i)\backslash j$ denotes the neighbors of the ith CN excluding the jth VN.

- $L_{APP,j}$ denotes the a posteriori log-likelihood ratio (LLR) of the jth VN.
- α_{ij} represents the LLR message sent from the ith CN to the jth VN.
- β_{ij} denotes the LLR message sent from the jth VN to the ith CN.
- q_i^t denotes the partial syndrome of row i at time t.
- We set $M_s = wm_sz$, $N_s = wn_sz$, $M_c = wm_cz$ and $N_c = wn_cz$.

When the concatenated SC-LDPC terminates, we decode all remaining source symbols and codewords in the last window. The last window contains all remaining blocks.

Algorithm 2 Joint sliding window-based belief propagation algorithm.

Set the time counter $t = 0$.

Set $\alpha_{ij} = \beta_{ij} = 0\ \forall i, j$, $L_{APP,j} = 0\ \forall j$, and $q_i^0 = 0\ \forall i$.

Initialization

if t==0 **then**

Set

$$L_{APP,j} = \begin{cases} \ln(1-p_1)/p_1,\ j = 1, 2, ..., N_s, \\ 2y_j/\sigma^2,\ j = N_s+1, ..., N_s+N_c, \end{cases} \tag{2.26}$$

where y_j denotes the jth received signal. When jth VN is punctured, $y_j = 0$.

else

Set

$$L_{APP,j} = \begin{cases} L_{APP,j+n_sz},\ j = 1, 2, ..., N_s - n_sz, \\ \ln(1-p_1)/p_1,\ j = N_s - n_sz + 1, ..., N_s, \\ L_{APP,j+n_cz},\ j = N_s+1, ..., N_s + (w-1)n_cz, \\ 2y_j/\sigma^2,\ j = N_s + (w-1)n_cz + 1, ..., N_s + N_c. \end{cases} \tag{2.27}$$

for $i = 1, 2, ..., M_s - m_sz$ **do**

Set $j \in \mathcal{N}(i)$

if $j \leq N_s - n_sz$ **then**

$$\alpha_{ij} = \alpha_{kf},\ k = i + m_sz, f = j + n_sz, \tag{2.28}$$

else if $N_s + 1 \leq j \leq N_s + N_c - n_c z$ **then**

$$\alpha_{ij} = \alpha_{kf},\ k = i + m_s z, f = j + n_c z, \tag{2.29}$$

else

Set $\alpha_{ij} = 0$.

end if

end for

for $i = M_s + 1, M_s + M_c - m_c z$ **do**

Set $j \in \mathcal{N}(i)$.

if $N_s + 1 \leq j \leq N_s + N_c - n_c z$ **then**

$$\alpha_{ij} = \alpha_{kf},\ k = i + m_c z, f = j + n_c z, \tag{2.30}$$

else

Set $\alpha_{ij} = 0$.

end if

end for

Set $\alpha_{ij} = 0$, $i = M_s - m_s z + 1, ..., M_s$ and $M_s + M_c - m_c z + 1, ..., M_s + M_c$, $j \in \mathcal{N}(i)$.

end if

Start iterating

Set the iteration counter $I = 0$.

while $I < I_{\max}$ **do**

(1). Update LLRs from VNs to CNs using

$$\beta_{ij} = L_{APP,j} - \alpha_{ij},\ j = 1, 2, \ldots, N_s + N_c \text{ and } i \in \mathcal{M}(j). \tag{2.31}$$

(2). Update LLRs from CNs to VNs

for $i = 1, \ldots, M_s + M_c,\ j \in \mathcal{N}(i)$ **do**

if $i \leq m_0 m_s z$ or $M_s < i \leq M_s + m_1 m_c z$ **then**

$$\alpha_{ij} = 2\tanh^{-1}\left((-1)^{q_i^t} \prod_{j' \in \mathcal{N}(i)\backslash j} \tanh(\beta_{ij'}/2)\right), \tag{2.32}$$

else

$$\alpha_{ij} = 2\tanh^{-1}\left(\prod_{j' \in \mathcal{N}(i)\backslash j} \tanh(\beta_{ij'}/2)\right). \tag{2.33}$$

end if

end for

(3). Calculate the posterior LLRs

for $i = 1, 2, \ldots, N_s + N_c$ **do**

$$L_{APP,j} = \begin{cases} \ln(1-p_1)/p_1 + \sum_{i' \in \mathcal{M}(j)} \alpha_{i'j},\ j \leq N_s, \\ 2y_j/\sigma^2 + \sum_{i' \in \mathcal{M}(j)} \alpha_{i'j},\ j \geq N_s + 1. \end{cases} \tag{2.34}$$

end for

Set $I = I + 1$.

end while

Output: When $j = 1, 2, ..., n_s z$ and $N_s + 1, ..., N_s + n_c z$, calculate

$$v_j^{'} = \begin{cases} 0,\ L_{APP,j} \geq 0, \\ 1,\ L_{APP,j} < 0. \end{cases} \tag{2.35}$$

$\boldsymbol{v}_{s,t}^{'} = \left[v_1^{'}, v_2^{'}, ..., v_{n_s z}^{'}\right]$ and $\boldsymbol{v}_{c,t}^{'} = \left[v_{N_s+1}^{'}, ..., v_{N_s+n_c z}^{'}\right]$ are the output source sequence and channel sequence at time t, respectively.

Calculate the partial syndrome: Set

$$\boldsymbol{q}_t^s = \boldsymbol{v}_{s,t}^{'} \begin{pmatrix} \mathbf{H}_{\mathrm{s}_0} \\ \mathbf{H}_{\mathrm{s}_1} \\ \mathbf{H}_{\mathrm{s}_2} \\ \vdots \\ \mathbf{H}_{\mathrm{s}_{m_0}} \end{pmatrix}^{\mathrm{T}}, \quad \boldsymbol{q}_t^c = \boldsymbol{v}_{c,t}^{'} \begin{pmatrix} \mathbf{H}_{\mathrm{c}_0} \\ \mathbf{H}_{\mathrm{c}_1} \\ \mathbf{H}_{\mathrm{c}_2} \\ \vdots \\ \mathbf{H}_{\mathrm{c}_{m_1}} \end{pmatrix}^{\mathrm{T}} \tag{2.36}$$

The two QC sub-block parity-check matrices above are obtained by lifting sub-block protomatrices on the $(t+1)$-th columns of $\mathbf{B}_{\mathrm{TD}}^{\mathrm{s}}$ and $\mathbf{B}_{\mathrm{TD}}^{\mathrm{c}}$, respectively.

$$q_i^t = \begin{cases} q_{i+m_s z}^{t-1} + q_t^s(i),\ 1 \le i \le m_0 m_s z, \\ q_t^s(i),\ m_0 m_s z < i \le (m_0+1) m_s z. \end{cases} \tag{2.37}$$

$$q_i^t = \begin{cases} q_{i+m_c z}^{t-1} + q_t^c(i - M_s),\ M_s + 1 \le i \le M_s + m_1 m_c z, \\ q_t^c(i - M_s),\ M_s + m_1 m_c z < i \le M_s + (m_1+1) m_c z, \end{cases} \tag{2.38}$$

where $q_t^s(i)$ and $q_t^c(i - M_s)$ denote the ith and $(i - M_s)$-th elements in $\boldsymbol{q}_t^s$ and $\boldsymbol{q}_t^c$, respectively.

Set $t = t + 1$, then go to ***Initialization***.

2.4.3 Calculation of source and channel thresholds

We can use the JP-EXIT algorithm introduced in Section 2.2.3 to calculate the channel thresholds and source thresholds based on the sub-block protomatrices in a window. We take an example to show the detailed process. We assume $w = 3$. The sub-block protomatrices in a

window shown in the blue dashed frame of Fig. 2.8 form a protomatix denoted by

$$\mathbf{B}^{w}_{\text{conc\_sc}} = \left(\begin{array}{ccc|ccc} (1\,1\,1\,1) & & & (0\,1) & & \\ (1\,1\,1\,1) & (1\,1\,1\,1) & & & (0\,1) & \\ (1\,1\,1\,1) & (1\,1\,1\,1) & (1\,1\,1\,1) & & & (0\,1) \\ \hline & & & (1\,1) & & \\ & & & (1\,1) & (1\,1) & \\ & & & (1\,1) & (1\,1) & (1\,1) \end{array}\right). \tag{2.39}$$

We can regard $\mathbf{B}^{w}_{\text{conc\_sc}}$ as a block code and use the JP-EXIT algorithm to calculate the source and channel thresholds of the concatenated SC-LDPC codes for a given w. Unlike the stopping condition of JP-EXIT algorithm for DP-LDPC codes, for concatenated SC-LDPC codes, if the mutual information (MI) between the APP-LLRs of the first n_s source VNs and first n_c channel VNs in a window and their corresponding symbols reaches "1" or the maximum number of iterations is reached, the iterations will stop.

2.5 Differential evolution method

The differential evolution (DE) algorithm is a good method to solve the non-linear optimization problem with the constraint [73]. We assume the goal function is $\min_{\boldsymbol{x}\in D} f(\boldsymbol{x})$. $\boldsymbol{x} = x_1, x_2, \ldots, x_K$ is the solution vector and K is its dimension. D is the constraint range for each entry in $\boldsymbol{x}$. Figure 2.9 shows the process of the DE algorithm.

We can use the DE method to design a code [29]. In this case, entries in a code are the solutions. $f(\boldsymbol{x})$ usually returns the channel threshold of a code consisting of $\boldsymbol{x}$. The function in the mutation operation will be slightly changed so that $x_{m_i}(i = 1, 2, ..., N)$ satisfy different constraints for different problems.

We give a toy example to show how the DE algorithm is used to design a code. We assume the structure of a code as (2.40). We want to use the DE method to search x's in **B** (2.40) to find a code with a low channel threshold. The constraints are $0 \leq x \leq 3$ and x is a non-negative

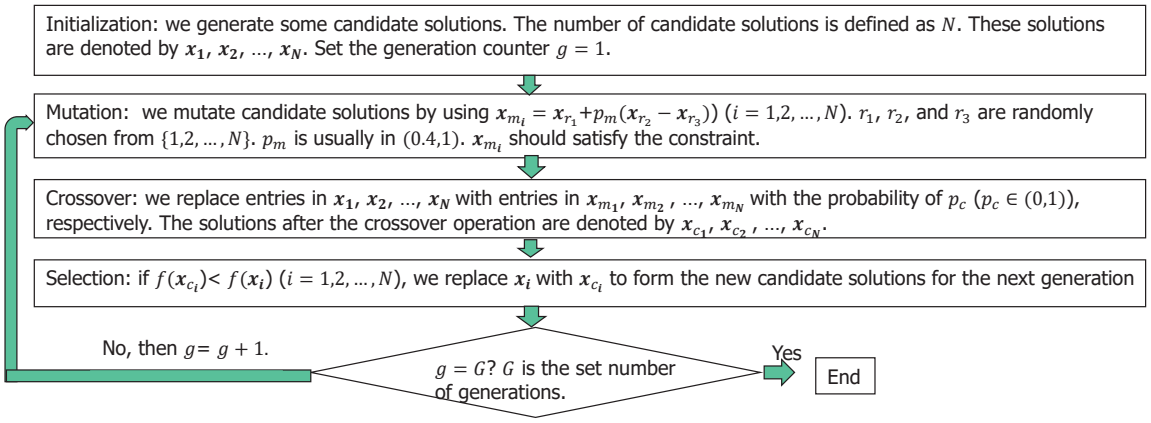


Figure 2.9: The process of the DE algorithm.

integer.

$$\mathbf{B} = \begin{pmatrix} 2 & 2 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 2 & 2 & 1 & 0 & 0 & 0 & x & 1 \\ 0 & 0 & 0 & 0 & x & x & x & x & x \\ 0 & 0 & 0 & 0 & x & x & x & x & x \\ 0 & 0 & 0 & 0 & x & x & x & x & x \end{pmatrix}. \tag{2.40}$$

- Initialization: The number of candidate codes is defined as N. Here we assume $N = 3$ for a simple explanation. We generate these three candidate codes shown as $\mathbf{B}_1$ (2.41), $\mathbf{B}_2$ (2.42), and $\mathbf{B}_3$ (2.43). The red values in these codes correspond to the values to be searched. We assume the channel thresholds of $\mathbf{B}_1$, $\mathbf{B}_2$, and $\mathbf{B}_3$ are 0.75 dB, 0.50 dB, and 0.00 dB, respectively.

$$\mathbf{B}_1 = \begin{pmatrix} 2 & 2 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 2 & 2 & 1 & 0 & 0 & 0 & 3 & 1 \\ 0 & 0 & 0 & 0 & 1 & 2 & 1 & 3 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 2 & 1 & 1 & 2 & 0 \end{pmatrix}. \tag{2.41}$$

$$\mathbf{B}_2 = \begin{pmatrix} 2 & 2 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 2 & 2 & 1 & 0 & 0 & 0 & 2 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 2 & 1 \\ 0 & 0 & 0 & 0 & 2 & 2 & 0 & 2 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 3 & 3 & 0 \end{pmatrix}. \tag{2.42}$$

$$\mathbf{B}_3 = \begin{pmatrix} 2 & 2 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 2 & 2 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 2 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 2 & 2 & 2 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 2 & 1 \end{pmatrix}. \tag{2.43}$$

- Mutation: We generate each code at this step by using

$$\mathbf{B}_{m_i} = \phi(\mathbf{B}_{m_{r_1}} + p_m(\mathbf{B}_{m_{r_2}} - \mathbf{B}_{m_{r_3}})) \ (i = 1, 2, \ldots, N), \tag{2.44}$$

where $\phi()$ outputs an integer by rounding the absolute value of the input. We assume the codes obtained after the mutaion operation are $\mathbf{B}_{m_1}$ (2.45), $\mathbf{B}_{m_2}$ (2.46), and $\mathbf{B}_{m_3}$ (2.47), respectively. When we obtain $\mathbf{B}_{m_1}$, we assume $r_1 = 3$, $r_2 = 1$, and $r_3 = 2$. When we obtain $\mathbf{B}_{m_2}$, we assume $r_1 = 3$, $r_2 = 2$, and $r_3 = 1$. When we obtain $\mathbf{B}_{m_3}$, we assume $r_1 = 2$, $r_2 = 3$, and $r_3 = 1$. Blue values in these codes are responding to values to be searched.

$$\mathbf{B}_{m_1} = \begin{pmatrix} 2 & 2 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 2 & 2 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 2 & 1 & 3 & 2 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 2 & 2 & 2 \\ 0 & 0 & 0 & 0 & 2 & 0 & 1 & 2 & 1 \end{pmatrix}. \tag{2.45}$$

$$\mathbf{B}_{m_2} = \begin{pmatrix} 2 & 2 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 2 & 2 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 2 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 2 & 2 & 3 & 2 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 3 & 1 \end{pmatrix}. \tag{2.46}$$

$$\mathbf{B}_{m_3} = \begin{pmatrix} 2 & 2 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 2 & 2 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 2 & 2 & 1 & 3 & 2 \\ 0 & 0 & 0 & 0 & 1 & 1 & 3 & 3 & 1 \end{pmatrix}. \tag{2.47}$$

- Crossover: We replace values in $\mathbf{B}_1$, $\mathbf{B}_2$, and $\mathbf{B}_3$ with those in $\mathbf{B}_{m_1}$, $\mathbf{B}_{m_2}$, and $\mathbf{B}_{m_3}$ with the probability of p_c, respectively. We assume $p_c = 0.8$ here and then obtain the codes after the crossover operation shown as $\mathbf{B}_{c_1}$ (2.48), $\mathbf{B}_{c_2}$ (2.49), and $\mathbf{B}_{c_3}$ (2.50), respectively.

$$\mathbf{B}_{c_1} = \begin{pmatrix} 2 & 2 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 2 & 2 & 1 & 0 & 0 & 0 & 3 & 1 \\ 0 & 0 & 0 & 0 & 2 & 1 & 3 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 2 & 2 \\ 0 & 0 & 0 & 0 & 2 & 0 & 1 & 2 & 1 \end{pmatrix}. \tag{2.48}$$

$$\mathbf{B}_{c_2} = \begin{pmatrix} 2 & 2 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 2 & 2 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 2 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 2 & 0 & 3 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 3 & 1 \end{pmatrix}. \tag{2.49}$$

$$\mathbf{B}_{c_3} = \begin{pmatrix} 2 & 2 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 2 & 2 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 2 & 1 & 3 & 2 \\ 0 & 0 & 0 & 0 & 1 & 0 & 3 & 2 & 1 \end{pmatrix}. \tag{2.50}$$

- Selection: we assume the channel thresholds of $\mathbf{B}_{c1}$, $\mathbf{B}_{c2}$, and $\mathbf{B}_{c3}$ are 0.85 dB, 0.65 dB, and -0.10 dB, respectively. We compare them with that the channel thresholds of $\mathbf{B}_1$, $\mathbf{B}_2$, and $\mathbf{B}_3$, i.e., 0.75 dB, 0.50 dB, and 0.00 dB, respectively. We can see that $\mathbf{B}_{c3}$ has a lower channel threshold than $\mathbf{B}_3$. So we generate the new candidate codes, i.e., $\mathbf{B}_1$, $\mathbf{B}_2$, and $\mathbf{B}_{c3}$ for the next generation, where $\mathbf{B}_3$ among the old candidate codes is replaced with $\mathbf{B}_{c3}$.

- We repeat the mutation, crossover, and selection operations until the set number of generations is reached.

2.6 Summary

This chapter provides an overview of the codes related to our research, laying the foundation for the introduction of our proposed joint source-channel coding schemes in the subsequent chapters. The codes discussed include LDPC codes, previous joint source-channel coding schemes based on DP-LDPC codes, SC-LDPC codes, and concatenated SC-LDPC codes, where a source SC-LDPC code and a channel SC-LDPC code are concatenated by using some identity matrices. This chapter also introduces the differential evolution method, which is used to design codes.

Chapter 3

New type of DP-LDPC codes

In the traditional LDPC channel encoding, linear and fast encoding can be performed when the parity portion of the parity-check matrix possesses a lower or upper triangular structure [39,40]. In this chapter, we propose a new class of DP-LDPC codes [74] which replaces the identity matrix $\mathbf{I}_{m_s}$ in the SCCV linking protomatrix $\mathbf{B}_{\text{sccv}}$ with a lower or upper triangular sub-base matrix. With the increased code design flexibility, we can design DP-LDPC codes with better channel thresholds while preserving the linear and fast source encoding property.

Figure 3.1 illustrates the structure of the new class of DP-LDPC codes (with the red connections). Compared with the traditional DP-LDPC codes shown in Fig. 2.3 presented in Section 2.2, the new class of DP-LDPC codes allows extra connections (denoted by red lines) between the CN set in the source P-LDPC code and the VN set in the channel P-LDPC. In other words, the CNs in the source P-LDPC code and VNs in the channel P-LDPC code are not linked in a one-to-one manner, but in a one-to-multiple manner. However, we set a constraint on the new connections such that the new SCCV linking protomatrix consists of a zero matrix and a lower/upper triangular base matrix with 1's on the diagonal. In other words, the new class of DP-LDPC codes can be denoted by

$$\mathbf{B}_{\text{J}} = \begin{pmatrix} \mathbf{B}_{\text{S}} & \mathbf{B}'_{\text{SCCV}} \\ \mathbf{0}_{m_c \times n_s} & \mathbf{B}_{\text{C}} \end{pmatrix}, \tag{3.1}$$

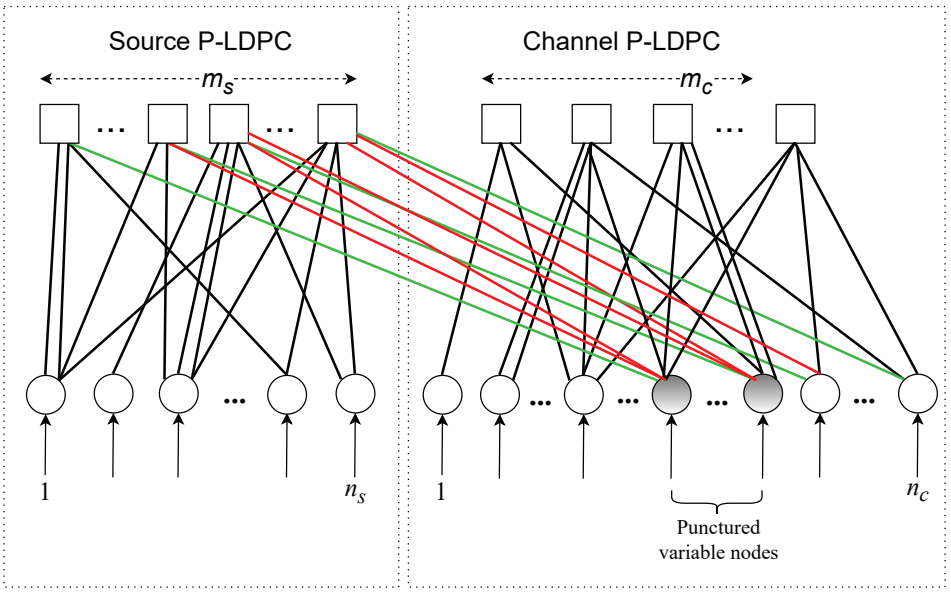


Figure 3.1: Representation of the traditional (without the red connections) and new class of (with the red connections) DP-LDPC codes.

where $\mathbf{B}'_{\text{sccv}}$ is the new SCCV linking protomatrix. Furthermore, the structure of $\mathbf{B}'_{\text{sccv}}$ can be written as

$$\mathbf{B}'_{\text{sccv}} = \begin{pmatrix} \mathbf{0}_{m_s \times m_c} & \mathbf{T}_{m_s} \end{pmatrix}, \tag{3.2}$$

where $\mathbf{T}_{m_s}$ is an upper or a lower triangular matrix with size $m_s \times m_s$ and 1's on the diagonal. For example, if $\mathbf{T}_{m_s}$ is a lower triangular matrix, it can be written as

$$\mathbf{T}_{m_s} = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ t_{2,1} & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \cdots & \ddots & 0 \\ t_{m_s,1} & t_{m_s,2} & \cdots & t_{m_s,m_s-1} & 1 \end{pmatrix}, \tag{3.3}$$

where $t_{i,j}$ $(i \in \{2,\ldots,m_s\}; j = 1,\ldots,i-1)$ are non-negative integers. We first lift the protomatrix $\mathbf{B}_{\text{J}}$ with a small lifting factor z_1 using the progressive-edge-growth (PEG) algorithm [60] to form $\mathbf{B}_{\text{J}}^{z_1} = \left(\begin{array}{c|c} \mathbf{B}_{\text{s}}^{z_1} & \mathbf{0}_{m_s z_1 \times m_c z_1}\ \mathbf{T}_{m_s}^{z_1} \\ \hline \mathbf{0}_{m_c z_1 \times n_s z_1} & \mathbf{B}_{\text{c}}^{z_1} \end{array}\right)$, $\mathbf{B}_{\text{s}}^{z_1}$ with size $m_s z_1 \times n_s z_1$ is obtained by lifting $\mathbf{B}_{\text{s}}$; $\mathbf{B}_{\text{c}}^{z_1}$ with size $m_c z_1 \times n_c z_1$ is obtained by lifting $\mathbf{B}_{\text{c}}$; and $\mathbf{T}_{m_s}^{z_1}$ with size $m_s z_1 \times m_s z_1$ is obtained by lifting $\mathbf{T}_{m_s}$. The objective of the lifting is to eliminate all entries

with values larger than 1, thereby obtaining a matrix with only 0's and 1's.

Then, we lift $\mathbf{B}_{\mathrm{J}}^{z_1}$ again with a lifting factor of z_2, forming a large quasi-cyclic low-density parity-check (QC-LDPC) matrix of appropriate size [61]. We denote the QC-LDPC matrix by

$$\mathbf{H}_{\mathrm{J}} = \left(\begin{array}{c|c} \mathbf{H}_{\mathrm{s}}^{z_1} & \mathbf{0}_{m_s z_1 z_2 \times m_c z_1 z_2} \; \mathbf{H}_{\mathrm{T}_{m_s}}^{z_1} \\ \hline \mathbf{0}_{m_c z_1 z_2 \times n_s z_1 z_2} & \mathbf{H}_{\mathrm{c}}^{z_1} \end{array} \right). \tag{3.4}$$

Each entry h in $\mathbf{H}_{\mathrm{J}}$ represents a circulant permutation matrix (CPM) with size $z_2 \times z_2$ obtained by cyclically right-shifting the identity matrix $\mathbf{I}_{z_2}$ by h columns. $\mathbf{H}_{\mathrm{s}}^{z_1}$, $\mathbf{H}_{\mathrm{c}}^{z_1}$, and $\mathbf{H}_{\mathrm{T}_{m_s}}^{z_1}$ are QC parity-check matrices obtained by lifting $\mathbf{B}_{\mathrm{s}}^{z_1}$, $\mathbf{B}_{\mathrm{c}}^{z_1}$, and $\mathbf{T}_{m_s}^{z_1}$, respectively. Note also that during the lifting process, we try to maximize the girth (shortest cycle) of the resultant QC-LDPC matrix.

3.1 Source encoding

We consider a binary independent and identically distributed (i.i.d.) source that follows a Bernoulli distribution, with the probability of "1" given by p_1, the probability of "0" given by $1 - p_1$. Given a sample source sequence $\boldsymbol{s}$ of length $1 \times N_s = 1 \times n_s z_1 z_2$, we divide it into $n_s z_1$ sub-sequences $\boldsymbol{s}_i$ $(i = 1, 2, \ldots, n_s z_1)$ each of length z_2. Thus we can rewrite $\boldsymbol{s}$ as $\boldsymbol{s} = (\boldsymbol{s}_1 \;\; \boldsymbol{s}_2 \;\; \cdots \;\; \boldsymbol{s}_{n_s z_1})$. Then, we generate the compressed source sequence $\boldsymbol{c}$ of length $1 \times M_s = 1 \times m_s z_1 z_2$ using $\mathbf{H}_{\mathrm{s}}^{z_1}$and $\mathbf{H}_{\mathrm{T}_{m_s}}^{z_1}$ in (3.4). We divide $\boldsymbol{c}$ into $m_s z_1$ groups each of length z_2 and represent $\boldsymbol{c}$ as $\boldsymbol{c} = (\boldsymbol{c}_1 \;\; \boldsymbol{c}_2 \;\; \cdots \;\; \boldsymbol{c}_{m_s z_1})$. Considering the first block row of $\mathbf{H}_{\mathrm{J}}$ shown in (3.4), we have $\sum_{i=1}^{n_s z_1} \boldsymbol{s}_i (\mathbf{I}^{h_s^{(1,i)}})^{\mathrm{T}} + \boldsymbol{c}_1 = \mathbf{0} \Rightarrow \boldsymbol{c}_1 = \sum_{i=1}^{n_s z_1} \boldsymbol{s}_i (\mathbf{I}^{h_s^{(1,i)}})^{\mathrm{T}}$, where $\mathbf{I}^{h_s^{(j,i)}}$ $(j = 1, 2, ..., m_s z_1; i = 1, 2, ..., n_s z_1)$ denotes the (j, i)-th circulant permutation matrix in $\mathbf{H}_s^{z_1}$. Then,

$$\boldsymbol{c}_j = \sum_{i=1}^{n_s z_1} \boldsymbol{s}_i (\mathbf{I}^{h_s^{(j,i)}})^{\mathrm{T}} + \sum_{k=1}^{j-1} \boldsymbol{c}_k (\mathbf{I}^{h_t^{(j,k)}})^{\mathrm{T}} \;, \tag{3.5}$$

where $\mathbf{I}^{h_t^{(j,k)}}$ denotes the (j, k)-th circulant permutation matrix in $\mathbf{H}_{\mathrm{T}_{m_s}}^{z_1}$. Afterwards, the compressed source symbols $\boldsymbol{c}$ is passed to the original channel coding to evaluate the parity-check

bits based on $\mathbf{H}_{\mathrm{c}}^{z_1}$.

3.2 Complexity and latency analyses

3.2.1 Source encoder

In the traditional DP-LDPC joint source-channel coding system shown in Fig. 2.3 presented in Section 2.2, $\boldsymbol{c}_j$ $(j = 1, 2, 3, ..., m_s z_1)$ can be computed in full parallel using

$$\boldsymbol{c}_j = \sum_{i=1}^{n_s z_1} \boldsymbol{s}_i (\mathbf{I}^{h_s^{(j,i)}})^T, \quad j = 1, 2, ..., m_s z_1 \ . \tag{3.6}$$

In a practical environment, however, $\boldsymbol{c}$ would not be derived in full parallel, because it would require a lot of hardware resources. A more feasible implementation of the hardware source encoder is to compute $\boldsymbol{c}_j$ sequentially, i.e., $\boldsymbol{c}_1$, $\boldsymbol{c}_2$, ..., $\boldsymbol{c}_{m_s z_1}$. For the proposed new DP-LDPC codes, $\boldsymbol{c}_j$ can be computed after all $\boldsymbol{c}_{j'}$ where $j' < j$ have been evaluated based on (3.5). Since only one $\boldsymbol{c}_j$ $(j = 1, 2, \ldots, m_s z_1)$ needs to be derived each time, the hardware resources for source encoding in the proposed DP-LDPC-based joint source-channel coding system will be similar to that of the original DP-LDPC-based joint source-channel coding system. Moreover, both (3.5) and (3.6) above can be completed with simple shift registers and Xor gates. Thus, the source encoding complexity of both systems are considered as low though the proposed system could be relatively more complex.

For these two DP-LDPC-based joint source-channel coding schemes, we can compute the z_2 values in each sequence $\boldsymbol{c}_j$ $(j = 1, 2, 3, ..., m_s z_1)$ in parallel. We can construct a balanced binary tree to implement the binary addition of multiple input symbols. The balanced binary tree structure with 7 inputs is shown in Fig. 3.2, where each circle denotes a binary addition (i.e., Xor gate). The number of levels is 3 and there are 6 binary addition operations. For a balanced binary tree structure with x inputs, $x - 1$ addition operations are needed to generate an output and the number of levels is given by $\lceil \log_2 x \rceil$. Thus the latency is given by $\lceil \log_2 x \rceil \tau_1$, where τ_1 denotes the time taken by a binary addition operation (Xor gate).

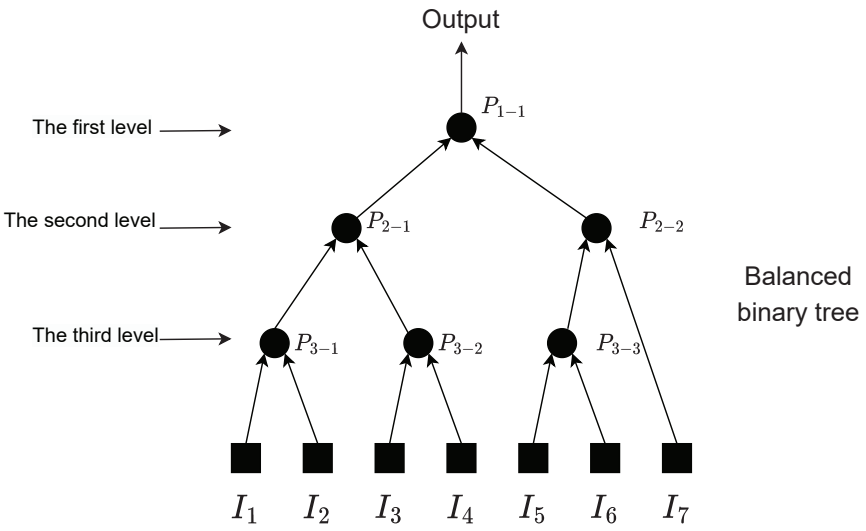


Figure 3.2: A balanced binary tree structure for the binary addition operations with 7 inputs. $I_1, I_2, ..., I_7$ are the input symbols. Black circles denote the binary addition operations. P_{i-j} denotes the output value of the jth binary addition operation at the ith level. This structure comes from the binary tree proposed in [75] and has been authorized by the author Francis C. M. Lau of this reference.

For the traditional DP-LDPC joint source-channel coding scheme, each of the z_2 values in the sequence $\boldsymbol{c}_j$ $(j = 1, 2, 3, ..., m_s z_1)$ is generated by the binary addition of $w_{\rm s}^j$ source symbols, where $w_{\rm s}^j$ denotes the jth row weight of $\mathbf{B}_{\rm s}^{z_1}$. In other words, for the balanced binary tree structure, there are $w_{\rm s}^j$ (= $\lceil j/z_1 \rceil$-th row weight of $(\mathbf{B}_{\rm s}\ \mathbf{B}_{\rm sccv})$ minus 1) inputs.

For the new DP-LDPC joint source-channel coding scheme, the values in the sequence $\boldsymbol{c}_j$ $(j = 1, 2, 3, ..., m_s z_1)$ are generated by the binary addition of $w_{\rm s}^j$ source symbols and $w_{\rm t}^j - 1$ compressed symbols, where $w_{\rm t}^j$ denotes the jth row weight of $\mathbf{T}_{m_s}^{z_1}$. In this case, for the balanced binary tree structure, there are $w_{\rm s}^j + w_{\rm t}^j - 1$ (= $\lceil j/z_1 \rceil$-th row weight of $(\mathbf{B}_{\rm s}\ \mathbf{B}'_{\rm sccv})$ minus 1) inputs.

In general, the number of Xor gates required in the source encoding process is determined by the largest possible number of input symbols, which is proportional to the largest row weight of $(\mathbf{B}_{\rm s}\ \mathbf{B}_{\rm sccv})$ (for the traditional DP-LDPC code) or $(\mathbf{B}_{\rm s}\ \mathbf{B}'_{\rm sccv})$ (for the proposed DP-LDPC code) minus 1. Specifically, the number of Xor gates required in the source encoding process equals the largest row weight of $(\mathbf{B}_{\rm s}\ \mathbf{B}_{\rm sccv})$ or $(\mathbf{B}_{\rm s}\ \mathbf{B}'_{\rm sccv})$ minus 2. Moreover, the percentage increase in source encoding latency is given by

$$\Delta_{\rm latency} = \frac{\sum_j \lceil \log_2(w_{\rm s}^j + w_{\rm t}^j - 1) \rceil - \sum_j \lceil \log_2 w_{\rm s}^j \rceil}{\sum_j \lceil \log_2 w_{\rm s}^j \rceil}. \tag{3.7}$$

Even though the percentage increases in source encoding complexity and latency may seem large, the increase in the combined encoder complexity and latency (when considering both source encoding and channel encoding) is much lower.

3.2.2 Decoder

The decoding complexity of the proposed DP-LDPC codes can be analyzed as follows.

As shown in Section 2.2.2, we use a joint belief propagation (BP) algorithm to decode the DP-LDPC code as one single code (i.e., $\mathbf{B}_{\mathrm{J}_0}$ or $\mathbf{B}_{\mathrm{J}}$ after lifting is considered), and to update the check-to-variable (C2V) messages and variable-to-check (V2C) messages iteratively. Since the computation of C2V messages is much more complex than that of V2C messages, we approximate the complexity of the decoder by the complexity of the check-node processors (CNPs). We further assume using the symmetric binary tree structure in [75] to compute C2V messages during the hardware implementation. Figure 3.3 reviews the structure when a check node (CN) has a degree of 14. For this CN, there are 14 V2C inputs and 14 C2V outputs. We use the look-up table (LUT) method to implement the "$\tanh$" function. A LUT has two inputs and generates an output. Firstly, we construct a balanced binary tree based on the 14 inputs. The number of levels equals $\lceil \log_2 14 \rceil = 4$. At the ith ($i = 1, 2, 3$) level, each node denotes a LUT. The two inputs of a LUT are the children of the node. At the lowest level (i.e., 4-th level), each node denotes a V2C input (i.e., represented by I_1 to I_{14}). In Fig. 3.3, P_{i-j} denotes the jth output at the ith level.

When we calculate the message passed from the CN to the first VN (i.e., O_1), $I_2, I_3, ..., I_{14}$ need to be used. Based on the binary tree structure, we need to use the outputs from $P_{4-2}, P_{3-2}, P_{2-2}$ and P_{1-2}, to complete the computation. We first use P_{2-2} and P_{1-2} to calculate P'_{2-1}. Then we use P'_{2-1} and P_{3-2} to calculate P'_{3-1}. Finally, we use P'_{3-1} and P_{4-2} to calculate O_1. As shown in Fig. 3.3, all LUTs and V2C input related to the calculation for O_1 are indicated by dashed circles. In order to reuse LUTs as much as possible, we can use P'_{3-1} and P_{4-1} to calculate O_2 (i.e., the message passed from the CN to the second VN). In fact, another tree is constructed by mirroring the balance binary tree from the second level downwards. The nodes in this mirror

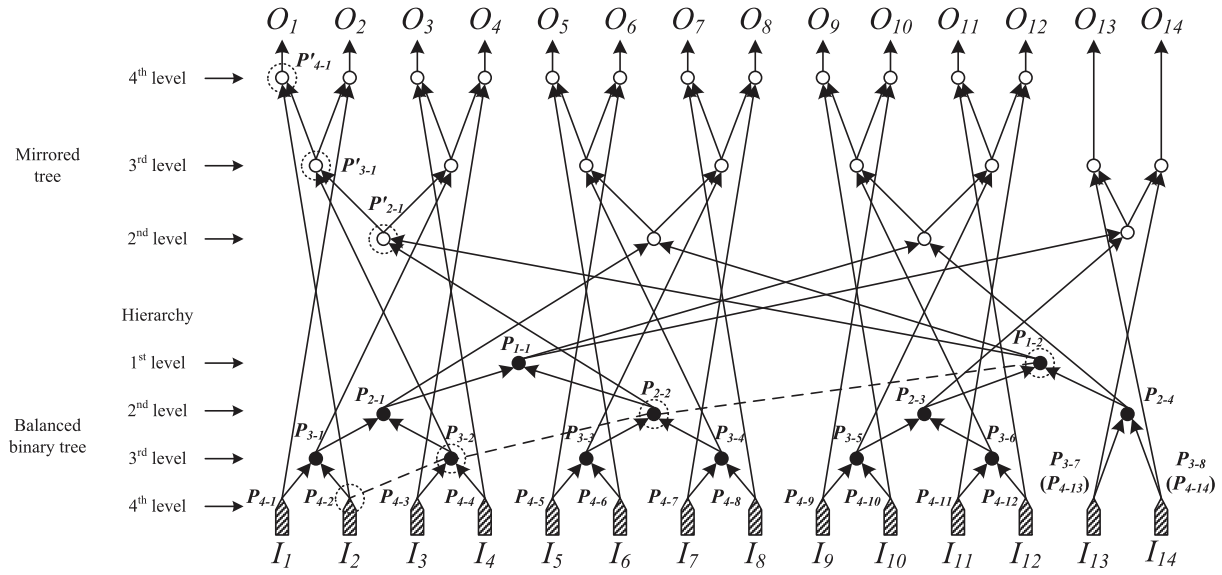


Figure 3.3: A symmetric binary tree structure of a C2V update in parallel. The check node has a degree of 14 [75]. $I_1, I_2, ..., I_{14}$ are V2C inputs and $O_1, O_2, ..., O_{14}$ are C2V outputs. Black and blank circles denote the look-up tables (LUTs). P_{i-j} denotes the jth output at the ith level. Dashed circles and lines denote the outputs and valid connections related to the calculation of O_1, respectively. This figure comes from [75] and has been authorized by the author Francis C. M. Lau of this reference.

tree shown in Fig. 3.3 are denoted by blank circles. At the second level of the mirror tree, the two inputs of each node consist of (i) the sibling of its symmetric node and (ii) the sibling of the parent of its symmetric node. At the $i > 2$ level of the mirror tree, the two inputs of each node consist of (i) its parent node and (ii) the sibling of its symmetric node.

We can see that the latency of a C2V update with the tree structure shown in Fig. 3.3 is related to the number of levels of the tree. When there are x inputs, the number of levels of a balanced binary tree is $\lceil \log_2 x \rceil$. As seen from the figure, the latency of a C2V update is therefore $2(\lceil \log_2 x \rceil - 1)\tau_2$, where τ_2 denotes the time of querying a LUT. Therefore, compared with the traditional DP-LDPC codes, the percentage increase in the decoding latency of the new DP-LDPC codes is given by

$$\Delta_{\text{latency,dec}} = \frac{\sum_j 2(\lceil \log_2 w_{\text{J}}^j \rceil - 1) - \sum_j 2(\lceil \log_2 w_{\text{J}_0}^j \rceil - 1)}{\sum_j 2(\lceil \log_2 w_{\text{J}_0}^j \rceil - 1)} = \frac{\sum_j (\lceil \log_2 w_{\text{J}}^j \rceil - \lceil \log_2 w_{\text{J}_0}^j \rceil)}{\sum_j \lceil \log_2 w_{\text{J}_0}^j \rceil - 1}, \tag{3.8}$$

where w_{J}^j and $w_{\text{J}_0}^j$ denote the jth row weight of $\mathbf{B}_{\text{J}}$ and $\mathbf{B}_{\text{J}_0}$, respectively. Similar to the analysis on the source encoder complexity, the complexity of the check-node-processor depends on the

Table 3.1: The channel thresholds $(E_s/N_0)_{\text{th}}$ (dB) of $\mathbf{B}^{0.04}$ in (3.9) for different x_1 and x_2 values. The Shannon limit is -7.0 dB.

x_1	0	0	0	0	1	2	3
x_2	1	2	3	0	0	0	0
$(E_s/N_0)_{\text{th}}$ (dB)	$\mathbf{-5.267}$	-5.204	-5.049	-5.127	-4.819	-4.526	-4.273

CN with the highest degree. For 14 V2C inputs shown in Fig. 3.3, we need $3(14-2)=36$ LUTs to implement a C2V update. When the highest row degree is x, there are $(x-2)$ LUTs in a balanced binary tree and $x+(x-4)$ LUTs in its mirrored binary tree. So, we need a total of $3(x-2)$ LUTs.

3.3 Results and discussions

3.3.1 Optimized codes and their error performance

Example #1: We consider the DP-LDPC code in [37, Eq. (16)] which is designed at $p_1=0.04$. We denote it by $\mathbf{B}_{\text{J}}^{0.04}=\mathbf{B}^{0.04}(x_1=x_2=0)$, where $\mathbf{B}^{0.04}$ is given as

$$\mathbf{B}^{0.04}=\left(\begin{array}{cccc|cccc|c} 2 & 2 & 1 & 1 & 0 & 0 & 0 & 1 & x_2 \\ 1 & 1 & 2 & 1 & 0 & 0 & 0 & x_1 & 1 \\ \hline 0 & 0 & 0 & 0 & 1 & 0 & 1 & 2 & 2 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 2 \end{array}\right). \tag{3.9}$$

For $\mathbf{B}_{\text{J}}^{0.04}$, $m_s=2, n_s=4, m_c=3, n_c=5$ and the last VN in the channel protograph is punctured. The overall symbol code rate of this DP-LDPC code is evaluated by $R=R_sR_c=1$, where $R_s=n_s/m_s$ is the source compression rate, $R_c=m_s/(n_c-n_p)$ denotes the channel code rate, and n_p denotes the number of punctured VNs in the channel protograph. By using the JP-EXIT algorithm [33,35,37] introduced in Section 2.2.3, the channel threshold of $\mathbf{B}_{\text{J}}^{0.04}$ is found to be -5.127 dB.

In this chapter, we assume the maximum entry value of a code is 3 to limit the search-

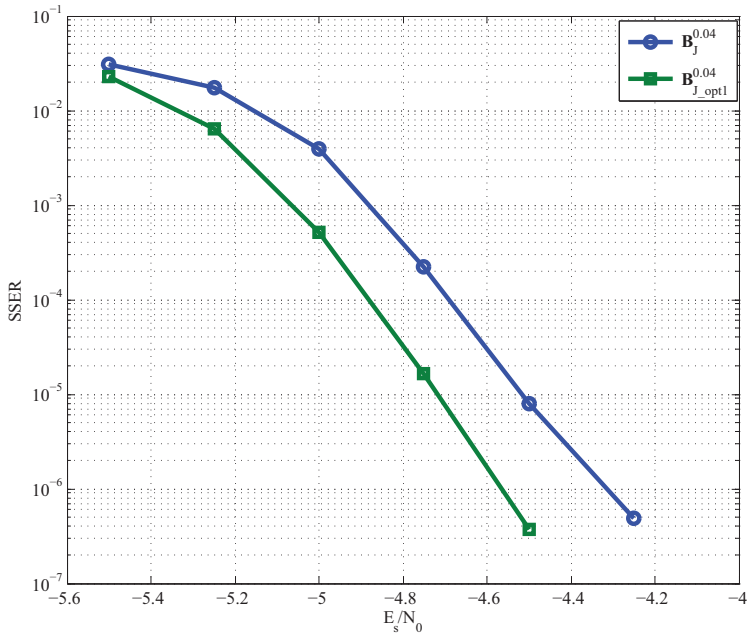


Figure 3.4: SSER performance comparison for $\mathbf{B}_{\mathrm{J}}^{0.04}$ and $\mathbf{B}_{\mathrm{J\_opt1}}^{0.04}$ when $R = 1$, $p_1 = 0.04$, and $N_s = z_1 z_2 n_s = 4 \cdot 800 \cdot 4 = 12800$. The Shannon limit is -7.00 dB.

ing space. To construct a new DP-LDPC code based on $\mathbf{B}_{\mathrm{J}}^{0.04}$, we consider all possible lower and upper triangular structures of $\begin{pmatrix} 1 & x_2 \\ x_1 & 1 \end{pmatrix}$, and evaluate the channel thresholds of the corresponding codes. We can see from Table 3.1 that when $x_1 = 0$ and $x_2 = 1$, $\mathbf{B}^{0.04}$ has the lowest channel threshold. We denote this optimized code by $\mathbf{B}_{\mathrm{J\_opt1}}^{0.04} = \mathbf{B}^{0.04}(x_1 = 0, x_2 = 1)$. The channel threshold of $\mathbf{B}_{\mathrm{J\_opt1}}^{0.04}$ is found to be -5.267 dB, which is 0.140 dB lower than the threshold of $\mathbf{B}_{\mathrm{J}}^{0.04}$. We also run simulations and record the source symbol error rate (SSER) of the code. For all the simulation results in this chapter, we set the maximum number of decoding iterations to $I_{\max} = 200$. The simulation will be terminated if (i) more than 2×10^5 frames have been simulated or (ii) more than 100 error frames have been found and no less than 5000 frames have been simulated. Figure 3.4 shows the simulated results for $\mathbf{B}_{\mathrm{J}}^{0.04}$ and $\mathbf{B}_{\mathrm{J\_opt1}}^{0.04}$. We can see that at an SSER of 10^{-6}, $\mathbf{B}_{\mathrm{J\_opt1}}^{0.04}$ outperforms $\mathbf{B}_{\mathrm{J}}^{0.04}$ by about 0.25 dB.

Example #2: [1] We consider two DP-LDPCs $\mathbf{B}_{\mathrm{J3}}^{0.01}$ and $\mathbf{B}_{\mathrm{J4}}^{0.01}$ in [35, Table I] which are designed for $p_1 = 0.01$ and $R = 2$. For $\mathbf{B}_{\mathrm{J3}}^{0.01}$, the source base matrix $\mathbf{B}_{\mathrm{J3-s}}^{0.01}$ and channel base matrix

[1] We swap the fourth and fifth columns of the SCCV linking matrices of $\mathbf{B}_{\mathrm{J3}}^{0.01}$ and $\mathbf{B}_{\mathrm{J4}}^{0.01}$ in [35, Table I] to obtain an identity matrix. Correspondingly, the fourth and fifth columns of the channel base matrices of these two codes are also swapped.

Table 3.2: The channel thresholds $(E_s/N_0)_{\text{th}}$ of DP-LDPCs. $R = 2$ and $p_1 = 0.01$. The Shannon limit is -12.02 dB.

	$\mathbf{B}_{\text{J3}}^{0.01}$	$\mathbf{B}_{\text{J3\_opt1}}^{0.01}$	$\mathbf{B}_{\text{J3\_opt2}}^{0.01}$	$\mathbf{B}_{\text{J3\_opt3}}^{0.01}$	$\mathbf{B}_{\text{J4}}^{0.01}$	$\mathbf{B}_{\text{J4\_opt1}}^{0.01}$	$\mathbf{B}_{\text{J4\_opt2}}^{0.01}$	$\mathbf{B}_{\text{J4\_opt3}}^{0.01}$
$(E_s/N_0)_{\text{th}}$ (dB)	-9.324	-9.555	-9.680	-9.734	-9.390	-9.616	-9.722	-9.744

$\mathbf{B}_{\text{J3}-\text{c}}^{0.01}$ are given by

$$\begin{cases} \mathbf{B}_{\text{J3}-\text{s}}^{0.01} = \begin{pmatrix} 1 & 1 & 2 & 1 & 3 & 1 & 3 & 1 \\ 1 & 2 & 1 & 2 & 1 & 2 & 1 & 2 \end{pmatrix}; \\ \mathbf{B}_{\text{J3}-\text{c}}^{0.01} = \begin{pmatrix} 1 & 0 & 0 & 3 & 0 \\ 0 & 1 & 1 & 1 & 2 \\ 0 & 1 & 1 & 2 & 1 \end{pmatrix}. \end{cases} \tag{3.10}$$

For $\mathbf{B}_{\text{J4}}^{0.01}$, the source base matrix $\mathbf{B}_{\text{J4}-\text{s}}^{0.01}$ and channel base matrix $\mathbf{B}_{\text{J4}-\text{c}}^{0.01}$ are given by

$$\begin{cases} \mathbf{B}_{\text{J4}-\text{s}}^{0.01} = \begin{pmatrix} 2 & 1 & 2 & 1 & 3 & 1 & 3 & 1 \\ 1 & 2 & 1 & 2 & 1 & 3 & 1 & 3 \end{pmatrix}; \\ \mathbf{B}_{\text{J4}-\text{c}}^{0.01} = \begin{pmatrix} 1 & 0 & 0 & 3 & 0 \\ 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 2 & 1 \end{pmatrix}. \end{cases} \tag{3.11}$$

For both $\mathbf{B}_{\text{J3}}^{0.01}$ and $\mathbf{B}_{\text{J4}}^{0.01}$, $m_s = 2, n_s = 8, m_c = 3, n_c = 5$ and $n_p = 1$. The punctured VN corresponds to the fourth VN in the channel protograph. Their SCCV linking base matrices are both $\mathbf{B}_{\text{J3}-\text{sccv}}^{0.01} = \mathbf{B}_{\text{J4}-\text{sccv}}^{0.01} = \mathbf{B}_{\text{sccv}}^{0.01}(x_1 = x_2 = 0)$, where

$$\mathbf{B}_{\text{sccv}}^{0.01} = \begin{pmatrix} 0 & 0 & 0 & 1 & x_2 \\ 0 & 0 & 0 & x_1 & 1 \end{pmatrix}. \tag{3.12}$$

Using the same method as in Example #1, we find three cases, i.e., $\mathbf{B}_{\text{sccv}}^{0.01}(x_1 = 1, x_2 = 0)$, $\mathbf{B}_{\text{sccv}}^{0.01}(x_1 = 2, x_2 = 0)$, and $\mathbf{B}_{\text{sccv}}^{0.01}(x_1 = 3, x_2 = 0)$, for which the constructed codes have lower channel thresholds than the original ones. We further denote the corresponding new DP-LDPC

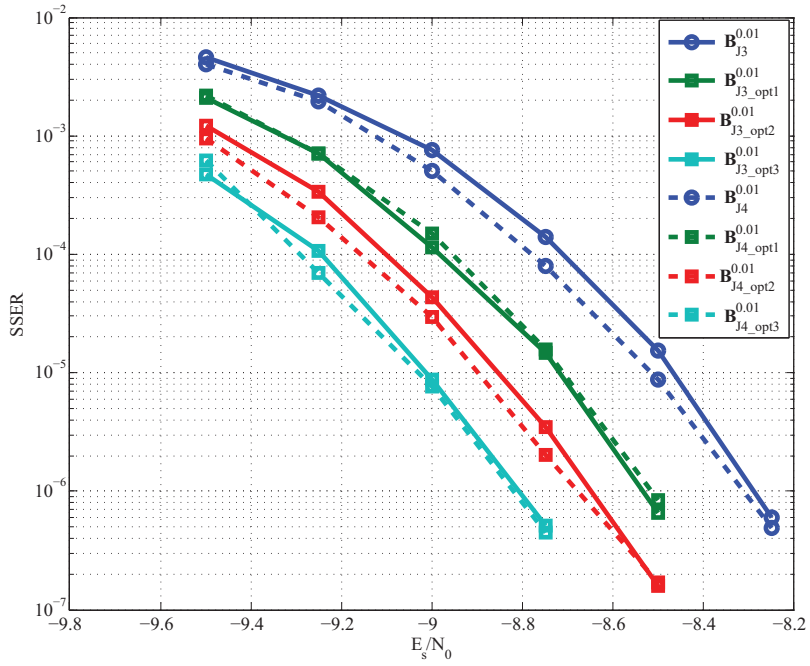


Figure 3.5: SSER performance of $\mathbf{B}_{\text{J3}}^{0.01}$ (solid line) and $\mathbf{B}_{\text{J4}}^{0.01}$ (dashed line) and their corresponding new DP-LDPC codes. $R = 2$, $p_1 = 0.01$, and $N_s = z_1 z_2 n_s = 4 \cdot 400 \cdot 8 = 12800$. The Shannon limit is -12.02 dB.

codes by $\mathbf{B}_{\text{J3(J4)\_opt1}}^{0.01}$, $\mathbf{B}_{\text{J3(J4)\_opt2}}^{0.01}$, and $\mathbf{B}_{\text{J3(J4)\_opt3}}^{0.01}$, respectively. In Table 3.2, we can see that the new class of DP-LDPC codes has better thresholds compared with the original ones. For example, the channel threshold of $\mathbf{B}_{\text{J3\_opt3}}^{0.01}$ is 0.41 dB lower than that of $\mathbf{B}_{\text{J3}}^{0.01}$; and the channel threshold of $\mathbf{B}_{\text{J4\_opt3}}^{0.01}$ is 0.354 dB lower than that of $\mathbf{B}_{\text{J4}}^{0.01}$.

The simulated SSERs in Fig. 3.5 show that the new class of DP-LDPC codes outperforms the original DP-LDPC codes. The SSER results are also consistent with the channel threshold results shown in Table 3.2. For example, in terms of SSER, $\mathbf{B}_{\text{J3\_opt3}}^{0.01}$ outperforms $\mathbf{B}_{\text{J3\_opt2}}^{0.01}$, which outperforms $\mathbf{B}_{\text{J3\_opt1}}^{0.01}$, which in turn outperforms $\mathbf{B}_{\text{J3}}^{0.01}$. When comparing the channel thresholds, $\mathbf{B}_{\text{J3\_opt3}}^{0.01} < \mathbf{B}_{\text{J3\_opt2}}^{0.01} < \mathbf{B}_{\text{J3\_opt1}}^{0.01} < \mathbf{B}_{\text{J3}}^{0.01}$. In particular, $\mathbf{B}_{\text{J3\_opt3}}^{0.01}$ and $\mathbf{B}_{\text{J4\_opt3}}^{0.01}$ have around 0.5 dB coding gains over $\mathbf{B}_{\text{J3}}^{0.01}$ and $\mathbf{B}_{\text{J4}}^{0.01}$, respectively, at an SSER of 10^{-6}.

Example #3: We design a traditional DP-LDPC code based on $p_1 = 0.14$ and then optimize the code using our proposed technique. We reduce the source compression rate to adapt to the increased p_1. We assume a source compression rate of $R_s = 5/4$ and an overall symbol rate $R = 1$. Thus, the channel code rate is $R_c = R/R_s = 4/5$. We set m_s and n_s to 4 and 5, respectively. For a good channel code, m_c should have a minimum value of 3. Here, we set

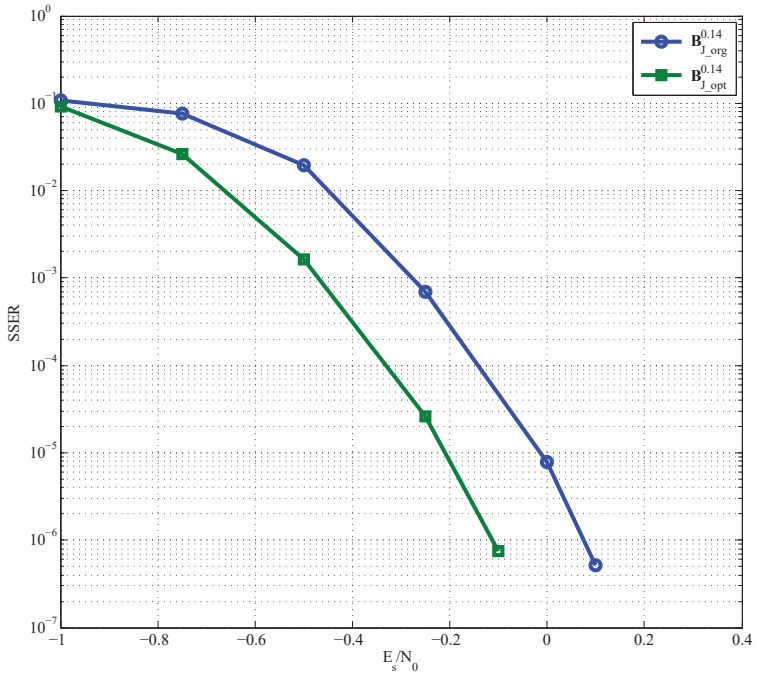


Figure 3.6: SSER performance of $\mathbf{B}^{0.14}_{\text{J\_org}}$ and $\mathbf{B}^{0.14}_{\text{J\_opt}}$. $R = 1$, $p_1 = 0.14$, and $N_s = z_1 z_2 n_s = 4 \cdot 400 \cdot 5 = 8000$. The Shannon limit is -2.05 dB.

$m_c = 3$. To match the required channel code rate, we set $n_c = 7$ and $n_p = 2$, i.e., there are 2 punctured VNs in the channel code. Using the differential evolution (DE) method introduced in Section 2.5, we obtain a traditional DP-LDPC code $\mathbf{B}^{0.14}_{\text{J\_org}} = \mathbf{B}^{0.14}(x_i = 0\ \forall i)$, where

$$\mathbf{B}^{0.14} = \left(\begin{array}{ccccc|ccccccc} 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & x_1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & x_2 & x_4 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & x_3 & x_5 & x_6 & 1 \\ \hline 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 2 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 3 & 1 & 2 & 0 & 1 \end{array}\right). \tag{3.13}$$

Two VNs with the highest degrees are punctured, i.e., the $(n_s + 4)$-th VN and the $(n_s + 7)$-th VN in $\mathbf{B}^{0.14}_{\text{J\_org}}$. The channel threshold of $\mathbf{B}^{0.14}_{\text{J\_org}}$ is -0.653 dB while the Shannon limit is -2.05 dB at $p_1 = 0.14$.

Next, we optimize $\mathbf{B}^{0.14}_{\text{J\_org}}$ by replacing the identity matrix in the SCCV linking matrix with a

lower or upper triangular base matrix. Here, we only consider the lower triangular base matrix to reduce the searching complexity. Using the same DE method mentioned above, we obtain a new DP-LDPC code $\mathbf{B}_{\text{J\_opt}}^{0.14} = \mathbf{B}^{0.14}(x_1 = x_3 = 1, x_2 = x_4 = x_5 = x_6 = 0)$. Its channel threshold is -0.840 dB, which is 0.187 dB lower than that of $\mathbf{B}_{\text{J\_org}}^{0.14}$. In Fig. 3.6, we can see that $\mathbf{B}_{\text{J\_opt}}^{0.14}$ outperforms $\mathbf{B}_{\text{J\_org}}^{0.14}$ by about 0.2 dB at an SSER of 10^{-6}.

3.3.2 Complexity and latency:

The following are the complexity and latency comparisons between new DP-LDPC codes and traditional DP-LDPC codes based on Example #1 to Example #3.

In Example #1, $\mathbf{B}_{\text{J}}^{0.04}$ is modified to $\mathbf{B}_{\text{J\_opt1}}^{0.04}$. The largest row weight of $(\mathbf{B}_{\text{s}}\ \mathbf{B}_{\text{sccv}})$ is 7 while that of $(\mathbf{B}_{\text{s}}\ \mathbf{B}'_{\text{sccv}})$ is 8. The percentage increase in the number of Xor gates used in source encoding therefore equals $\Delta_{\text{xor}} = [(8-2)-(7-2)]/(7-2) = 20\%$. However, there is no change in the source encoding latency because $(\lceil \log_2(8-1) \rceil - \lceil \log_2(7-1) \rceil) = 0$. The largest row weight of $\mathbf{B}_{\text{J}}^{0.04}$ is 7 while that of $\mathbf{B}_{\text{J\_opt1}}^{0.04}$ is 8. The percentage increase in the number of LUTs used in C2V updates equals $(3 \cdot (8-2) - 3 \cdot (7-2))/(3 \cdot (7-2)) = 20\%$. However, there is no change in the decoding latency because $(\lceil \log_2 8 \rceil - \lceil \log_2 7 \rceil) = 0$.

In Example #2, we consider the case when $\mathbf{B}_{\text{J4}}^{0.01}$ is modified to $\mathbf{B}_{\text{J4\_opt3}}^{0.01}$. The largest row weight of $(\mathbf{B}_{\text{s}}\ \mathbf{B}_{\text{sccv}})$ is 15 while that of $(\mathbf{B}_{\text{s}}\ \mathbf{B}'_{\text{sccv}})$ is 18. The percentage increase in the number of Xor gates used in source encoding therefore equals $\Delta_{\text{xor}} = [(18-2)-(15-2)]/(15-2) = 23.1\%$. Moreover, the percentage increase in the source encoding latency equals

$$\Delta_{\text{latency}} = \frac{(\lceil \log_2(18-1) \rceil + \lceil \log_2(15-1) \rceil) - (\lceil \log_2(15-1) \rceil + \lceil \log_2(15-1) \rceil)}{\lceil \log_2(15-1) \rceil + \lceil \log_2(15-1) \rceil} = 12.5\%. \tag{3.14}$$

The largest row weight of $\mathbf{B}_{\text{J4}}^{0.01}$ is 15 while that of $\mathbf{B}_{\text{J4\_opt3}}^{0.01}$ is 18. The percentage increase in the number of LUTs used in C2V updates equals $(3*(18-2)-3*(15-2))/(3*(15-2)) = 23.1\%$. Moreover, the percentage increase in the decoding latency equals

$$\Delta_{\text{latency,dec}} = \frac{(\lceil \log_2(18) \rceil + \lceil \log_2(15) \rceil) - (2 \cdot \lceil \log_2(15) \rceil)}{2 \cdot (\lceil \log_2(15) \rceil - 1) + 2 \cdot (\lceil \log_2(4) \rceil - 1) \rceil + \lceil \log_2(5) \rceil - 1} = 10\%. \tag{3.15}$$

In Example #3, we consider the case when $\mathbf{B}_{\text{J\_org}}^{0.14}$ is modified to $\mathbf{B}_{\text{J\_opt}}^{0.14}$. The largest row weight of $(\mathbf{B}_{\text{s}}\ \mathbf{B}_{\text{sccv}})$ is 5 while that of $(\mathbf{B}_{\text{s}}\ \mathbf{B}'_{\text{sccv}})$ is 6. The percentage increase in the number of Xor gates used in source encoding therefore equals $\Delta_{\text{xor}} = [(6-2)-(5-2)]/(5-2) = 33.3\%$. Moreover, the percentage increase in the source encoding latency equals

$$\Delta_{\text{latency}} = \frac{(\lceil\log_2(3-1)\rceil + \lceil\log_2(6-1)\rceil + \lceil\log_2(5-1)\rceil + \lceil\log_2(4-1)\rceil) - (2\cdot\lceil\log_2(3-1)\rceil + 2\cdot\lceil\log_2(5-1)\rceil)}{2\cdot\lceil\log_2(3-1)\rceil + 2\cdot\lceil\log_2(5-1)\rceil}$$
$$= 33.3\%. \tag{3.16}$$

The largest row weights of $\mathbf{B}_{\text{J\_org}}^{0.14}$ and $\mathbf{B}_{\text{J\_opt}}^{0.14}$ are both 8. Therefore, there is no change in the number of LUTs used in C2V updates. Moreover, the percentage increase in the decoding latency equals

$$\frac{(\lceil\log_2(3)\rceil + \lceil\log_2(6)\rceil + \lceil\log_2(5)\rceil + \lceil\log_2(4)\rceil) - (2\cdot\lceil\log_2(3)\rceil + 2\cdot\lceil\log_2(5)\rceil)}{2\cdot(\lceil\log_2(3)\rceil - 1) + 2\cdot(\lceil\log_2(5)\rceil - 1) + (\lceil\log_2(4)\rceil - 1) + (\lceil\log_2(5)\rceil - 1) + (\lceil\log_2(8)\rceil - 1)}$$
$$= 0. \tag{3.17}$$

Table 3.3 shows the complexity and latency increase of the new DP-LDPC codes compared with the traditional ones. Even though the percentage increase in the source encoding complexity may seem large, the increase in the overall encoder complexity (when considering both source encoding and channel encoding) is much lower.

Table 3.3: Comparison of complexity and latency between the new DP-LDPC codes and the traditional DP-LDPC codes.

Code	$\mathbf{B}_{\text{J\_opt1}}^{0.04}$ vs $\mathbf{B}_{\text{J}}^{0.04}$	$\mathbf{B}_{\text{J4\_opt3}}^{0.01}$ vs $\mathbf{B}_{J4}^{0.01}$	$\mathbf{B}_{\text{J\_opt}}^{0.14}$ vs $\mathbf{B}_{\text{J\_org}}^{0.14}$
Complexity increase in source encoding	20%	23.1%	33.3%
$\Delta_{\text{latency,source}}$	0	12.5%	33.3%
Complexity increase in decoding	20%	23.1%	0
$\Delta_{\text{latency,dec}}$	0	10%	0

3.3.3 Summary of code design rule

We investigate different node-puncturing combinations in the DP-LDPC codes presented in Example #1 to Example #3.

In Example #1, the fifth variable node (VN) in the channel code is punctured, i.e., the ninth

Table 3.4: The channel thresholds $(E_s/N_0)_{\text{th}}$ (dB) of $\mathbf{B}^{0.04}$ in (3.9) for different x_1 and x_2 values and the eighth VN is punctured.

x_1 (punctured node)	0	0	0	0	1	2	3
x_2 (non-punctured node)	1	2	3	0	0	0	0
$(E_s/N_0)_{\text{th}}$ (dB)	-4.977	-4.870	-4.744	-5.031	$\mathbf{-5.074}$	-4.845	-4.546

Table 3.5: The channel thresholds $(E_s/N_0)_{\text{th}}$ (dB) of $\mathbf{B}_{\text{J3}}^{0.01}$ in (3.10) for different x_1 and x_2 values when the fifth VN in its channel protomatrix is punctured.

x_1 (non-punctured node)	0	0	0	0	1	2	3
x_2 (punctured node)	1	2	3	0	0	0	0
$(E_s/N_0)_{\text{th}}$ (dB)	$\mathbf{-9.394}$	-9.288	-9.043	-9.079	-9.048	-9.018	-8.972

Table 3.6: The channel thresholds $(E_s/N_0)_{\text{th}}$ (dB) of $\mathbf{B}_{\text{J4}}^{0.01}$ in (3.11) for different x_1 and x_2 values when the fifth VN in its channel protomatrix is punctured.

x_1 (non-punctured node)	0	0	0	0	1	2	3
x_2 (punctured node)	1	2	3	0	0	0	0
$(E_s/N_0)_{\text{th}}$ (dB)	$\mathbf{-9.309}$	-9.123	-8.833	-9.111	-9.117	-9.109	-9.092

VN in $\mathbf{B}^{0.04}$ (3.9) is punctured. Table 3.1 shows that $\mathbf{B}^{0.04}$ has the lowest channel threshold of -5.267 dB when $x_1 = 0$ (non-punctured node) and $x_2 = 1$ (punctured node).

Next, we suppose the eighth VN (instead of the ninth VN) in $\mathbf{B}^{0.04}$ is punctured. Table 3.4 shows the channel thresholds of $\mathbf{B}^{0.04}$ when the eighth VN is punctured, for different x_1 and x_2 values. We can see that $\mathbf{B}^{0.04}$ has the lowest channel threshold of -5.074 dB when $x_1 = 1$ (punctured node) and $x_2 = 0$ (non-punctured node).

In Example #2, the fourth VNs in the channel protomatrices of $\mathbf{B}_{\text{J3}}^{0.01}$ (3.10) and $\mathbf{B}_{\text{J4}}^{0.01}$ (3.11) are punctured. Table 3.2 shows that new codes with the lower channel thresholds can be obtained when $x_1 = 1$, 2, or 3 (punctured node) and $x_2 = 0$ (non-punctured node).

Next, we suppose the fifth VNs (instead of the fourth VNs) in the channel protomatrices of (3.10) and (3.11) are punctured. Table 3.5 and Table 3.6 show the channel thresholds of $\mathbf{B}_{\text{J3}}^{0.01}$ and $\mathbf{B}_{\text{J4}}^{0.01}$, respectively, for different x_1 and x_2 values when the fifth VNs in their channel protomatrices are punctured. We can see from both tables that the new codes achieve the lowest channel thresholds when $x_1 = 0$ (non-punctured node) and $x_2 = 1$ (punctured node). These achievable thresholds are lower than those where $x_1 = x_2 = 0$ (the original codes).

In Example #3, for $\mathbf{B}^{0.14}$ (3.13), the ninth and twelfth VNs are punctured. By using a

differential evolution (DE) method, a new code ($x_1 = x_3 = 1$ and $x_2 = x_4 = x_5 = x_6 = 0$) with a channel threshold 0.187 dB lower than the original one is obtained. We consider the puncturing of the ninth and twelfth VNs as "Case 1" and list the thresholds in Table 3.7.

Next, we select two VNs among the ninth VN to twelfth VN in $\mathbf{B}^{0.14}$ to be punctured, and we consider all different puncturing combinations. Then we use the DE method to search x_i ($i = 1, 2, ..., 6$) and find a new code with a lower channel threshold. Table 3.7 lists the channel thresholds of the original codes $\mathbf{B}^{0.14}_{\text{org}}$ ($x_i = 0\ \forall i$) with different combinations of punctured VNs. The table also lists the new codes and the corresponding thresholds. Six cases in Table 3.7 are discussed as follows.

Table 3.7: The channel thresholds $(E_s/N_0)_{\text{th}}$ (dB) of $\mathbf{B}^{0.14}$ for different x_i ($x_i = 1, 2, ..., 6$) values and different puncturing combinations.

	Indexes of punctured VNs	Codes	x_i values	$(E_s/N_0)_{\text{th}}$ (dB)
Case 1 (This result has been shown in Example #3)	$9, 12$	$\mathbf{B}^{0.14}_{\text{org0}}$	$x_i = 0\ \forall i$	-0.653
		$\mathbf{B}^{0.14}_{\text{opt0}}$	$x_1 = x_3 = 1$ (punctured VN), $x_i = 0$ ($i = 2, 4, 5, 6$)	-0.840
Case 2	$9, 10$	$\mathbf{B}^{0.14}_{\text{org1}}$	$x_i = 0\ \forall i$	0.433
		$\mathbf{B}^{0.14}_{\text{opt1}}$	$x_1 = x_2 = x_3 = 1$ (punctured VN), $x_5 = 1$ (punctured VN), $x_i = 0$ ($i = 4, 6$)	-0.225
Case 3	$9, 11$	$\mathbf{B}^{0.14}_{\text{org2}}$	$x_i = 0\ \forall i$	-0.459
		$\mathbf{B}^{0.14}_{\text{opt2}}$	$x_1 = x_3 = 1$ (punctured VN), $x_6 = 1$ (punctured VN), $x_i = 0$ ($i = 2, 4, 5$)	-0.770
Case 4	$10, 11$	$\mathbf{B}^{0.14}_{\text{org3}}$	$x_i = 0\ \forall i$	0.623
		$\mathbf{B}^{0.14}_{\text{opt3}}$	$x_4 = x_5 = 1$ (punctured VN), $x_i = 0$ ($i = 1, 2, 3, 6$)	-0.012
Case 5	$10, 12$	$\mathbf{B}^{0.14}_{\text{org4}}$	$x_i = 0\ \forall i$	0.622
		$\mathbf{B}^{0.14}_{\text{opt4}}$	$x_4 = x_5 = 1$ (punctured VN), $x_i = 0$ ($i = 1, 2, 3, 6$)	0.054
Case 6	$11, 12$	$\mathbf{B}^{0.14}_{\text{org5}}$	$x_i = 0\ \forall i$	-0.402
		$\mathbf{B}^{0.14}_{\text{opt5}}$	$x_6 = 1$ (punctured VN), $x_i = 0$ ($i = 1, 2, .., 5$)	-0.466

1. Case 1: The ninth and twelfth VNs are punctured. This puncturing method has been presented in Example #3. The newly constructed code is $\mathbf{B}^{0.14}$, where $x_1 = x_3 = 1$ and $x_2 = x_4 = x_5 = x_6 = 0$, denoted by $\mathbf{B}^{0.14}_{\text{opt0}}$ here. It has a channel threshold 0.187 dB lower than its corresponding original code $\mathbf{B}^{0.14}_{\text{org0}}$.

2. Case 2: The ninth and tenth VNs are punctured. The newly constructed code is $\mathbf{B}^{0.14}$, where $x_1 = x_2 = x_3 = x_5 = 1$ and $x_4 = x_6 = 0$, denoted by $\mathbf{B}^{0.14}_{\text{opt1}}$. It has a channel threshold 0.658 dB lower than its corresponding original code $\mathbf{B}^{0.14}_{\text{org1}}$.

3. Case 3: The ninth and eleventh VNs are punctured. The newly constructed code is $\mathbf{B}^{0.14}$, where $x_1 = x_3 = x_6 = 1$ and $x_2 = x_4 = x_5 = 0$, denoted by $\mathbf{B}^{0.14}_{\text{opt2}}$. It has a channel threshold 0.311 dB lower than its corresponding original code $\mathbf{B}^{0.14}_{\text{org2}}$.

4. Case 4: The tenth and eleventh VNs are punctured. The newly constructed code is $\mathbf{B}^{0.14}$, where $x_4 = x_5 = 1$ and $x_1 = x_2 = x_3 = x_6 = 0$, denoted by $\mathbf{B}^{0.14}_{\text{opt3}}$. It has a channel threshold 0.635 dB lower than its corresponding original code $\mathbf{B}^{0.14}_{\text{org3}}$.

5. Case 5: The tenth and twelfth VNs are punctured. The newly constructed code is $\mathbf{B}^{0.14}$, where $x_4 = x_5 = 1$ and $x_1 = x_2 = x_3 = x_6 = 0$, denoted by $\mathbf{B}^{0.14}_{\text{opt4}}$. It has a channel threshold of 0.568 dB lower than its corresponding original code $\mathbf{B}^{0.14}_{\text{org4}}$.

6. Case 6: The eleventh and twelfth VNs are punctured. The newly constructed code is $\mathbf{B}^{0.14}$, where $x_6 = 1$ and $x_1 = x_2 = x_3 = x_4 = x_5 = 0$, denoted by $\mathbf{B}^{0.14}_{\text{opt5}}$. It has a channel threshold of 0.064 dB lower than its corresponding original code $\mathbf{B}^{0.14}_{\text{org5}}$.

To summarize, entries corresponding to the punctured VNs are prefered to be considered when optimizing our code design based on traditional ones. In all cases, our code designs can achieve channel thresholds lower than those of the original codes.

3.4 Conclusion

In this chapter, we have proposed a new class of DP-LDPC codes, which replaces the identity matrix in the source-check-channel-variable (SCCV) linking matrix of a traditional DP-LDPC code with a lower or upper triangular matrix. Both theoretical and simulation results have demonstrated the superiority of the proposed DP-LDPC codes over the traditional ones. In the next chapter, we propose a new class of spatially coupled DP-LDPC codes for further performance improvement motivated by the superiority of SC-LDPC codes over LDPC block codes.

Chapter 4

New type of spatially coupled DP-LDPC codes

For the joint source-channel coding system based on DP-LDPC codes, two protograph-based LDPC codes are concatenated. In [47], two spatially coupled LDPC (SC-LDPC) codes are concatenated by an identity matrix for the JSCC system and a sliding window joint source-channel decoder is exploited. Better error performance is achieved for concatenated SC-LDPC codes compared with DP-LDPC block codes. Motivated by the result that spatially coupling the codes can improve the error performance, we propose a joint source-channel coding scheme that two SC-LDPC codes are spatially coupled [76]. Specifically, the parity-check matrices of two SC-LDPC codes are connected by a spatially-coupled SCCV linking matrix. By doing this, the flexibility of the code design is increased. Moreover, the correlation between codewords generated at different times is increased. The current compressed source symbols are not only related to the previous source symbols, but also related to the previously generated channel codewords due to the spatially coupled structure of the SCCV linking protomatrix. We call the proposed code "spatially-coupled double-protograph-based LDPC joint source-channel code" (SC-DP-LDPC JSCC), or just SC-DP-LDPC code for short.

We introduce two types of SC-DP-LDPC codes. The first one is SC-DP-LDPC terminated code and its protomatrix is given by $\mathbf{B}_{\mathrm{TD}}$ in Fig. 4.1. Referring to the figure, m_0 and m_1,

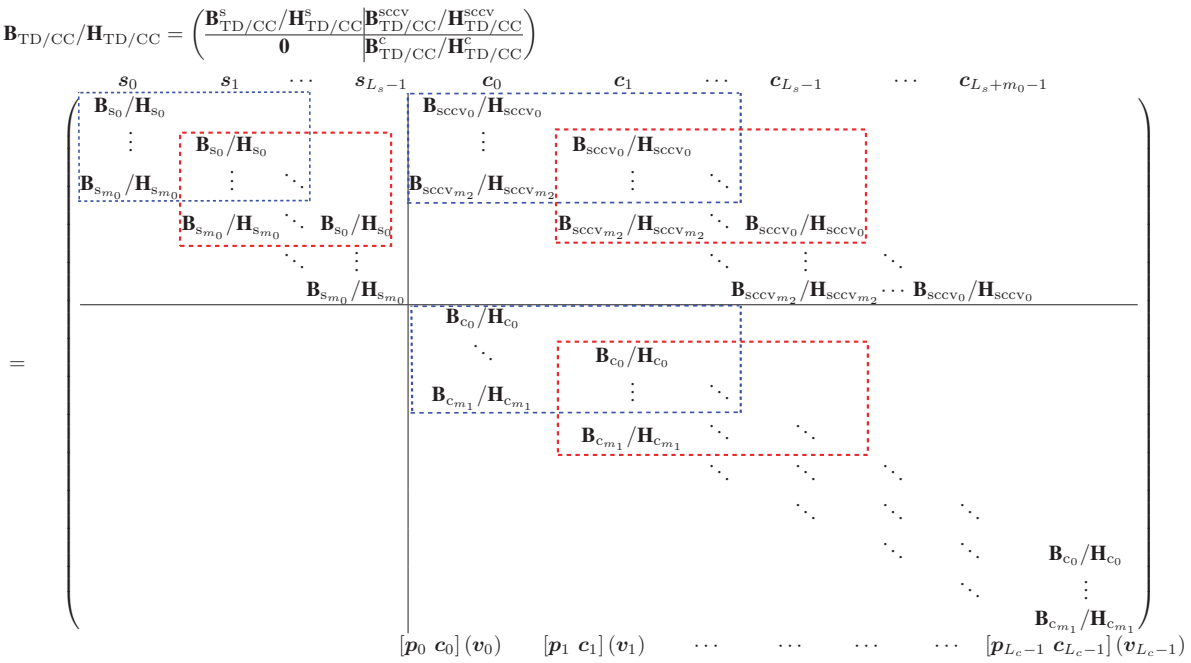


Figure 4.1: Protomatrix and parity-check matrix of proposed SC-DP-LDPC terminated codes are denoted by $\mathbf{B}_{\mathrm{TD}}$ and $\mathbf{H}_{\mathrm{TD}}$, respectively. When L_s and L_c are infinite, the protomatrix and parity-check matrix of SC-DP-LDPC convolutional code are denoted by $\mathbf{B}_{\mathrm{CC}}$ and $\mathbf{H}_{\mathrm{CC}}$, respectively.

respectively, represent the syndrome former memories of the source and channel SC-P-LDPC codes. Moreover, L_s and L_c denote the coupling lengths of the source SC-P-LDPC code and the channel SC-P-LDPC code, respectively. When both L_s and L_c tend to infinity, the SC-DP-LDPC terminated code becomes the SC-DP-LDPC convolutional code (SC-DP-LDPC-CC), whose protomatrix is denoted by $\mathbf{B}_{\mathrm{CC}}$.

In $\mathbf{B}^{\mathrm{s}}_{\mathrm{TD}}$ or $\mathbf{B}^{\mathrm{s}}_{\mathrm{CC}}$, the sub-base matrices $\mathbf{B}_{\mathrm{s}_i}$ $(i = 0, 1, \ldots, m_0)$ and $\mathbf{B}_{\mathrm{c}_i}$ $(i = 0, 1, \ldots, m_1)$ can, respectively, be constructed from the source and channel protomatrices of a DP-LDPC block code, e.g., from $\mathbf{B}_{\mathrm{s}}$ and $\mathbf{B}_{\mathrm{c}}$ in $\mathbf{B}_{\mathrm{J}}$ (3.1). Moreover, they should satisfy [70]

$$\sum_{i=0}^{m_0} \mathbf{B}_{\mathrm{s}_i} = \mathbf{B}_{\mathrm{s}} \text{ and } \sum_{i=0}^{m_1} \mathbf{B}_{\mathrm{c}_i} = \mathbf{B}_{\mathrm{c}}, \tag{4.1}$$

where $\mathbf{B}_{\mathrm{c}_0} = \begin{pmatrix} \mathbf{T}_{m_c} & \mathbf{B}_{m_c \times m_s} \end{pmatrix}$, $\mathbf{B}_{m_c \times m_s}$ is a base matrix of size $m_c \times m_s$, and $\mathbf{T}_{m_c}$ denotes a lower or upper triangular protomatrix of size $m_c \times m_c$ such that linear encoding can be implemented. We assume the sizes of $\mathbf{B}_{\mathrm{s}_i}$ $(i = 0, 1, \ldots, m_0)$ and $\mathbf{B}_{\mathrm{c}_i}$ $(i = 0, 1, \ldots, m_1)$ are $m_s \times n_s$ and $m_c \times n_c$, respectively.

Moreover, the sub-base matrices $\mathbf{B}_{\mathrm{s}_i}$ and $\mathbf{B}_{\mathrm{c}_i}$ are coupled by the SC-SCCV linking (proto)matrix, which consists of component base matrices $\mathbf{B}_{\mathrm{sccv}_i}$ $(i = 0, 1, \ldots, m_2)$ of size $m_s \times n_c$. Here m_2 denotes the syndrome former memory of the SC-SCCV linking matrix. We assume that $\mathbf{B}_{\mathrm{sccv}_0}$ has the same structure as $\mathbf{B}'_{\mathrm{sccv}}$ (i.e., $= \begin{pmatrix} \mathbf{0} & \mathbf{T}_{m_s} \end{pmatrix}$) so as to allow linear source compression. Note that if $\mathbf{T}_{m_s} = \mathbf{I}_{m_s}$ and $\mathbf{B}_{\mathrm{sccv}_i}$ $(i = 1, \ldots, m_2)$ are all zero matrices, the proposed SC-DP-LDPC code degenerates to the code in [47], i.e., the source SC-P-LDPC code and the channel SC-P-LDPC code are concatenated. In general, however, $\mathbf{B}_{\mathrm{sccv}_i}$ $(i = 1, \ldots, m_2)$ are not all zero matrices, and the two SC-P-LDPC codes are not concatenated. It is because the generation of the current compressed symbols depends not only on current and previous input source symbols, but also on previously generated channel codewords. (Details will be given in the next subsection.) To design the sub-protomatrices $\mathbf{B}_{\mathrm{sccv}_i}$ $(i = 1, \ldots, m_2)$, we apply the differential evolution algorithm in this chapter. For the SC-DP-LDPC terminated code $\mathbf{B}_{\mathrm{TD}}$, the corresponding coupling length for the SC-SCCV "terminated protomatrix" is $L_s + m_0$.

For the SC-DP-LDPC convolutional code $\mathbf{B}_{\mathrm{CC}}$, its overall code rate R_{CC} is the same as that of the corresponding DP-LDPC block code, i.e.,

$$R_{\mathrm{CC}} = n_s/(n_c - n_p), \tag{4.2}$$

where n_p is the number of punctured VNs in each $\mathbf{B}_{\mathrm{c}_i}$ $(i = 0, 1, \ldots, m_1)$; n_s and n_c are the number of variable nodes in $\mathbf{B}_{\mathrm{s}_i}$ $(i = 0, 1, ..., m_0)$ and $\mathbf{B}_{\mathrm{c}_i}$ $(i = 0, 1, ..., m_1)$, respectively. For the SC-DP-LDPC terminated code $\mathbf{B}_{\mathrm{TD}}$, its overall code rate is given by

$$R_{\mathrm{TD}} = n_s L_s/(L_c(n_c - n_p)) = L_s/L_c R_{\mathrm{CC}}. \tag{4.3}$$

To construct a parity-check matrix from the protomatrix, we can lift the protomatrix twice with lifting factors z_1 and z_2, respectively. The overall lifting factor equals $z = z_1 z_2$. Referring to Fig. 4.1, the parity-check matrix generated based on $\mathbf{B}_{\mathrm{TD}}$ is denoted as $\mathbf{H}_{\mathrm{TD}}$. Similarly, $\mathbf{B}_{\mathrm{CC}}$ becomes $\mathbf{H}_{\mathrm{CC}}$ after lifting.

4.1 Encoder

We assume a binary independent and identically distributed Bernoulli source and the probability of "1" in the source is denoted by p_1. We assume $m_0 = m_1 = m_2$ for a simple explanation. At time t $(= 0, 1, 2, \ldots)$, the source sequence of size $1 \times n_s z$ (z is the overall lifting factor) is denoted by $\boldsymbol{s}_t$. The compressed source sequence of size $1 \times m_s z$ is denoted by $\boldsymbol{c}_t$ at time t. The parity-check bit sequence of size $1 \times m_c z$ for the channel encoder is denoted by $\boldsymbol{p}_t$ at time t. The channel codeword sequence of size $1 \times n_c z$ is denoted by $\boldsymbol{v}_t$ at time t.

1. At $t = 0$, we generate the codeword $\boldsymbol{v}_0$ based on the input $\boldsymbol{s}_0$ and the parity-check matrix

$$\left(\begin{array}{c|cc} \boldsymbol{s}_0 & \boldsymbol{v}_0 = [\boldsymbol{p}_0 & \boldsymbol{c}_0] \\ \hline \mathbf{H}_{\mathrm{s}_0} & \mathbf{0}_{m_s z_1 z_2 \times m_c z_1 z_2} & \mathbf{H}^{z_1}_{\mathrm{T}_{m_s}} \\ \hline & \multicolumn{2}{c}{\mathbf{H}_{\mathrm{c}_0}} \end{array}\right), \tag{4.4}$$

where $\mathbf{H}_{\mathrm{s}_0}$ is on the 1-st row of $\mathbf{H}^{\mathrm{s}}_{\mathrm{TD}}$/$\mathbf{H}^{\mathrm{s}}_{\mathrm{CC}}$ in Fig. 4.1; $\mathbf{H}_{\mathrm{c}_0}$ is on the 1-st row of $\mathbf{H}^{\mathrm{c}}_{\mathrm{TD}}$/$\mathbf{H}^{\mathrm{c}}_{\mathrm{CC}}$; and $[\mathbf{0}_{m_s z_1 z_2 \times m_c z_1 z_2}\ \mathbf{H}^{z_1}_{\mathrm{T}_{m_s}}] = \mathbf{H}_{\mathrm{sccv}_0}$ is on the 1-st row of $\mathbf{H}^{\mathrm{sccv}}_{\mathrm{TD}}$/$\mathbf{H}^{\mathrm{sccv}}_{\mathrm{CC}}$. $\mathbf{0}_{m_s z_1 z_2 \times m_c z_1 z_2}$ is a zero matrix of size $m_s z_1 z_2 \times m_c z_1 z_2$. $\mathbf{H}^{z_1}_{\mathrm{T}_{m_s}}$ of size $m_s z_1 \times m_s z_1$ is the quasi-cyclic (QC) parity-check matrix obtained by lifting $\mathbf{T}_{m_s}$ in $\mathbf{B}_{\mathrm{sccv}_0} = [\mathbf{0}_{m_s \times m_c}\ \mathbf{T}_{m_s}]$ by $z = z_1 z_2$. $\mathbf{B}_{\mathrm{sccv}_0}$ here is on the 1-st row of $\mathbf{B}^{\mathrm{sccv}}_{\mathrm{CC}}$/$\mathbf{B}^{\mathrm{sccv}}_{\mathrm{TD}}$ in Fig. 4.1. First, $\boldsymbol{c}_0$ is computed based on $\boldsymbol{s}_0$ and $(\mathbf{H}_{\mathrm{s}_0} \mid \mathbf{0}_{m_s z_1 z_2 \times m_c z_1 z_2}\ \mathbf{H}^{z_1}_{\mathrm{T}_{m_s}})$. Then $\boldsymbol{p}_0$ is computed based on $\boldsymbol{c}_0$ and $\mathbf{H}_{\mathrm{c}_0}$.

2. At $0 < t < m_0$, the source sequence $\boldsymbol{s}_t$ is input into the encoder. $\boldsymbol{v}_t$ is generated based on the known $\boldsymbol{s}_0, \ldots, \boldsymbol{s}_t$ and $\boldsymbol{v}_0, \ldots, \boldsymbol{v}_{t-1}$ and the following parity-check matrix

$$\left(\begin{array}{cccc|ccc|c} \boldsymbol{s}_0 & \cdots & \boldsymbol{s}_{t-1} & \boldsymbol{s}_t & \boldsymbol{v}_0 & \cdots & \boldsymbol{v}_{t-1} & \boldsymbol{v}_t = [\boldsymbol{p}_t \quad \boldsymbol{c}_t] \\ \hline \mathbf{H}_{\mathrm{s}_t} & \cdots & \mathbf{H}_{\mathrm{s}_1} & \mathbf{H}_{\mathrm{s}_0} & \mathbf{H}_{\mathrm{sccv}_t} & \cdots & \mathbf{H}_{\mathrm{sccv}_1} & \mathbf{0}_{m_s z_1 z_2 \times m_c z_1 z_2}\ \mathbf{H}^{z_1}_{\mathrm{T}_{m_s}} \\ \hline & & & & \multicolumn{3}{c|}{\mathbf{H}_{\mathrm{c}_t} \quad \cdots \quad \mathbf{H}_{\mathrm{c}_1}} & \mathbf{H}_{\mathrm{c}_0} \end{array}\right), \tag{4.5}$$

where $\mathbf{H}_{\mathrm{s}_i}$ and $\mathbf{H}_{\mathrm{c}_i}$ $(i = 0, 1, ..., t)$ are on the $(t+1)$-th row of $\mathbf{H}^{\mathrm{s}}_{\mathrm{TD}}$/$\mathbf{H}^{\mathrm{s}}_{\mathrm{CC}}$ and $\mathbf{H}^{\mathrm{c}}_{\mathrm{TD}}$/$\mathbf{H}^{\mathrm{c}}_{\mathrm{CC}}$, respectively, in Fig. 4.1; and $\mathbf{H}_{\mathrm{sccv}_i}$ $(i = 1, \ldots, t)$ and $\mathbf{H}_{\mathrm{sccv}_0} = [\mathbf{0}_{m_s z_1 z_2 \times m_c z_1 z_2}\ \mathbf{H}^{z_1}_{\mathrm{T}_{m_s}}]$ are

on the $(t+1)$-th row of $\mathbf{H}_{\mathrm{TD}}^{\mathrm{sccv}}$/$\mathbf{H}_{\mathrm{CC}}^{\mathrm{sccv}}$. We first obtain the compressed source symbols $\boldsymbol{c}_t$ using the linear encoding method. Then we generate $\boldsymbol{p}_t$ based on $(\mathbf{H}_{\mathrm{c}_t} \ \cdots \ \mathbf{H}_{\mathrm{c}_1} \mid \mathbf{H}_{\mathrm{c}_0})$ and the inputs consisting of $\boldsymbol{v}_0, \ldots, \boldsymbol{v}_{t-1}$ and $\boldsymbol{c}_t$.

3. At $t \geq m_0$ ($m_0 \leq t < L_s - 1$ for the SC-DP-LDPC terminated code), the source sequence $\boldsymbol{s}_t$ is input into the encoder. $\boldsymbol{v}_t$ is generated based on the known $\boldsymbol{s}_{t-m_0}, \ldots, \boldsymbol{s}_t$ and $\boldsymbol{v}_{t-m_0}, \ldots, \boldsymbol{v}_{t-1}$ and the following parity-check matrix

$$\begin{array}{c} \begin{array}{cccc|ccc|c} \boldsymbol{s}_{t-m_0} & \cdots & \boldsymbol{s}_{t-1} & \boldsymbol{s}_t & \boldsymbol{v}_{t-m_0} & \cdots & \boldsymbol{v}_{t-1} & \boldsymbol{v}_t = [\boldsymbol{p}_t \quad \boldsymbol{c}_t] \end{array} \\ \left(\begin{array}{cccc|ccc|cc} \mathbf{H}_{\mathrm{s}_{m_0}} & \cdots & \mathbf{H}_{\mathrm{s}_1} & \mathbf{H}_{\mathrm{s}_0} & \mathbf{H}_{\mathrm{sccv}_{m_0}} & \cdots & \mathbf{H}_{\mathrm{sccv}_1} & \mathbf{0}_{m_s z_1 z_2 \times m_c z_1 z_2} & \mathbf{H}_{\mathrm{T}_{m_s}}^{z_1} \\ \hline & & & & \mathbf{H}_{\mathrm{c}_{m_0}} & \cdots & \mathbf{H}_{\mathrm{c}_1} & \multicolumn{2}{c}{\mathbf{H}_{\mathrm{c}_0}} \end{array} \right) \end{array}, \tag{4.6}$$

where $\mathbf{H}_{\mathrm{s}_i}$ and $\mathbf{H}_{\mathrm{c}_i}$ $(i = 0, 1, ..., m_0)$ are on the $(t+1)$-th row of $\mathbf{H}_{\mathrm{TD}}^{\mathrm{s}}$/$\mathbf{H}_{\mathrm{CC}}^{\mathrm{s}}$ and $\mathbf{H}_{\mathrm{TD}}^{\mathrm{c}}$/$\mathbf{H}_{\mathrm{CC}}^{\mathrm{c}}$, respectively, in Fig. 4.1; and $\mathbf{H}_{\mathrm{sccv}_i}$ $(i = 1, \ldots, m_0)$ and $\mathbf{H}_{\mathrm{sccv}_0} = [\mathbf{0}_{m_s z_1 z_2 \times m_c z_1 z_2} \ \mathbf{H}_{\mathrm{T}_{m_s}}^{z_1}]$ are on the $(t+1)$-th row of $\mathbf{H}_{\mathrm{TD}}^{\mathrm{sccv}}$/$\mathbf{H}_{\mathrm{CC}}^{\mathrm{sccv}}$. We first obtain the compressed source symbols $\boldsymbol{c}_t$ using the linear encoding method. Then we generate $\boldsymbol{p}_t$ based on $(\mathbf{H}_{\mathrm{c}_{m_0}} \ \cdots \ \mathbf{H}_{\mathrm{c}_1} \mid \mathbf{H}_{\mathrm{c}_0})$ and the inputs consisting of $\boldsymbol{v}_{t-m_0}, \ldots, \boldsymbol{v}_{t-1}$ and $\boldsymbol{c}_t$.

4. When the SC-DP-LDPC code is terminated, its protomatrix $\mathbf{B}_{\mathrm{TD}}$ is shown in Fig. 4.2 on the next page. At $t = L_s - 1$, the last source block $\boldsymbol{s}_{L_s-1}$ is input into the encoder.

   (a) Firstly, we generate $\boldsymbol{v}_{L_s-1}$ based on the known $\boldsymbol{s}_{L_s-m_0-1}$, ..., $\boldsymbol{s}_{L_s-1}$, $\boldsymbol{v}_{L_s-1-m_2}$, ..., $\boldsymbol{v}_{L_s-2}$, and the (lifted version of) the following base matrix

$$\begin{array}{c} \begin{array}{ccc|ccc|c} \boldsymbol{s}_{L_s-m_0-1} & \ldots & \boldsymbol{s}_{L_s-1} & \boldsymbol{v}_{L_s-m_2-1} & \ldots & \boldsymbol{v}_{L_s-2} & \boldsymbol{v}_{L_s-1} \end{array} \\ \left(\begin{array}{ccc|ccc|c} \mathbf{B}_{\mathrm{s}_{m_0}} & \ldots & \mathbf{B}_{\mathrm{s}_0} & \mathbf{B}_{\mathrm{sccv}_{m_2}} & \ldots & \mathbf{B}_{\mathrm{sccv}_1} & \mathbf{B}_{\mathrm{sccv}_0} \\ \hline & & & \mathbf{B}_{\mathrm{c}_{m_1}} & \ldots & \mathbf{B}_{\mathrm{c}_1} & \mathbf{B}_{\mathrm{c}_0} \end{array} \right) \end{array}, \tag{4.7}$$

   where $\mathbf{B}_{\mathrm{s}_i}$, $\mathbf{B}_{\mathrm{c}_i}$ and $\mathbf{B}_{\mathrm{sccv}_i}$ $(i = 0, 1, \ldots, m_0;\ m_0 = m_1 = m_2)$ are, respectively, on the L_s-th row of the source SC-LDPC protomatrix, the channel SC-LDPC protomatrix and the SC-SCCV linking protomatrix of $\mathbf{B}_{\mathrm{TD}}$.

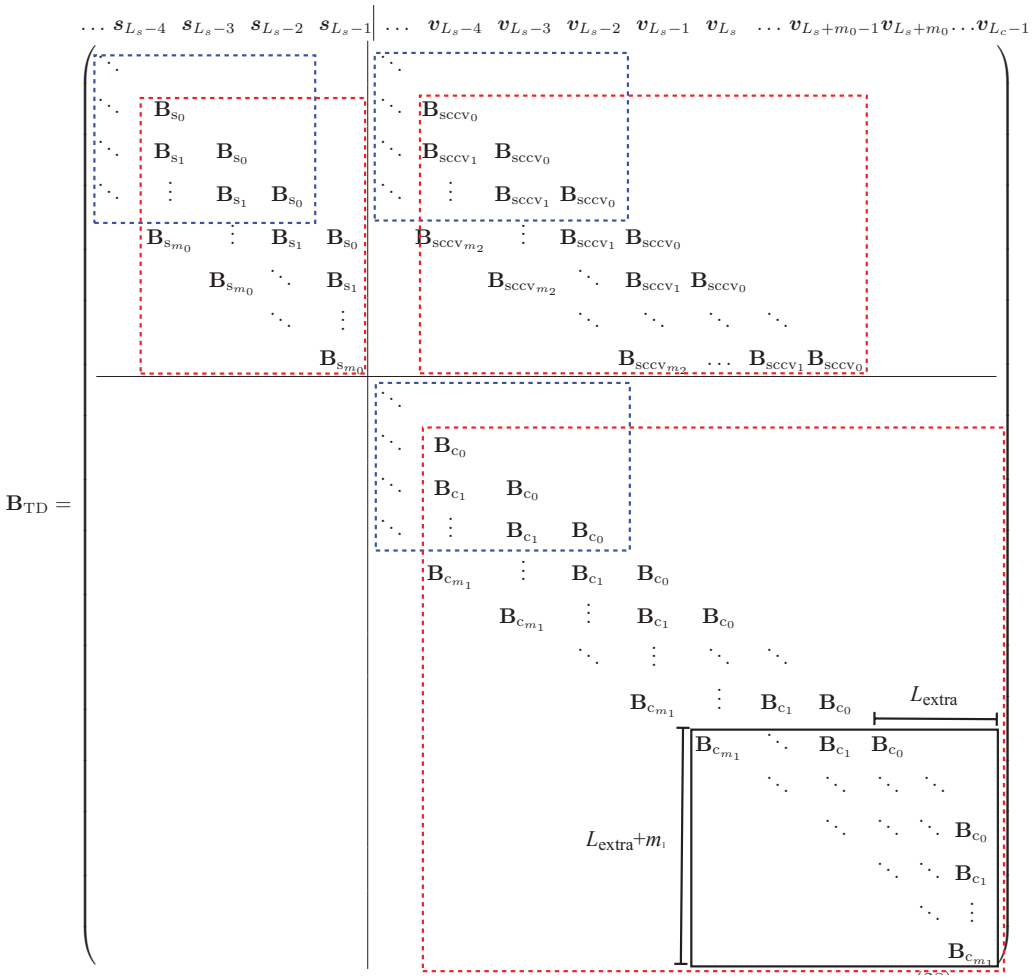


Figure 4.2: The protomatrix of $\mathbf{B}_{\mathrm{TD}}$.

(b) Secondly, without any new source input block, we continue to generate the channel codewords $\boldsymbol{v}_{L_s}$, ..., $\boldsymbol{v}_{L_s+m_0-1}$. Specifically, we generate $\boldsymbol{v}_{L_s-1+j}$ $(j = 1, ..., m_0)$ based on the known $\boldsymbol{s}_{L_s-m_0-1+j}$, ..., $\boldsymbol{s}_{L_s-1}$, $\boldsymbol{v}_{L_s-m_2-1+j}$, ..., $\boldsymbol{v}_{L_s-2+j}$ and the following base matrix

$$\begin{array}{c|ccc|ccc|c} \boldsymbol{s}_{L_s-m_0-1+j} & \cdots & \boldsymbol{s}_{L_s-1} & \boldsymbol{v}_{L_s-m_2-1+j} & \cdots & \boldsymbol{v}_{L_s-2+j} & \boldsymbol{v}_{L_s-1+j} \\ \hline \mathbf{B}_{\mathrm{s}_{m_0}} & \cdots & \mathbf{B}_{\mathrm{s}_j} & \mathbf{B}_{\mathrm{sccv}_{m_2}} & \cdots & \mathbf{B}_{\mathrm{sccv}_1} & \mathbf{B}_{\mathrm{sccv}_0} \\ & & & \mathbf{B}_{\mathrm{c}_{m_1}} & \cdots & \mathbf{B}_{\mathrm{c}_1} & \mathbf{B}_{\mathrm{c}_0} \end{array}, \tag{4.8}$$

where $(\mathbf{B}_{\mathrm{s}_{m_0}}, ..., \mathbf{B}_{\mathrm{s}_j})$, $\mathbf{B}_{\mathrm{c}_i}$ and $\mathbf{B}_{\mathrm{sccv}_i}$ $(i = 0, ..., m_1;\ m_0 = m_1 = m_2)$ are, respectively, on the $(L_s + j)$-th row of the source SC-LDPC protomatrix, the channel

SC-LDPC protomatrix and the SC-SCCV linking protomatrix of $\mathbf{B}_{\mathrm{TD}}$.

(c) Thirdly, we refer to the channel SC-LDPC sub-protomatrix shown in the black frame in Fig. 4.2 on the previous page. L_{extra} block columns are added after the block column corresponding to $\boldsymbol{v}_{L_s+m_0-1}$ (i.e., after the (L_s+m_0)-th block column of the channel SC-LDPC protomatrix of $\mathbf{B}_{\mathrm{TD}}$). Using this sub-protomatrix, the extra channel codeword sequences $\boldsymbol{v}_{L_s+m_0}, \boldsymbol{v}_{L_s+m_0+1}, \cdots, \boldsymbol{v}_{L_s+m_0+L_{\mathrm{extra}}-1}$ (consisting of mainly parity bits) are derived. As can be seen, this sub-protomatrix, i.e.,

$$\mathbf{B}_{\mathrm{remaining}} = \begin{array}{c} \begin{array}{cccc|ccc} \boldsymbol{v}_{L_s+m_0-m_1} & \cdots & \boldsymbol{v}_{L_s+m_0-1} & & \boldsymbol{v}_{L_s+m_0} & \cdots & \boldsymbol{v}_{L_c-1} \end{array} \\ \left(\begin{array}{cccc|ccc} \mathbf{B}_{\mathrm{c}_{m_1}} & \cdots & \mathbf{B}_{\mathrm{c}_1} & & \mathbf{B}_{\mathrm{c}_0} & & \\ & \ddots & \ddots & & \ddots & \ddots & \\ & & \ddots & & \ddots & \ddots & \mathbf{B}_{\mathrm{c}_0} \\ & & & & \ddots & \ddots & \mathbf{B}_{\mathrm{c}_1} \\ & & & & & \ddots & \vdots \\ & & & & & & \mathbf{B}_{\mathrm{c}_{m_1}} \end{array}\right) \end{array}, \tag{4.9}$$

consists of $L_{\mathrm{extra}}+m_1$ block rows and $L_{\mathrm{extra}}+m_1$ block columns. (Since the last channel codeword sequence is also denoted by $\boldsymbol{v}_{L_c-1}$, we have $L_s+m_0+L_{\mathrm{extra}}=L_c$). To ensure that encoding could be performed based on $\mathbf{B}_{\mathrm{remaining}}$ and the known $\boldsymbol{v}_{L_s+m_0-m_1}$, $\boldsymbol{v}_{L_s+m_0-m_1+1}$, …, $\boldsymbol{v}_{L_s+m_0-1}$, the number of parity-check equations in $\mathbf{B}_{\mathrm{remaining}}$ should be no more than the number of variable nodes in $\boldsymbol{v}_{L_s+m_0}$, $\boldsymbol{v}_{L_s+m_0+1}$, …, $\boldsymbol{v}_{L_s+m_0+L_{\mathrm{extra}}-1}$, i.e.,

$$m_c(L_{\mathrm{extra}}+m_1)-m_{\text{all-zero}} \leq n_c L_{\mathrm{extra}}, \tag{4.10}$$

where $m_{\text{all-zero}}$ denotes the number of all-zero rows in $\mathbf{B}_{\mathrm{remaining}}$. In other words, L_{extra} should be the smallest integer selected such that (4.10) is satisfied.

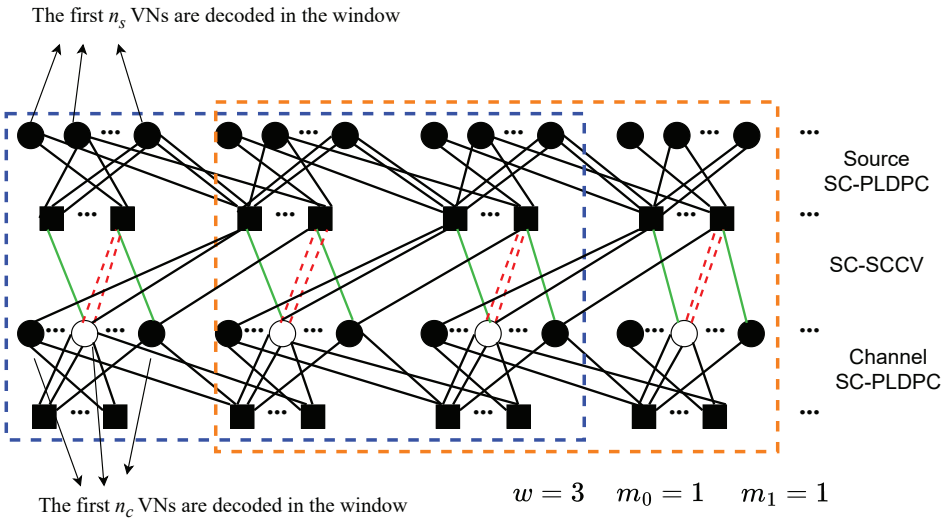


Figure 4.3: Protograph of SC-DP-LDPC codes when $m_0 = m_1 = m_2 = 1$ and $w = 3$.

4.2 Sliding window joint BP decoder

We apply a sliding window decoding, which is proposed in [47] and introduced in Section 2.4.2, to our JSCC system and we use w to denote the window size. We refer to the sliding window defined by the blue dashed frames in Fig. 4.1 (i.e., the blue dashed frame in Fig. 4.3). In this window (and also other windows except the last one), w block columns and w block rows of the source SC-LDPC parity-check matrix, the channel SC-LDPC parity-check matrix, and the SC-SCCV linking parity-check matrix are contained. The first $n_s z$ source symbols and the first $n_c z$ channel codewords can be decoded by the joint BP algorithm introduced in Section 2.2.2. In the next decoding timeslot, the window in Fig. 4.1 will slide to the right and downward, i.e., moving from the blue dashed frames to the red dashed frames (i.e., moving the blue frame in Fig. 4.3 to the orange frame), and so on. Moreover, all updated log-likelihood ratio (LLR) messages and previously decoded source symbols and channel codewords would be used to facilitate decoding the source symbols and channel codewords in this window.

For the SC-DP-LDPC terminated codes, the last window denoted by red frames in Fig. 4.2 contains the last w block columns and their connected rows of the source SC-DP-LDPC parity-check matrix, the last $w + m_0$ block columns and their connected rows of the SC-SCCV linking parity-check matrix, and the last $w + m_0 + L_{\text{add}}$ block columns and their connected rows of

the channel SC-DP-LDPC parity-check matrix. In the last time slot, we decode all the source symbols in the last window using the joint BP algorithm. Also, decoding in the last window can be facilitated by previously decoded source symbols and channel codewords and messages updated in the previous window denoted by the blue frames in Fig. 4.2. The decoding method of the SC-DP-LDPC codes is similar to that of concatenated SC-LDPC code [47]. The main difference between them is the structure of SCCV linking matrix in the window.

4.3 Threshold calculation

The source threshold and channel threshold for an SC-DP-LDPC code are calculated using a window-based joint protograph-based extrinsic information transfer (JP-EXIT) algorithm similar to that in [47]. We use p_{th} and $(E_s/N_0)_{\text{th}}$ to denote the source threshold and the channel threshold, respectively, where E_s represents the average energy per source symbol and N_0 denotes the power spectral density of the additive white Gaussian noise (AWGN) channel.

$$\mathbf{B}_w = \left(\begin{array}{cccc|cccc} \mathbf{B}_{\text{s}_0} & & & & \mathbf{B}_{\text{sccv}_0} & & & \\ \mathbf{B}_{\text{s}_1} & \mathbf{B}_{\text{s}_0} & & & \mathbf{B}_{\text{sccv}_1} & \mathbf{B}_{\text{sccv}_0} & & \\ \vdots & \ddots & \ddots & & \vdots & \ddots & \ddots & \\ \mathbf{B}_{\text{s}_{m_0}} & \cdots & \mathbf{B}_{\text{s}_1} & \mathbf{B}_{\text{s}_0} & \mathbf{B}_{\text{sccv}_{m_2}} & \cdots & \mathbf{B}_{\text{sccv}_1} & \mathbf{B}_{\text{sccv}_0} \\ \hline & & & & \mathbf{B}_{\text{c}_0} & & & \\ & & & & \mathbf{B}_{\text{c}_1} & \mathbf{B}_{\text{c}_0} & & \\ & & & & \vdots & \ddots & \ddots & \\ & & & & \mathbf{B}_{\text{c}_{m_1}} & \cdots & \mathbf{B}_{\text{c}_1} & \mathbf{B}_{\text{c}_0} \end{array}\right) \tag{4.11}$$

We apply the protomatrix in a window to the algorithm. The protomatrix includes w block rows and block columns of the source SC-LDPC protomatrix, the channel SC-LDPC protomatrix, and the SC-SCCV linking protomatrix. For example, $\mathbf{B}_w$ in (4.11) denotes the protomatrix in a window when $w = m_0 + 1$ in the case of $m_0 = m_1 = m_2$. A channel threshold $(E_s/N_0)_{\text{th}}$ is defined as the smallest value of E_s/N_0 which makes $I_{\text{APP}}(j)$ for the first n_s source VNs and

first n_c channel VNs in the protomatrix in the window reaching "1" for a given p_1. A source threshold p_{th} is defined as the largest value of p_1 which makes $I_{\text{APP}}(j)$ for $j = 1, 2, \ldots, n_s$ reaching "1" when the channel information is perfect, i.e, the noise is very small.

4.4 Optimization method

We use the differential evolution (DE) [29, 34, 73] (introduced in Section 2.5) to search the source sub-protomatrices $\mathbf{B}_{\mathrm{s}_{i_\mathrm{s}}}$ $(i_\mathrm{s} = 0, 1, ..., m_0)$ and the channel sub-protomatrices $\mathbf{B}_{\mathrm{c}_{i_\mathrm{c}}}$ $(i_\mathrm{c} = 0, 1, ..., m_1)$ with low channel thresholds based on the window-based JP-EXIT algorithm. $\mathbf{B}_{\mathrm{s}_{i_\mathrm{s}}}$ $(i_\mathrm{s} = 0, 1, ..., m_0)$ and $\mathbf{B}_{\mathrm{c}_{i_\mathrm{c}}}$ $(i_\mathrm{c} = 0, 1, ..., m_1)$ should satisfy the following conditions.

(i) $\sum_{i_\mathrm{s}=0}^{m_0} \mathbf{B}_{\mathrm{s}_{i_\mathrm{s}}} = \mathbf{B}_\mathrm{s}$, where $\mathbf{B}_\mathrm{s}$ denotes the source block code in a DP-LDPC block code $\mathbf{B}_\mathrm{J}$ (3.1);

(ii) $\sum_{i_\mathrm{c}=0}^{m_1} \mathbf{B}_{\mathrm{c}_{i_\mathrm{c}}} = \mathbf{B}_\mathrm{c}$, where $\mathbf{B}_\mathrm{c}$ denotes the channel block code in a DP-LDPC block code $\mathbf{B}_\mathrm{J}$ (3.1);

(iii) The part of $\mathbf{B}_{\mathrm{c}_0}$ corresponding to the parity-check bits is a lower or upper triangular protomatrix whose diagonal elements are all "1" and other non-zero elements are non-negative integers.

We define the following parameters in the DE algorithm.

- G as the number of generations;
- D as the number of candidate matrices;
- p_m as the mutation probability.
- p_c as the crossover probability.

The search process is as follows.

Step a) Initialization: Set the generation counter $g = 0$. We randomly generate D source sub-protomatrices and channel sub-protomatrices at the g-th generation. They all satisfy conditions (i) to (iii). We use $\mathbf{B}^d_{\mathrm{s}_{i_\mathrm{s}}}$ $(i_\mathrm{s} = 0, 1, ..., m_0)$ and $\mathbf{B}^d_{\mathrm{c}_{i_\mathrm{c}}}$ $(i_\mathrm{c} = 0, 1, ..., m_1)$ to denote the d-th source sub-protomatrices and channel sub-protomatrices, respectively.

Step b) Mutation and Crossover: We generate the d-th $(d = 1, 2, ..., D)$ source sub-protomatrices and channel sub-protomatrices by

$$\begin{aligned}(\mathbf{B}^d_{\mathrm{s}_{i_\mathrm{s}}})_{\mathrm{mutat}} &= \Theta(\mathbf{B}^{r_1}_{\mathrm{s}_{i_\mathrm{s}}} + p_m(\mathbf{B}^{r_2}_{\mathrm{s}_{i_\mathrm{s}}} - \mathbf{B}^{r_3}_{\mathrm{s}_{i_\mathrm{s}}})),\ i_\mathrm{s} = 0, 1, ..., m_0,\\ (\mathbf{B}^d_{\mathrm{c}_{i_\mathrm{c}}})_{\mathrm{mutat}} &= \Theta(\mathbf{B}^{r_1}_{\mathrm{c}_{i_\mathrm{c}}} + p_m(\mathbf{B}^{r_2}_{\mathrm{c}_{i_\mathrm{c}}} - \mathbf{B}^{r_3}_{\mathrm{c}_{i_\mathrm{c}}})),\ i_\mathrm{c} = 0, 1, ..., m_1,\end{aligned} \tag{4.12}$$

respectively, at the mutation step. The function $\Theta(x)$ returns the integer by rounding x's absolute value. r_1, r_2, and r_3 are randomly chosen from $\{1, 2, ..., D\}$. Moreover, we should make sure the sub-protomatrices satify conditions (i) to (iii). Next, we implement the crossover operation. Entries in each sub-protomatrix group $(\mathbf{B}^d_{(\mathrm{s/c})_{i_{(\mathrm{s/c})}}})_{\mathrm{mutat}}$ $(i_\mathrm{s} = 0, 1, .., m_0;\ i_\mathrm{c} = 0, 1, ..., m_1)$ obtained after the mutation operation are chosen with the probability of p_c to replace values in the corresponding sub-protomatrix group $\mathbf{B}^d_{\mathrm{s/c}_{i_{\mathrm{s/c}}}}$. The d-th sub-protomatrix group after the crossover operation is denoted by $(\mathbf{B}^d_{\mathrm{s}_{i_\mathrm{s}}})_{\mathrm{crsvr}}$ $(i_\mathrm{s} = 0, 1, .., m_0)$ and $(\mathbf{B}^d_{\mathrm{c}_{i_\mathrm{c}}})_{\mathrm{crsvr}}$ $(i_\mathrm{c} = 0, 1, .., m_1)$. We also need to make sure this sub-protomatrix group satisfies conditions (i) to (iii).

Step c) Selection: Record the channel thresholds of the SC-DP-LDPC codes obtained after the crossover operation. We compare them with the channel thresholds of $\mathbf{B}^d_{\mathrm{s}_{i_\mathrm{s}}}$ $(i_\mathrm{s} = 0, 1, ..., m_0)$ and $\mathbf{B}^d_{\mathrm{c}_{i_\mathrm{c}}}$ $(i_\mathrm{c} = 0, 1, ..., m_1; d = 1, 2, ..., D)$. If the threshold of the d-th sub-protomatrix group obtained at Step b) is lower, the code group ($\mathbf{B}^d_{\mathrm{s}_{i_\mathrm{s}}}$ $(i_\mathrm{s} = 0, 1, ..., m_0)$ and $\mathbf{B}^d_{\mathrm{c}_{i_\mathrm{c}}}$ $(i_\mathrm{c} = 0, 1, ..., m_1)$) is replaced with this group. After this selection operation, D sub-protomatrix groups are obtained for the next generation and we set $g = g + 1$;

Step d) Repeat Step b) and Step c) until $g = G$.

4.5 Results and analysis

In this section, we first construct some source SC-P-LDPC and channel SC-P-LDPC code pairs from DP-LDPC block codes [74] proposed in Section 3.3 using the aforementioned differential evolution (DE) algorithm. The parameters in DE are set as: $G = 100$, $D = 100$, $p_m = 0.5$, and $p_c = 0.8$. Then we design corresponding SC-SCCV linking matrices for these code pairs. We run simulations and record the source symbol error rates (SSERs) of the codes. We set the maximum (minimum) numbers of simulated frames to $G_{\max}$ ($G_{\min}$) for an SC-DP-LDPC terminated code. The number of iterations performed within each window is $I_{\max} = 100$. At a given E_s/N_0, the simulation is stopped and the SSER is recorded if (i) the number of simulated frames has reached $G_{\max}$, or (ii) the number of frames simulated is not smaller than $G_{\min}$ and the number of error frames is at least 50. When comparing the SSER performance of SC-DP-LDPC codes with that of DP-LDPC block codes, we assume that both decoders have the same decoding latency, i.e., both the block decoder and the window decoder need to receive the same number of channel inputs before starting the decoding process [47].

Example #1: We start with the DP-LDPC block code

$$\mathbf{B}_{\text{J\_opt1}}^{0.04} = \begin{pmatrix} \mathbf{B}_{\text{s}}^{0.04} & \mathbf{B}_{\text{sccv}}^{0.04} \\ \mathbf{0}_{m_c \times n_s} & \mathbf{B}_{\text{c}}^{0.04} \end{pmatrix} = \left(\begin{array}{cccc|ccccc} 1 & 1 & 2 & 1 & 0 & 0 & 0 & 1 & 0 \\ 2 & 2 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ \hline 0 & 0 & 0 & 0 & 1 & 1 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 2 & 2 \end{array}\right), \tag{4.13}$$

which is designed at $p_1 = 0.04$ in [74] and has been presented in Section 3.3.1. Setting $m_0 = m_1 = 1$, we construct the source sub-protomatrices $\mathbf{B}_{\text{s}_0}^{0.04}$ and $\mathbf{B}_{\text{s}_1}^{0.04}$ and channel sub-

protomatrices $\mathbf{B}_{c_0}^{0.04}$ and $\mathbf{B}_{c_1}^{0.04}$ for the SC-P-LDPC code pairs and obtain

$$\mathbf{B}_{s_1}^{0.04} = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \end{pmatrix} \; ; \; \mathbf{B}_{s_0}^{0.04} = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{pmatrix} \; ;$$
$$\mathbf{B}_{c_1}^{0.04} = \begin{pmatrix} 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 \end{pmatrix} \; ; \; \mathbf{B}_{c_0}^{0.04} = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \end{pmatrix} . \tag{4.14}$$

Moreover, VNs corresponding to the fourth column of the channel sub-protomatrices are punctured. Supposing we use only the SCCV linking protomatrix $\mathbf{B}_{\text{sccv}}^{0.04} = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \end{pmatrix}$ in $\mathbf{B}_{\text{J\_opt1}}^{0.04}$ to concatenate the source SC-P-LDPC code and the channel SC-P-LDPC code in (4.14), we denote the corresponding SC-DP-LDPC code as $\mathbf{B}_{\text{TD}}^{0.04}$.

Next, we set $m_2 = 1$ and window size $w = 8$, and use the DE algorithm to construct a SC-SCCV linking matrix for the source SC-P-LDPC code and the channel SC-P-LDPC code in (4.14). To reduce the searching complexity, each entry in the SC-SCCV linking protomatrix is set to either "0" or "1". The SC-SCCV linking sub-protomatrices obtained are

$$\mathbf{B}_{\text{sccv}_1}^{0.04} = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix} \; ; \; \mathbf{B}_{\text{sccv}_0}^{0.04} = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \end{pmatrix} , \tag{4.15}$$

and the corresponding SC-DP-LDPC code is denoted as $\mathbf{B}_{\text{TD}_{\text{new}}}^{0.04}$. Furthermore, when the lower triangular base matrix in $\mathbf{B}_{\text{sccv}_0}^{0.04}$ is replaced by an identity matrix, i.e., $\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$, we denote the corresponding SC-DP-LDPC code as $\mathbf{B}_{\text{TD}_{\text{new}'}}^{0.04}$.

Table 4.1 shows the source thresholds and channel thresholds of the codes constructed above, i.e., $\mathbf{B}_{\text{TD}}^{0.04}$, $\mathbf{B}_{\text{TD}_{\text{new}}}^{0.04}$, and $\mathbf{B}_{\text{TD}_{\text{new}'}}^{0.04}$. Moreover, the thresholds of some state-of-the-art DP-LDPC block codes designed at $p_1 = 0.04$ are listed [34,36,37]. As in [47], using a single SCCV linking protomatrix to concatenate the source SC-P-LDPC code and the channel SC-P-LDPC code lowers the channel threshold of the block code $\mathbf{B}_{\text{J\_opt1}}^{0.04}$ [74] from -5.267 dB to -5.909 dB

Table 4.1: Channel thresholds and source thresholds of different codes at $p_1 = 0.04$. For overall code rates of 1.000 and 0.985, the corresponding Shannon limits are -7.00 dB and -7.01 dB, respectively.

Code	$\mathbf{B}^{0.04}_{\mathrm{TD_{new}}}$	$\mathbf{B}^{0.04}_{\mathrm{TD_{new'}}}$	$\mathbf{B}^{0.04}_{\mathrm{TD}}$	$\mathbf{B}^{0.04}_{\mathrm{J\_opt1}}$ [74]	($\mathbf{B}_{\mathrm{s1}}$,$\mathbf{B}_{\mathrm{c1}}$) in [34, Table I]	$\mathbf{B}^{\mathrm{opti\_1}}_{\mathrm{J}}$ [37]	($\mathbf{B}^{0.04}_{\mathrm{J-new1}}$) [36, Table I]
w	8	8	8	Not applicable			
$(E_s/N_0)_{\mathrm{th}}$ (dB)	-6.218	-6.147	-5.909	-5.267	-5.135	-5.127	-5.573
p_{th}	0.090	0.090	0.090	0.082	0.063	0.082	0.082

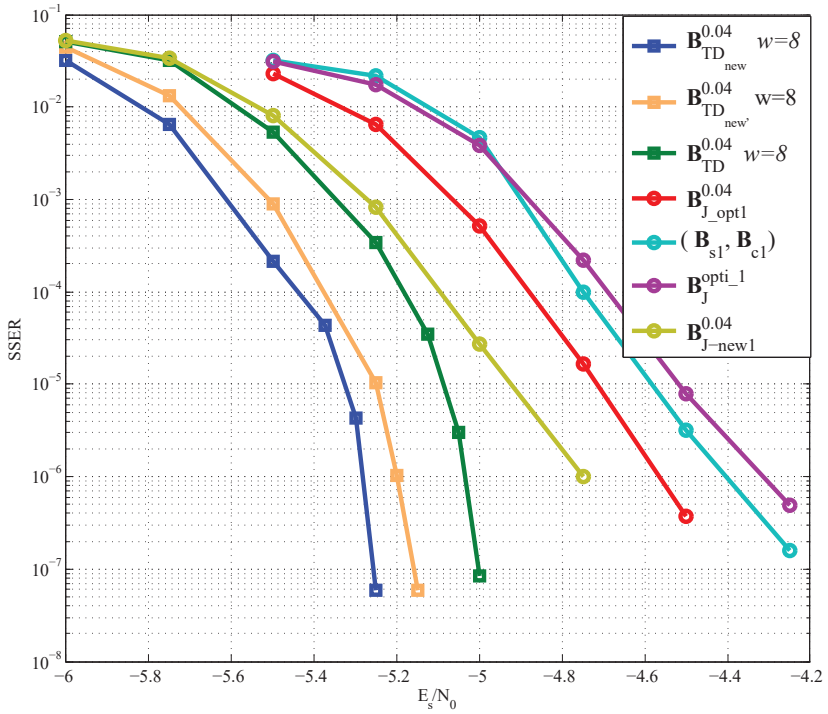


Figure 4.4: SSER performance comparison between $\mathbf{B}^{0.04}_{\mathrm{TD_{new}}}$, $\mathbf{B}^{0.04}_{\mathrm{TD_{new'}}}$, $\mathbf{B}^{0.04}_{\mathrm{TD}}$, and the state-of-the-art DP-LDPC block codes at $p_1 = 0.04$ under the same decoding latency. $L_s = 128$, $L_c = 130$, and $z = 400$ for SC-DP-LDPC codes. For DP-LDPC block codes, the lifting factors are all $z = 3200$. Overall code rates of SC-DP-LDPC codes and DP-LDPC codes are 0.985 and 1.000, respectively.

($\mathbf{B}^{0.04}_{\mathrm{TD}}$). Moreover, by using the proposed spatially-coupled SCCV linking protomatrices, the threshold can be further reduced to -6.147 dB ($\mathbf{B}^{0.04}_{\mathrm{TD_{new'}}}$) and -6.218 dB ($\mathbf{B}^{0.04}_{\mathrm{TD_{new}}}$). Among all the codes, the proposed SC-DP-LDPC code $\mathbf{B}^{0.04}_{\mathrm{TD_{new}}}$ achieves the lowest channel threshold (-6.218 dB). In addition, all the SC-DP-LDPC codes i.e., $\mathbf{B}^{0.04}_{\mathrm{TD}}$, $\mathbf{B}^{0.04}_{\mathrm{TD_{new}}}$ and $\mathbf{B}^{0.04}_{\mathrm{TD_{new'}}}$, achieve better (lower) channel thresholds and better (higher) source thresholds than the block codes.

Figure 4.4 plots the SSER performance of these codes under the same decoding latency. To make a fair comparison, we set a large source coupling length, i.e., $L_s = 128$, and a large channel coupling length, i.e., $L_c = 130$ for SC-DP-LDPC codes. Also, we set $G_{\max} = 6000$ and

$G_{\min} = 200$. We can see that our proposed SC-DP-LDPC code $\mathbf{B}^{0.04}_{\mathrm{TD_{new}}}$ shows the best SSER performance over all E_s/N_0 values. In particular, at an SSER of 10^{-6}, $\mathbf{B}^{0.04}_{\mathrm{TD_{new}}}$ has gains of about 0.08 dB, 0.20 dB, and 0.70 dB over $\mathbf{B}^{0.04}_{\mathrm{TD_{new'}}}$, $\mathbf{B}^{0.04}_{\mathrm{TD}}$, and $\mathbf{B}^{0.04}_{\mathrm{J\_opt1}}$ [74], respectively. $\mathbf{B}^{0.04}_{\mathrm{TD_{new}}}$ also outperforms other state-of-the-art DP-LDPC block codes. The SSER results of the codes are also consistent with the theoretical channel thresholds.

Example #2: Using the DP-LDPC block code

$$\mathbf{B}^{0.01}_{\mathrm{J3\_opt3}} = \begin{pmatrix} \mathbf{B}^{0.01}_{\mathrm{s}} & \mathbf{B}^{0.01}_{\mathrm{sccv}} \\ \mathbf{0}_{m_c \times n_s} & \mathbf{B}^{0.01}_{\mathrm{c}} \end{pmatrix} = \left(\begin{array}{cccccccc|ccccc} 1 & 1 & 2 & 1 & 3 & 1 & 3 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 2 & 1 & 2 & 1 & 2 & 1 & 2 & 0 & 0 & 0 & 3 & 1 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 2 & 1 \end{array}\right) \tag{4.16}$$

designed at $p_1 = 0.01$ in [74] and presented in Section 3.3.1, we set $m_0 = m_1 = 1$ and construct the source sub-protomatrices $\mathbf{B}^{0.01}_{\mathrm{s}_0}$ and $\mathbf{B}^{0.01}_{\mathrm{s}_1}$ and the channel sub-protomatrices $\mathbf{B}^{0.01}_{\mathrm{c}_0}$ and $\mathbf{B}^{0.01}_{\mathrm{c}_1}$, which are given by

$$\begin{aligned} \mathbf{B}^{0.01}_{\mathrm{s}_1} &= \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 0 & 2 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 \end{pmatrix} ; \ \mathbf{B}^{0.01}_{\mathrm{s}_0} = \begin{pmatrix} 1 & 1 & 2 & 0 & 2 & 1 & 1 & 0 \\ 1 & 2 & 1 & 1 & 0 & 1 & 1 & 1 \end{pmatrix} ; \\ \mathbf{B}^{0.01}_{\mathrm{c}_1} &= \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 2 & 0 \end{pmatrix} ; \ \mathbf{B}^{0.01}_{\mathrm{c}_0} = \begin{pmatrix} 1 & 0 & 0 & 3 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 \end{pmatrix} . \end{aligned} \tag{4.17}$$

Moreover, VNs corresponding to the fourth column of the channel sub-protomatrices are punctured. When we use only the SCCV linking protomatrix $\mathbf{B}^{0.01}_{\mathrm{sccv}} = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 3 & 1 \end{pmatrix}$ in $\mathbf{B}^{0.01}_{\mathrm{J3\_opt3}}$ to concatenate the source and channel SC-P-LDPC code pair in (4.17), we denote the corresponding SC-DP-LDPC code as $\mathbf{B}^{0.01}_{\mathrm{TD}}$.

Next, we set $m_2 = 1$ and $w = 8$, and use the DE algorithm to construct a SC-SCCV

Table 4.2: Channel thresholds and source thresholds of different codes at $p_1 = 0.01$. For overall code rates of 2.000 and 1.969, the corresponding Shannon limits are -12.02 dB and -12.03 dB, respectively.

Code	$\mathbf{B}_{\mathrm{TD}_{\mathrm{new}}}^{0.01}$	$\mathbf{B}_{\mathrm{TD}_{\mathrm{new}'}}^{0.01}$	$\mathbf{B}_{\mathrm{TD}}^{0.01}$	$\mathbf{B}_{\mathrm{J3\_opt3}}^{0.01}$ [74]	$\mathbf{B}_{\mathrm{J3}}^{0.01}$ [35, Table I]	$\mathbf{B}_{\mathrm{J-new}}^{0.01}$ [36, Table II]	$\mathbf{B}_{\mathrm{J2}}^{\mathrm{opt1}}$ [77, Table II]	$(3,12)$ &$(3,6)$ [47]	$(5,20)$ &$(3,6)$ [47]
w	8	8	8	Not applicable				16	16
$(E_s/N_0)_{\mathrm{th}}$ (dB)	-10.915	-10.809	-10.499	-9.734	-9.324	-9.725	-10.050	-9.703	-10.242
p_{th}	0.031	0.031	0.031	0.028	0.028	0.028	0.028	0.031	0.036

linking matrix for the source and channel SC-P-LDPC code pair in (4.17). We obtain a new SC-DP-LDPC code $\mathbf{B}_{\mathrm{TD}_{\mathrm{new}}}^{0.01}$, whose SC-SCCV linking sub-protomatrices are

$$\mathbf{B}_{\mathrm{sccv}_1}^{0.01} = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix} \; ; \; \mathbf{B}_{\mathrm{sccv}_0}^{0.01} = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \end{pmatrix} . \tag{4.18}$$

When we use $\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ to replace the lower triangular base matrix in $\mathbf{B}_{\mathrm{sccv}_0}^{0.01}$, we denote the corresponding code as $\mathbf{B}_{\mathrm{TD}_{\mathrm{new}'}}^{0.01}$.

Table 4.2 compares the channel thresholds of $\mathbf{B}_{\mathrm{TD}_{\mathrm{new}}}^{0.01}$, $\mathbf{B}_{\mathrm{TD}_{\mathrm{new}'}}^{0.01}$, $\mathbf{B}_{\mathrm{TD}}^{0.01}$, state-of-the-art DP-LDPC block codes designed at $p_1 = 0.01$ [35, 36, 74, 77], and concatenated SC-P-LDPC codes proposed in [47]. We can see that $\mathbf{B}_{\mathrm{TD}_{\mathrm{new}}}^{0.01}$ achieves the best (lowest) channel threshold among all codes. In particular, $\mathbf{B}_{\mathrm{TD}_{\mathrm{new}}}^{0.01}$ achieves a channel threshold which is 1.212 dB and 0.673 dB lower than those of $(3,12)\&(3,6)$ and $(5,20)\&(3,6)$ code pairs [47], respectively.

Figure 4.5 plots the SSER performance of SC-DP-LDPC codes and DP-LDPC block codes under the same decoding latency. We can see that $\mathbf{B}_{\mathrm{TD}_{\mathrm{new}}}^{0.01}$ has gains of about 0.15 dB, 0.20 dB, and 0.35 dB over $\mathbf{B}_{\mathrm{TD}_{\mathrm{new}'}}^{0.01}$, $\mathbf{B}_{\mathrm{TD}}^{0.01}$, and $\mathbf{B}_{\mathrm{J3\_opt3}}^{0.01}$ at an SSER of 10^{-6}, respectively. Moreover, $\mathbf{B}_{\mathrm{TD}_{\mathrm{new}}}^{0.04}$ outperforms other DP-LDPC block codes [35, 36, 77]. The SSER results are also consistent with the theoretical threshold given in Table 4.2.

Table 4.3 shows the channel thresholds of $\mathbf{B}_{\mathrm{TD}_{\mathrm{new}}}^{0.04}$ and $\mathbf{B}_{\mathrm{TD}_{\mathrm{new}}}^{0.01}$ for different w values. Considering $\mathbf{B}_{\mathrm{TD}_{\mathrm{new}}}^{0.04}$, the channel threshold is reduced from -5.949 dB to -6.218 dB (reduced by 0.269 dB) and then to -6.250 dB (reduced by another 0.032 dB) when w is increased from 6 to 8 and then to 10. The marginal reduction in channel threshold becomes very small when w is

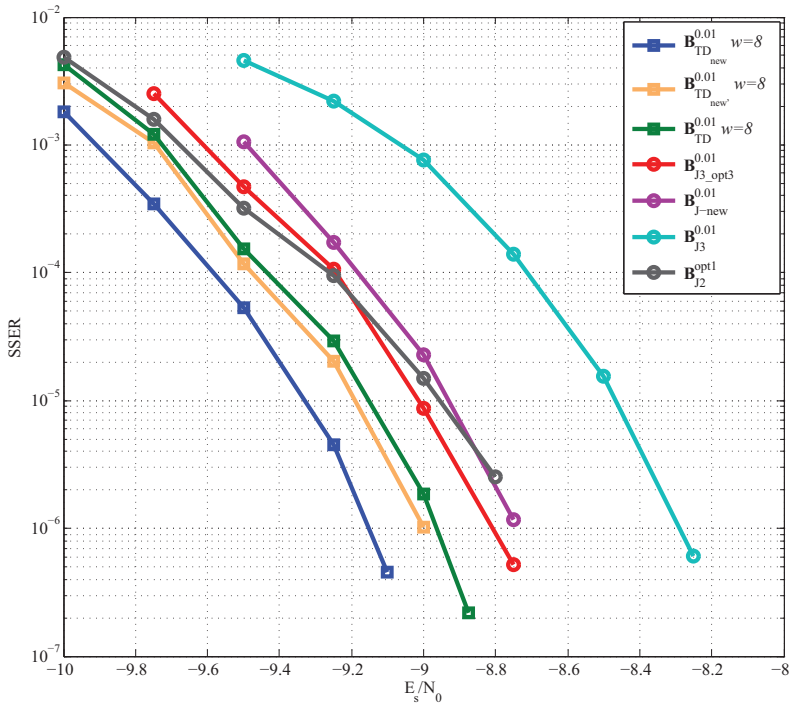


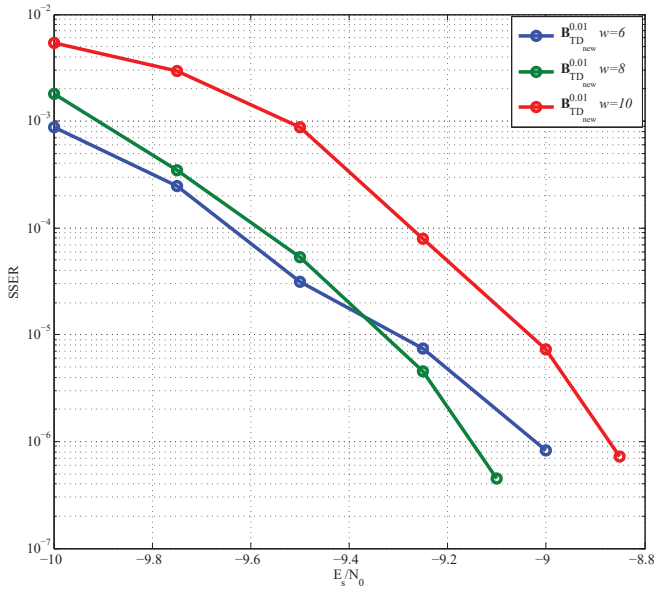
Figure 4.5: SSER performance of $\mathbf{B}^{0.01}_{\mathrm{TD_{new}}}$ and state-of-the-art DP-LDPC block codes at $p_1 = 0.01$ under the same decoding latency. $L_s = 128$, $L_c = 130$, and $z = 200$ for SC-DP-LDPC codes. For all DP-LDPC block codes, their lifting factors are $z = 1600$. Overall code rates of SC-DP-LDPC codes and DP-LDPC codes are 1.969 and 2.000, respectively.

Table 4.3: Channel thresholds of $\mathbf{B}^{0.04}_{\mathrm{TD_{new}}}$ and $\mathbf{B}^{0.01}_{\mathrm{TD_{new}}}$. The Shannon limits equal -7.00 dB and -12.02 dB when $(p_1 = 0.04, R_{\mathrm{CC}} = 1)$ and $(p_1 = 0.01, R_{\mathrm{CC}} = 2)$, respectively.

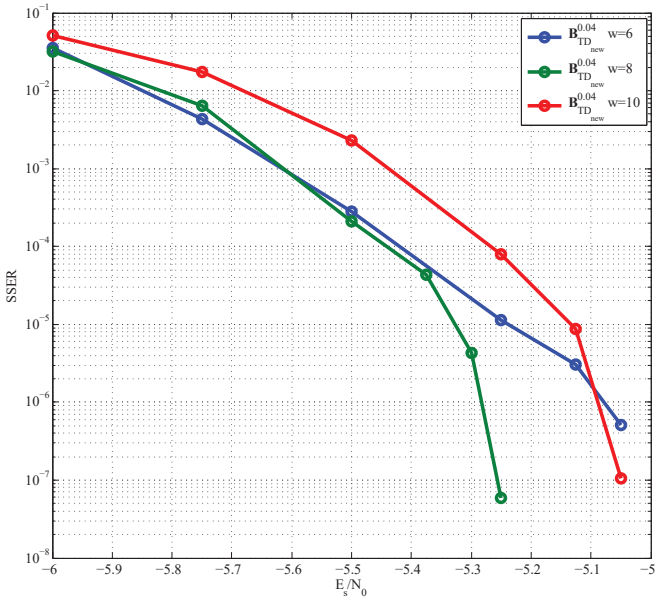
Code	$\mathbf{B}^{0.04}_{\mathrm{TD_{new}}}$			$\mathbf{B}^{0.01}_{\mathrm{TD_{new}}}$		
w	6	8	10	6	8	10
$(E_s/N_0)_{\mathrm{th}}$ (dB)	-5.949	-6.218	-6.250	-10.855	-10.915	-10.930

increased from 8 to 10. The same observation is found for $\mathbf{B}^{0.01}_{\mathrm{TD_{new}}}$.

Figure 4.6 plots the error performance of $\mathbf{B}^{0.04}_{\mathrm{TD_{new}}}$ and $\mathbf{B}^{0.01}_{\mathrm{TD_{new}}}$. To ensure that the number of source symbols in a window remains the same under different combinations of w and z, we set $(w = 6, z = 534)$, $(w = 8, z = 400)$, and $(w = 10, z = 320)$ for $\mathbf{B}^{0.04}_{\mathrm{TD_{new}}}$; and $(w = 6, z = 268)$, $(w = 8, z = 200)$, and $(w = 10, z = 160)$ for $\mathbf{B}^{0.01}_{\mathrm{TD_{new}}}$. For $\mathbf{B}^{0.04}_{\mathrm{TD_{new}}}$ and $\mathbf{B}^{0.01}_{\mathrm{TD_{new}}}$, they both yield the best overall error performance with $w = 8$, particularly in the high E_s/N_0 region. Notably, their channel thresholds remain relatively stable after w is increased to 8. Based on this observation, it can be concluded that to achieve the best combination of w and z and obtain the best error performance, it is advisable to first determine the minimum



(a)



(b)

Figure 4.6: SSER performance of $\mathbf{B}^{0.04}_{\mathrm{TD}_{\mathrm{new}}}$ and $\mathbf{B}^{0.01}_{\mathrm{TD}_{\mathrm{new}}}$ for different w values and different z values when $p_1 = 0.01$ and $p_1 = 0.04$, respectively. We set $L_s = 128$ and $L_c = 130$. To obtain almost the same number of source symbols in a window, we set: (a) $z = 268, 200, 160$ for $\mathbf{B}^{0.01}_{\mathrm{TD}_{\mathrm{new}}}$ when $w = 6, 8, 10$, respectively, at $p_1 = 0.01$; (b) $z = 534, 400, 320$ for $\mathbf{B}^{0.04}_{\mathrm{TD}_{\mathrm{new}}}$ when $w = 6, 8, 10$, respectively, at $p_1 = 0.04$.

value of w based on the condition that the channel threshold does not decrease significantly as w increases. Once the minimum w is determined, the lifting factor is determined based on the fixed code length in a window.

Table 4.4: Channel thresholds of different codes at $p_1 = 0.02$. Shannon limit equals -9.21 dB.

Code	$\mathbf{B}^{0.01}_{\mathrm{TD}_{\mathrm{new}}}$	$\mathbf{B}^{0.01}_{\mathrm{TD}_{\mathrm{new}'}}$	$(3,12)\&(3,6)$ [47]	$(5,20)\&(3,6)$ [47]
w	8	8	16	16
$(E_s/N_0)_{\mathrm{th}}$ (dB)	-8.273	-8.210	-7.617	-8.124

In Table 4.4, the channel thresholds of $\mathbf{B}^{0.01}_{\mathrm{TD}_{\mathrm{new}}}$, $\mathbf{B}^{0.01}_{\mathrm{TD}_{\mathrm{new}'}}$, and $(3,12)\&(3,6)$ and $(5,20)\&(3,6)$ code pairs at $p_1 = 0.02$ are listed. Again, the proposed SC-DP-LDPC codes achieve the best channel thresholds.

Figure 4.7 plots the SSER performance of these four codes at $p_1 = 0.01$ and $p_1 = 0.02$ under the same decoding latency. We set $L_s = 13$ and $L_c = 15$ for $\mathbf{B}^{0.01}_{\mathrm{TD}_{\mathrm{new}}}$ and $\mathbf{B}^{0.01}_{\mathrm{TD}_{\mathrm{new}'}}$, $L_s = 26$ and $L_c = 30$ for the $(3,12)\&(3,6)$ code pair, and $L_s = 39$ and $L_c = 45$ for the $(5,20)\&(3,6)$ code pair, to make sure they have the same code rate $R_{\mathrm{TD}} \approx 1.733$. We also set $G_{\max} = 60000$ and $G_{\min} = 500$ in the simulations. We can see that at an SSER of 10^{-6}, $\mathbf{B}^{0.01}_{\mathrm{TD}_{\mathrm{new}}}$ has gains of about 0.20 dB and 0.18 dB over $\mathbf{B}^{0.01}_{\mathrm{TD}_{\mathrm{new}'}}$ at $p_1 = 0.01$ and $p_1 = 0.02$, respectively. $\mathbf{B}^{0.01}_{\mathrm{TD}_{\mathrm{new}}}$ has gains of about 1.80 dB and 1.25 dB, respectively, over $(3,12)\&(3,6)$ and $(5,20)\&(3,6)$ code pairs at an SSER of 10^{-6} when $p_1 = 0.01$. It also has gains of about 1.20 dB and 0.95 dB, respectively, over $(3,12)\&(3,6)$ and $(5,20)\&(3,6)$ code pairs at an SSER of 10^{-6} when $p_1 = 0.02$.

4.6 Conclusions

We have proposed a novel JSCC scheme based on SC-DP-LDPC codes. The special feature of the proposed code is that the source spatially-coupled LDPC code and the channel spatially-coupled LDPC code are linked by spatially-coupled SCCV connections. Theoretical and simulation results show that the proposed codes are superior to the concatenated SC-P-LDPC codes and state-of-the-art DP-LDPC block codes. Existing research on DP-LDPC-based joint source-channel coding schemes has primarily concentrated on optimizing codes for low-entropy sources. The proposed SC-DP-LDPC code and the proposed DP-LDPC code introduced in Chapter 3 also focus on low-entropy sources (i.e., small p_1 values). It has been proved in [29,49,50] and introduced in Chapter 2.2.4 that source thresholds can be improved by adding

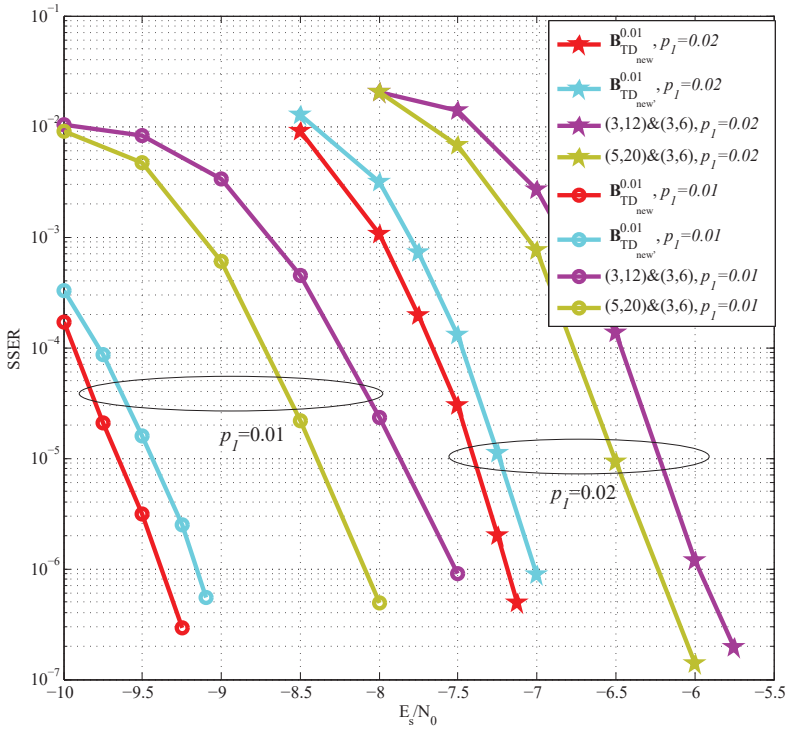


Figure 4.7: SSER performance of $\mathbf{B}^{0.01}_{\mathrm{TD}_{\mathrm{new}}}$, $\mathbf{B}^{0.01}_{\mathrm{TD}_{\mathrm{new}'}}$ and concatenated SC-P-LDPC codes proposed in [47] under the same decoding latency and the same code rate. $L_s = 13$, $L_c = 15$, $z = 200$, and $w = 8$ for $\mathbf{B}^{0.01}_{\mathrm{TD}_{\mathrm{new}}}$ and $\mathbf{B}^{0.01}_{\mathrm{TD}_{\mathrm{new}'}}$. $L_s = 26$, $L_c = 30$, $z = 200$, and $w = 16$ for $(3, 12)\&(3, 6)$. $L_s = 39$, $L_c = 45$, $z = 200$, and $w = 16$ for $(5, 20)\&(3, 6)$.

a source-variable-channel-check (SVCC) linking matrix. With a high source threshold, a joint source-channel coding system can handle high-entropy sources without the changing source compression rate. In the next chapter, we propose two novel types of joint source-channel codes that exhibit excellent error performance for both low-entropy and high-entropy sources.

Chapter 5

Novel DP-LDPC codes and SC-DP-LDPC codes with SVCC connections

In this chapter, we propose a new type of joint source-channel block codes, where the source VNs and channel CNs are connected for a high source threshold and the one-to-one connections between source CNs and channel VNs are replaced with the one-to-many connections (i.e., $\mathbf{I}_{m_s}$ is replaced with $\mathbf{T}_{m_s}$) for a good channel threshold. We construct some new joint source-channel block codes for both low-entropy and high-entropy sources by considering both source and channel thresholds. Theoretical analyses and simulation results both show the proposed codes outperform state-of-the-art DP-LDPC codes for low-entropy and high-entropy sources. Due to the outstanding performance of SC-DP-LDPC [76] over the DP-LDPC block code, we also spatially couple the joint source-channel block codes and propose a new type of spatially coupled joint source-channel codes (SC-JSCC).

The followings are our main contributions.

1. We propose a general structure of a joint source-channel block code (JSC-BC) and present its encoding method. Based on the JP-EXIT algorithm, we provide a simplified algorithm called untransmitted protograph-based extrinsic information transfer (UP-EXIT) algorithm, which only includes source VNs and punctured VNs in the channel protograph in calculating the source threshold of the code.

2. We design some new JSC-BCs for sources with different entropies. To save searching time, we impose some code design rules. We first search for candidate JSC-BCs with high source thresholds by using our proposed UP-EXIT algorithm, and then select the codes among those candidate codes with low channel thresholds by using the JP-EXIT algorithm. By doing this, searching time is limited due to the small protomatrix used in UP-EXIT. Simulation results and theoretical channel thresholds both show the new JSC-BCs outperforming existing DP-LDPC block codes.

3. We propose a new type of spatially-coupled joint source-channel code (SC-JSCC), present its encoding method. and use the sliding window joint belief propagation (BP) decoding algorithm to decode the SC-JSCC.

4. Based on the JSC-BCs, we construct some new SC-JSCCs. Theoretical analyses and error rate simulation results show that new SC-JSCC codes, whose source, channel, SCCV, and SVCC protomatrices all have spatially coupled structures, can have better error performance than the proposed JSC-BCs and existing SC-DP-LDPC codes. Moreover, the SC-JSCCs can obtain channel thresholds close to the Shannon limit for both low-entropy and high-entropy sources.

The structure of this chapter is as follows. Section 5.1 shows the structure, the encoding method, the protograph-based analysis method, and the design method of the proposed joint source-channel block code (JSC-BC). In the same section, the thresholds and simulation results of the newly constructed JSC-BCs are also presented. In Section 5.2, we introduce the structure and encoding method of the proposed spatially coupled joint source-channel code (SC-JSCC). Based on the JSC-BCs, we construct some SC-JSCCs. Some error performance comparison results are also given. Finally, we give some concluding remarks in Section 5.3.

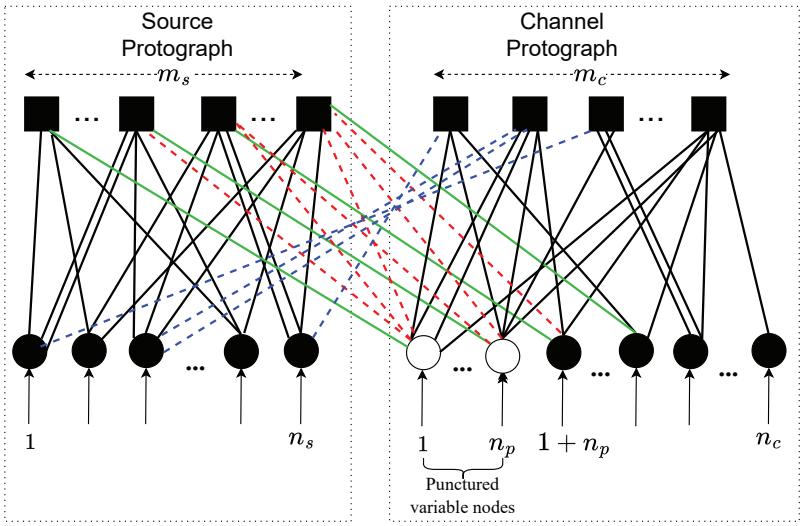


Figure 5.1: Protograph representation of joint source-channel block code.

5.1 Protograph-based joint source-channel block code

Figure 5.1 illustrates the protograph of the proposed joint source-channel block code (JSC-BC). Compared to Fig. 3.1 which represents the DP-LDPC JSCC system proposed in Section 3, Figure 5.1 involves new connections between VNs in the source protograph and the CNs in the channel protograph. Moreover, it can be denoted by a joint protomatrix, which is denoted by

$$\mathbf{B}_{\mathrm{J_{new}}} = \begin{pmatrix} \mathbf{B}_{\mathrm{s}} & \mathbf{B}'_{\mathrm{sccv}} \\ \mathbf{B}_{\mathrm{svcc}} & \mathbf{B}_{\mathrm{c}} \end{pmatrix}, \tag{5.1}$$

where

$$\mathbf{B}'_{\mathrm{sccv}} = \begin{pmatrix} \mathbf{T}_{m_s} & \mathbf{0}_{m_s \times m_c} \end{pmatrix}. \tag{5.2}$$

$\mathbf{B}_{\mathrm{s}}$ of size $m_s \times n_s$ and $\mathbf{B}_{\mathrm{c}}$ of size $m_c \times n_c$ are the source protomatrix and the channel protomatrix, respectively. $\mathbf{T}_{m_s}$ of size $m_s \times m_s$ is a lower or upper triangular matrix with "1"s on the diagonal. $\mathbf{B}_{\mathrm{svcc}}$ of size $m_c \times n_s$ is the source-variable-channel-check (SVCC) linking base matrix, which represents the blue connections in Fig. 5.1.

5.1.1 Encoder

We lift the protomatrix $\mathbf{B}_{\mathrm{J_{new}}}$ to form a large low-density parity-check matrix with a quasi-cyclic structure in two steps. Firstly, we lift $\mathbf{B}_{\mathrm{J_{new}}}$ with a relatively small lifting factor z_1 to eliminate all entries with values larger than 1, thereby obtaining a matrix with only 0's and 1's, which is denoted by

$$\mathbf{B}_{\mathrm{J_{new}}}^{z_1} = \left(\begin{array}{c|c} \mathbf{B}_{\mathrm{s}}^{z_1} & \mathbf{T}_{m_s z_1}\ \mathbf{0}_{m_s z_1 \times m_c z_1} \\ \hline \mathbf{B}_{\mathrm{svcc}}^{z_1} & \mathbf{B}_{\mathrm{c}}^{z_1} \end{array}\right). \tag{5.3}$$

$\mathbf{B}_{\mathrm{s}}^{z_1}$, $\mathbf{B}_{\mathrm{c}}^{z_1}$, $\mathbf{T}_{m_s z_1}$, and $\mathbf{B}_{\mathrm{svcc}}^{z_1}$ are the matrices obtained by lifting $\mathbf{B}_{\mathrm{s}}$, $\mathbf{B}_{\mathrm{c}}$, $\mathbf{T}_{m_s}$, and $\mathbf{B}_{\mathrm{svcc}}$ in the first lifting step, respectively.

Secondly, we lift $\mathbf{B}_{\mathrm{J_{new}}}^{z_1}$ with a large lifting factor z_2 to obtain a large parity-check matrix with a quasi-cyclic structure, which is denoted by

$$\mathbf{H}_{\mathrm{J_{new}}} = \left(\begin{array}{c|c} \mathbf{H}_{\mathrm{s}} & \mathbf{H}_{\mathrm{T}}\ \mathbf{0}_{m_s z_1 z_2 \times m_c z_1 z_2} \\ \hline \mathbf{H}_{\mathrm{svcc}} & \mathbf{H}_{\mathrm{c}} \end{array}\right). \tag{5.4}$$

$\mathbf{H}_{\mathrm{J_{new}}}$ contains $(m_s + m_c)z_1$ rows and $(n_s + n_c)z_1$ columns of sub-matrices. The submatrix in the ith row and jth column is represented by $\mathbf{I}_{z_2}^{h_{i,j}}$, which denotes a circulant permutation matrix (CPM) with size $z_2 \times z_2$ obtained by cyclically right-shifting the identity matrix $\mathbf{I}_{z_2}$ by $h_{i,j}$ columns. The second lifting aims to maximize the girth (shortest cycle) of the resultant parity-check matrix.

We use $\boldsymbol{s}$ to denote the source sequence of length $1 \times N_s = 1 \times n_s z_1 z_2$. Entries in $\boldsymbol{s}$ are "0" or "1". The probability of "1" in $\boldsymbol{s}$ is denoted by p_1 and the probability of "0" given by $1 - p_1$. The distribution of "0" and "1" in $\boldsymbol{s}$ follows a Bernoulli distribution. By using the same linear source compression method introduced in Section 3.1 [74], we first obtain the compressed source sequence $\boldsymbol{u}$ of length $1 \times M_s = 1 \times m_s z_1 z_2$ based on $\left(\begin{array}{c|c} \mathbf{H}_{\mathrm{s}} & \mathbf{H}_{\mathrm{T}}\ \mathbf{0}_{m_s z_1 z_2 \times m_c z_1 z_2} \end{array}\right)$, where $\mathbf{H}_{\mathrm{s}}$ and $\mathbf{H}_{\mathrm{T}}$ are, respectively, the parity-check matrices obtained by lifting $\mathbf{B}_{\mathrm{s}}^{z_1}$ and $\mathbf{T}_{m_s z_1}$ in the second lifting step.

Then we combine $\boldsymbol{s}$ and $\boldsymbol{c}$ as the input for the channel encoder. Next, we generate parity-

check bits based on $\left(\begin{array}{c|c} \mathbf{H}_{\text{svcc}} & \mathbf{H}_{\text{c}} \end{array} \right)$, where $\mathbf{H}_{\text{svcc}}$ and $\mathbf{H}_{\text{c}}$ are obtained by lifting $\mathbf{B}_{\text{svcc}}^{z_1}$ and $\mathbf{B}_{\text{c}}^{z_1}$, respectively. The parity-check bits and $\boldsymbol{c}$ (except punctured nodes) are transmitted over the channel. The overall code rate of a JSC-BC is $R = n_s/(n_c - n_p)$, where n_p denotes the number of punctured VNs in $\mathbf{B}_{\text{c}}$.

5.1.2 Calculation of Source Thresholds

In [32], a source protograph-based extrinsic information transfer (SPEXIT) algorithm is applied to calculate the source threshold of a double protograph with no connections between VNs in the source P-LDPC code and CNs in the channel P-LDPC code. In [29], a generalized source protograph-based extrinsic information transfer (GSP-EXIT) algorithm does not consider the case with punctured variable nodes. Thus both algorithms are not suitable for calculating the source thresholds of the proposed JSC-BC. Here, we propose a general algorithm, called the *untransmitted protograph-based EXIT (UP-EXIT) algorithm*, for calculating the source threshold of the proposed JSC-BC. We denote the submatrix corresponding to the untransmitted VNs (i.e., source VNs and punctured channel VNs) by $\mathbf{B}_{\text{u}}$, i.e.,

$$\mathbf{B}_{\text{u}} = \begin{pmatrix} \mathbf{B}_{\text{s}} & \mathbf{B}_{\text{sccv}}^{\text{p}} \\ \mathbf{B}_{\text{svcc}} & \mathbf{B}_{\text{c}}^{\text{p}} \end{pmatrix}, \tag{5.5}$$

where $\begin{pmatrix} \mathbf{B}_{\text{sccv}}^{\text{p}} \\ \mathbf{B}_{\text{c}}^{\text{p}} \end{pmatrix}$ denotes the first n_p columns of $\begin{pmatrix} \mathbf{B}_{\text{sccv}}^{'} \\ \mathbf{B}_{\text{c}} \end{pmatrix}$ and corresponds to the n_p punctured channel VNs. As shown in Fig. 5.1, we assume that the punctured VNs in the channel protograph are located at the first n_p VNs of the channel protograph.

We define the following parameters.

- $\mathcal{V} = \{V_1, V_2, ..., V_{n_s+n_c}\}$ denotes the set of VNs in $\mathbf{B}_{\text{J}_{\text{new}}}$;
- $\mathcal{V}_{\text{u}} = \{V_1, V_2, ..., V_{n_s+n_p}\}$ denotes the set of untransmitted VNs, i.e., source VNs and punctured channel VNs;

- $\mathcal{C} = \{C_1, C_2, ..., C_{m_s+m_c}\}$ denotes the set of CNs;
- $I_{\text{A\_VC}}(i,j)$ denotes the a priori mutual information (AMI) from the jth VN $\in \mathcal{V}$ to the ith CN;
- $I_{\text{A\_V}_\text{u}\text{C}}(i,j)$ denotes the a priori mutual information (AMI) from the jth VN $\in \mathcal{V}_\text{u}$ to the ith CN;
- $I_{\text{A\_CV}}(i,j)$ denotes the AMI from the ith CN $\in \mathcal{C}$ to the jth VN $\in \mathcal{V}$;
- $I_{\text{E\_CV}}(i,j)$ denotes the extrinsic mutual information (EMI) from the ith CN $\in \mathcal{C}$ to the jth VN $\in \mathcal{V}$;
- $I_{\text{A\_CV}_\text{u}}(i,j)$ denotes the AMI from the ith CN $\in \mathcal{C}$ to the jth VN $\in \mathcal{V}_\text{u}$;
- $I_{\text{E\_CV}_\text{u}}(i,j)$ denotes the extrinsic mutual information (EMI) from the ith CN $\in \mathcal{C}$ to the jth VN $\in \mathcal{V}_\text{u}$;
- $I_{\text{E\_VC}}(i,j)$ denotes the EMI from the jth VN $\in \mathcal{V}$ to the ith CN;
- $I_{\text{E\_V}_\text{u}\text{C}}(i,j)$ denotes the EMI from the jth VN $\in \mathcal{V}_\text{u}$ to the ith CN;
- $I_{\text{APP}}(j)$ denotes the MI between the a posteriori probability log-likelihood ratio (APP-LLR) of the jth VN $\in \mathcal{V}$ and its corresponding symbol.

We use p_{th} to denote the source threshold, i.e., the maximum value of p_1 which allows $I_{\text{APP}}(j)$ ($j = 1, 2, ..., n_s + n_p$) reaching "1" when the channel information is perfect.

The traditional joint protograph-based EXIT (JP-EXIT) algorithm [33] has been introduced in Section 2.2.3. In evaluating the source thresholds of the JSC-BC, the channel information is assumed to be perfect. In other words, we can assume that (i) $I_{\text{A\_VC}}(i,j) = I_{\text{E\_VC}}(i,j) = 1$ for $i = 1, 2, \ldots, m_s + m_c, j = n_s + n_p + 1, n_s + n_p + 2, ..., n_s + n_c$; and (ii) $I_{\text{APP}}(j) = 1$ for $j = n_s + n_p + 1, ..., n_s + n_c$. Based on the above, (2.19) to (2.21) can be simplified. Specifically,

we simplify (2.19) to

$$I_{\text{E\_V}_\text{u}\text{C}}(i,j)=\begin{cases}\psi(e_{i,j})J_{\text{BSC}}\Bigg(\sum\limits_{i'\neq i}e_{i',j}[J^{-1}(I_{\text{A\_CV}_\text{u}}(i',j))]^2+ \\ \qquad (e_{i,j}-1)[J^{-1}(I_{\text{A\_CV}_\text{u}}(i,j))]^2,p_1\Bigg), \\ \qquad\qquad\qquad j=1,2,...n_s;\forall i, \\ \psi(e_{i,j})J\Bigg(\sqrt{\sum\limits_{i'\neq i}e_{i',j}[J^{-1}(I_{\text{A\_CV}_\text{u}}(i',j))]^2} \\ \qquad \overline{+(e_{i,j}-1)[J^{-1}(I_{\text{A\_CV}_\text{u}}(i,j))]^2}\Bigg), \\ \qquad\qquad\qquad j=n_s+1,...,n_s+n_p;\forall i,\end{cases} \tag{5.6}$$

because

1. $\sigma^2_{\text{ch}}(j)=0$ ($j=n_s+1,n_s+2,...,n_s+n_p$) for the punctured channel VNs; and
2. $I_{\text{E\_VC}}(i,j)=1$ ($i=1,2,...,m_s+m_c, j=n_s+n_p+1,n_s+n_p+2,...,n_s+n_c$) when the channel information is perfect.

We also simplify (2.20) to

$$I_{\text{E\_CV}_\text{u}}(i,j)=\psi(e_{i,j})\Bigg(1-J\Bigg(\sqrt{\sum_{j'\neq j}e_{i,j'}[J^{-1}(1-I_{\text{A\_V}_\text{u}\text{C}}(i,j'))]^2} \\ \overline{+(e_{i,j}-1)[J^{-1}(1-I_{\text{A\_V}_\text{u}\text{C}}(i,j))]^2}\Bigg)\Bigg), \\ j=1,2,...n_s+n_p;\forall i, \tag{5.7}$$

because

1. $I_{\text{E\_VC}}(i,j)$ ($j=n_s+n_p+1,...,n_s+n_c$ and $\forall i$) are fixed to "1" when the channel information is perfect, and hence $I_{\text{E\_CV}}(i,j)$ ($j=n_s+n_p+1,...,n_s+n_c$ and $\forall i$) needs not to be calculated;
2. $1-I_{\text{A\_VC}}(i,j)=0$ ($j=n_s+n_p+1,...,n_s+n_c$ and $\forall i$) when the channel information is perfect.

Finally, we simplify (2.21) to

$$I_{\mathrm{APP}}(j) = \begin{cases} J_{\mathrm{BSC}}\left(\sum_i e_{i,j}[J^{-1}(I_{\mathrm{A\_CV_u}}(i,j))]^2, p_1\right), \\ \qquad\qquad j = 1, 2, ..., n_s, \\ J\left(\sqrt{\sum_i e_{i,j}[J^{-1}(I_{\mathrm{A\_CV_u}}(i,j))]^2}\right), \\ \qquad\qquad j = n_s + 1, ..., n_s + n_p, \end{cases} \tag{5.8}$$

because $I_{\mathrm{APP}}(j) = 1$ ($j = n_s + n_p + 1, \ldots, n_s + n_c$) when the channel information is perfect.

Algorithm 3 shows the process of obtaining the source threshold p_{th} by using the UP-EXIT algorithm, which is relatively simple compared to the JP-EXIT algorithm. Using (5.6), (5.7), (5.8) instead of (2.19), (2.20), and (2.21), respectively, the whole matrix $\mathbf{B}_{\mathrm{J_{new}}}$ used in JP-EXIT can be reduced to $\mathbf{B}_{\mathrm{u}}$ for UP-EXIT ($\mathbf{B}_{\mathrm{u}}$ is the sub-protomatrix corresponding to the untransmitted VNs).

Algorithm 3 UP-EXIT algorithm.

Given a sub-protomatrix $\mathbf{B}_{\mathrm{u}}$ shown in (5.5), set the maximum number of iterations l_{max}, step size $\hat{p}_1$, and tolerance value θ ($\theta = 10^{-6}$ in this thesis).

Set b_{flag} = true.

Use a sufficiently large $p_1 < 0.5$.

Start of UP-EXIT algorithm

while b_{flag} **do**

Set $I_{\mathrm{E\_V_uC}}(i,j)$=$I_{\mathrm{A\_CV_u}}(i,j)$=$I_{\mathrm{E\_CV_u}}(i,j)$=$I_{\mathrm{A\_V_uC}}(i,j)$= 0 and $I_{\mathrm{APP}}(j) = 0, i = 1, 2, \ldots, m_s+ m_c$, j=1, 2, . . ., $n_s + n_p$.

Set $l = 1$.

while $l \leq l_{max}$ **do**

Update MI from VNs to CNs by calculating (5.6).

Set $I_{\mathrm{A\_V_uC}}(i,j) = I_{\mathrm{E\_V_uC}}(i,j)\ \forall i, j$.

Update MI from CNs to VNs by calculating (5.7).

Set $I_{\mathrm{A\_CV_u}}(i,j) = I_{\mathrm{E\_CV_u}}(i,j)\ \forall i, j$.

Next, calculate $I_{\mathrm{APP}}(j)$ by (5.8).

if $\sum_{j=1}^{n_s+n_p}(1-I_{\mathrm{APP}}(j)) < \theta$ **then**

Set $p_{\mathrm{th}} = p_1$ and $b_{\mathrm{flag}} =$ false.

End of UP-EXIT algorithm

else

Set $l = l + 1$.

end if

end while

if $b_{\mathrm{flag}} =$ true **then**

Set $p_1 = p_1 - \hat{p}_1$.

end if

end while

5.1.3 Code Design and Results

In this section, we construct some JSC-BCs for low-entropy and high-entropy sources. Source thresholds and channel thresholds can be calculated by using the UP-EXIT algorithm and JP-EXIT algorithm in [33], respectively. The UP-EXIT algorithm requires the use of only the sub-protomatrix corresponding to the untransmitted VNs; whereas the JP-EXIT algorithm requires the use of the complete $\mathbf{B}_{\mathrm{J_{new}}}$.

In designing and optimizing $\mathbf{B}_{\mathrm{J_{new}}}$, we apply some existing design rules for DP-LDPC codes [33,34,50] and fix the elements of some columns so as to restrict searching space and to reduce searching time. As a result, we only need to design elements in columns corresponding to the untransmitted VNs in $\mathbf{B}_{\mathrm{J_{new}}}$. **Algorithm 4** shows our code design algorithm. Since the sub-protomatrices used in UP-EXIT are small, it is easier and faster to find candidate codes with high source thresholds in Step 1). Then we can find a code with a low channel threshold among these codes in Step 2).

We compare the following four types of codes.

Algorithm 4 Optimization of JSC-BCs

Step 1) Based on the UP-EXIT algorithm, we further use the differential evolution (DE) algorithm [29] to find high-source-threshold sub-protomatrices corresponding to the untransmitted VNs.

Step 2) Among the candidates found in Step 1), we perform the JP-EXIT algorithm to further find the codes with low channel thresholds. Finally, we select the code with a low channel threshold and a high source threshold.

i) Type-I: The proposed joint source-channel block code shown in (5.1), i.e., $\mathbf{B}_{\mathrm{J_{new}}} = \begin{pmatrix} \mathbf{B}_{\mathrm{s}} & \mathbf{B}'_{\mathrm{sccv}} \\ \mathbf{B}_{\mathrm{svcc}} & \mathbf{B}_{\mathrm{c}} \end{pmatrix}$, where $\mathbf{B}'_{\mathrm{sccv}} = \begin{pmatrix} \mathbf{T}_{m_s} & \mathbf{0}_{m_s \times m_c} \end{pmatrix}$ and $\mathbf{B}_{\mathrm{svcc}}$ is a non-zero protomatrix.

ii) Type-II: The DP-LDPC code $\mathbf{B}_{\mathrm{J}} = \begin{pmatrix} \mathbf{B}_{\mathrm{s}} & \mathbf{B}'_{\mathrm{sccv}} \\ \mathbf{0}_{m_c \times n_s} & \mathbf{B}_{\mathrm{c}} \end{pmatrix}$ proposed in Section 3 and shown in (3.1), where $\mathbf{B}'_{\mathrm{sccv}} = \begin{pmatrix} \mathbf{T}_{m_s} & \mathbf{0}_{m_s \times m_c} \end{pmatrix}$.

iii) Type-III: The DP-LDPC code $\mathbf{B}_{\mathrm{J}_0} = \begin{pmatrix} \mathbf{B}_{\mathrm{s}} & \mathbf{B}_{\mathrm{sccv}} \\ \mathbf{0}_{m_c \times n_s} & \mathbf{B}_{\mathrm{c}} \end{pmatrix}$ introduced in Section 2.2 and shown in (2.4), where $\mathbf{B}_{\mathrm{sccv}} = \begin{pmatrix} \mathbf{I}_{m_s} & \mathbf{0}_{m_s \times m_c} \end{pmatrix}$. $\mathbf{B}_{\mathrm{J3}}^{0.01}$ [35], $\mathbf{B}_{\mathrm{J-new}}^{0.01}$ [36], and ($\mathbf{B}_{\mathrm{s1}}$,$\mathbf{B}_{\mathrm{c1}}$) in [34] are codes with structures as $\mathbf{B}_{\mathrm{J}}$.

iv) Type-IV: The DP-LDPC code $\mathbf{B}_{\mathrm{J}}^{*} = \begin{pmatrix} \mathbf{B}_{\mathrm{s}} & \mathbf{B}_{\mathrm{sccv}} \\ \mathbf{B}_{\mathrm{svcc}} & \mathbf{B}_{\mathrm{c}} \end{pmatrix}$, where $\mathbf{B}_{\mathrm{sccv}} = \begin{pmatrix} \mathbf{I}_{m_s} & \mathbf{0}_{m_s \times m_c} \end{pmatrix}$ and $\mathbf{B}_{\mathrm{svcc}}$ is a non-zero protomatrix. This type of DP-LDPC codes has been introduced in Section 2.2.4 and shown in (2.22). $\mathbf{B}_{\mathrm{J}}^{\mathrm{opti\_4}}$ [37], $\mathbf{B}_{\mathrm{J-new2}}^{0.04}$ [36], ($\mathbf{B}_{\mathrm{J3}}, \mathbf{B}_{\mathrm{1}}^{opt-3}$) [50], $\mathbf{B}_1$ and $\mathbf{B}_2$ in [29], $\mathbf{B}_{\mathrm{J}}^{\mathrm{opti\_3}}$ [37], and $\mathbf{B}_{\mathrm{J-new2}}^{0.04}$ [36] are codes with structures as $\mathbf{B}_{\mathrm{J}}^{*}$. Here, $\mathbf{B}_1$ denotes the code consisting of (25) and (26) in [29]; and (25) and (26) in [29] form $\mathbf{B}_{\mathrm{s}}$ and $\begin{pmatrix} \mathbf{B}_{\mathrm{sccv}} \\ \mathbf{B}_{\mathrm{c}} \end{pmatrix}$ in $\mathbf{B}_{\mathrm{J}}^{*}$, respectively; and the first three VNs in the SVCC linking base matrix of $\mathbf{B}_1$ are of degree-2. Similarly, $\mathbf{B}_2$ denotes the code consisting of (19) and (25) in [29]; and (19) and (25) in [29] form $\mathbf{B}_{\mathrm{s}}$ and $\begin{pmatrix} \mathbf{B}_{\mathrm{sccv}} \\ \mathbf{B}_{\mathrm{c}} \end{pmatrix}$ of $\mathbf{B}_{\mathrm{J}}^{*}$, respectively; and the $\{1, 3, 5, 7\}$-th

VNs in the SVCC linking base matrix of $\mathbf{B}_2$ are of degree-2.

5.1.3.1 Low-entropy sources

In this section, we design some codes for low-entropy sources.

Example #1: We consider the DP-LDPC block code

$$\begin{aligned}\mathbf{B}_{\mathrm{J3\_opt3}}^{0.01} &= \left(\begin{array}{c|c} \mathbf{B}_{\mathrm{s}}^{0.01} & \mathbf{B}_{\mathrm{sccv}}^{0.01} \\ \hline \mathbf{0}_{m_c \times n_s} & \mathbf{B}_{\mathrm{c}}^{0.01} \end{array}\right) \\ &= \left(\begin{array}{cccccccc|ccccc} 1 & 1 & 2 & 1 & 3 & 1 & 3 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 2 & 1 & 2 & 1 & 2 & 1 & 2 & 3 & 1 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 1 & 0 & 1 & 1 \end{array}\right)\end{aligned} \tag{5.9}$$

which is designed at $p_1 = 0.01$ [74] and proposed in Section 3.3. Firstly, we add connections between VNs in $\mathbf{B}_{\mathrm{s}}^{0.01}$ and CNs in $\mathbf{B}_{\mathrm{c}}^{0.01}$. We apply the differential evolution (DE) algorithm [29] (introduced in Section 2.5) based on the JP-EXIT algorithm to optimize these connections, aiming at obtaining a code with a low channel threshold. Additionally, the code obtained should have a higher source threshold than $\mathbf{B}_{\mathrm{J3\_opt3}}^{0.01}$. Subsequently, we obtain

$$\begin{aligned}\mathbf{B}_{\mathrm{J_{new\_0.01}}}^{\mathrm{opt1}} &= \left(\begin{array}{c|c} \mathbf{B}_{\mathrm{s}}^{0.01} & \mathbf{B}_{\mathrm{sccv}}^{0.01} \\ \hline \mathbf{B}_{\mathrm{svcc}}^{0.01} & \mathbf{B}_{\mathrm{c}}^{0.01} \end{array}\right) \\ &= \left(\begin{array}{cccccccc|ccccc} 1 & 1 & 2 & 1 & 3 & 1 & 3 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 2 & 1 & 2 & 1 & 2 & 1 & 2 & 3 & 1 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 3 & 0 & 3 & 0 & 3 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 1 & 0 & 1 & 1 \end{array}\right)\end{aligned}. \tag{5.10}$$

Secondly, we optimize $\mathbf{B}_{\mathrm{J_{new}}}$ using **Algorithm 4**. We aim to find a code with a high source

threshold and a low channel threshold given $p_1 = 0.01$. We fix the elements corresponding to the transmitted channel VNs in $\mathbf{B}_{\mathrm{J_{new}}}$, which include a column with weight-1, two columns with weight-2, and a column with weight-3. Additionally, one VN in $\mathbf{B}_{\mathrm{s}}$ is assigned with weight-2. Subsequently, we obtain

$$\mathbf{B}_{\mathrm{J\_new}}^{0.01} = \left(\begin{array}{cccccccc|ccccc} 1 & x & x & x & x & x & x & x & 1 & 0 & 0 & 0 & 0 \\ 1 & x & x & x & x & x & x & x & x & 1 & 0 & 0 & 0 \\ \hline 0 & x & x & x & x & x & x & x & x & 0 & 1 & 0 & 0 \\ 0 & x & x & x & x & x & x & x & x & 0(1) & 0 & 1 & 1 \\ 0 & x & x & x & x & x & x & x & x & 1(0) & 0 & 1 & 2 \end{array}\right), \tag{5.11}$$

where x's denote the entries to be determined. As can be seen in (5.11), x's are located in all four sub-protomatrices (source protomatrix, channel protomatrix, SCCV linking protomatrix, and SVCC linking protomatrix) of $\mathbf{B}_{\mathrm{J_{new}}}$. The maximum value of x is set to 3. The first VN in the channel code is the punctured VN and its degree should be the largest. As shown in [74], the decoding complexity is related to the maximum row weight. (Unless otherwise stated, we use the joint belief propagation algorithm [37], which has been introduced in Section 2.2.2, to decode the JSC-BC codes.) Based on our observations of existing DP-LDPC codes designed at $p_1 = 0.01$, we set the maximum row weight to 16 as to maintain the same decoding complexity as (5.9). By using **Algorithm 4** to search for x's in (5.11), we obtain

$$\mathbf{B}_{\mathrm{J_{new\_0.01}}}^{\mathrm{opt2}} = \left(\begin{array}{cccccccc|ccccc} 1 & 3 & 3 & 3 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 2 & 2 & 1 & 2 & 2 & 1 & 3 & 1 & 0 & 0 & 0 \\ \hline 0 & 0 & 3 & 2 & 0 & 0 & 0 & 0 & 3 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 2 & 0 & 0 & 1 & 2 \end{array}\right). \tag{5.12}$$

Table 5.1 lists the source and channel thresholds of $\mathbf{B}_{\mathrm{J_{new\_0.01}}}^{\mathrm{opt1}}$, $\mathbf{B}_{\mathrm{J_{new\_0.01}}}^{\mathrm{opt2}}$, and state-of-the-art DP-LDPC block codes designed at $p_1 = 0.01$ [35, 36, 74]. We can see that $\mathbf{B}_{\mathrm{J_{new\_0.01}}}^{\mathrm{opt2}}$ achieves

Table 5.1: Channel thresholds and source thresholds of different codes at $p_1 = 0.01$. Shannon limit equals -12.02 dB.

Code	$\mathbf{B}^{\text{opt1}}_{\text{J}_{\text{new\_0.01}}}$	$\mathbf{B}^{\text{opt2}}_{\text{J}_{\text{new\_0.01}}}$	$\mathbf{B}^{0.01}_{\text{J3\_opt3}}$ [74]	$\mathbf{B}^{0.01}_{\text{J3}}$ [35, Table I]	$\mathbf{B}^{0.01}_{\text{J}-\text{new}}$ [36, Table II]
$(E_s/N_0)_{\text{th}}$ (dB)	-9.787	$\mathbf{-10.170}$	-9.734	-9.324	-9.725
p_{th}	0.041	**0.084**	0.028	0.028	0.028

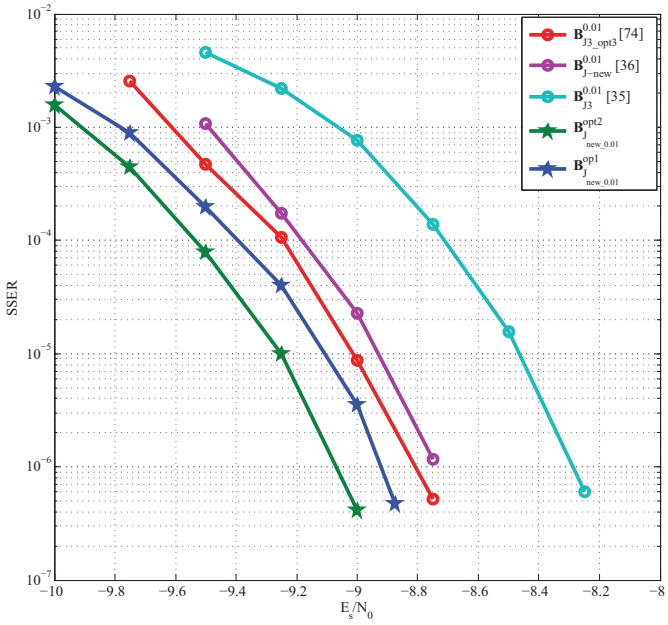


Figure 5.2: SSER performance comparison between $\mathbf{B}^{\text{opt1}}_{\text{J}_{\text{new\_0.01}}}$, $\mathbf{B}^{\text{opt2}}_{\text{J}_{\text{new\_0.01}}}$, and state-of-the-art DP-LDPC block codes at $p_1 = 0.01$. The lifting factor is $z = z_1 z_2 = 4 \times 400 = 1600$.

the lowest channel threshold and the highest source threshold among all codes. Figure 5.2 plots the source symbol error rate (SSER) performance of these codes under different E_s/N_0 (in dB). We can see that $\mathbf{B}^{\text{opt2}}_{\text{J}_{\text{new\_0.01}}}$ has the best error performance among all codes. It has gains of about 0.1 dB and 0.25 dB, respectively, over $\mathbf{B}^{\text{opt1}}_{\text{J}_{\text{new\_0.01}}}$ and $\mathbf{B}^{0.01}_{\text{J3\_opt3}}$ [74] at an SSER of 10^{-6}. $\mathbf{B}^{\text{opt1}}_{\text{J}_{\text{new\_0.01}}}$ also has better error performance than $\mathbf{B}^{0.01}_{\text{J3\_opt3}}$. The SSER results are consistent with the theoretical channel thresholds given in Table 5.1.

Example #2: We consider the code

$$\mathbf{B}_{\mathrm{J\_opt1}}^{0.04} = \left(\begin{array}{c|c} \mathbf{B}_{\mathrm{s}}^{0.04} & \mathbf{B}_{\mathrm{sccv}}^{0.04} \\ \hline \mathbf{0}_{m_c \times n_s} & \mathbf{B}_{\mathrm{c}}^{0.04} \end{array}\right) = \left(\begin{array}{cccc|ccccc} 1 & 1 & 1 & 2 & 1 & 0 & 0 & 0 & 0 \\ 1 & 2 & 2 & 1 & 1 & 1 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 2 & 2 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 2 & 0 & 0 & 1 & 1 \end{array}\right) \tag{5.13}$$

which is designed at $p_1 = 0.04$ [74] and proposed in Section 3.3. Similar to Example #1, we firstly add and optimize connections between VNs in $\mathbf{B}_{\mathrm{s}}^{0.04}$ and CNs in $\mathbf{B}_{\mathrm{c}}^{0.04}$ and obtain

$$\mathbf{B}_{\mathrm{J_{new\_0.04}}}^{\mathrm{opt1}} = \left(\begin{array}{c|c} \mathbf{B}_{\mathrm{s}}^{0.04} & \mathbf{B}_{\mathrm{sccv}}^{0.04} \\ \hline \mathbf{B}_{\mathrm{svcc}}^{0.04} & \mathbf{B}_{\mathrm{c}}^{0.04} \end{array}\right) = \left(\begin{array}{cccc|ccccc} 1 & 1 & 1 & 2 & 1 & 0 & 0 & 0 & 0 \\ 1 & 2 & 2 & 1 & 1 & 1 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 1 & 2 & 2 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 2 & 0 & 0 & 1 & 1 \end{array}\right). \tag{5.14}$$

Secondly, we optimize $\mathbf{B}_{\mathrm{J_{new}}}$ using **Algorithm 4**. We aim to find a code with a high source threshold and a low channel threshold given $p_1 = 0.04$. Similar to Example #1, we fix the elements corresponding to the transmitted channel VNs and the elements corresponding to one

Table 5.2: Channel thresholds and source thresholds of different codes at $p_1 = 0.04$. The Shannon limit equals -7.00 dB.

Code	$\mathbf{B}^{\text{opt1}}_{\text{J}_{\text{new\_0.04}}}$	$\mathbf{B}^{\text{opt2}}_{\text{J}_{\text{new\_0.04}}}$	$\mathbf{B}^{0.04}_{\text{J\_opt1}}$ [74]	$(\mathbf{B}_{\text{s1}},\mathbf{B}_{\text{c1}})$ in [34, Table I]	$\mathbf{B}^{\text{opti\_4}}_{\text{J}}$ [37]	$(\mathbf{B}^{0.04}_{\text{J}-\text{new2}})$ [36, Table I]
$(E_s/N_0)_{\text{th}}$ (dB)	-5.202	$\mathbf{-5.880}$	-5.267	-5.135	-5.568	-5.729
p_{th}	0.137	**0.217**	0.082	0.063	0.137	0.129

source VN, and we obtain

$$\mathbf{B}^{0.04}_{\text{J}_{\text{new}}} = \left(\begin{array}{cccc|ccccc} 1 & x & x & x & 1 & 0 & 0 & 0 & 0 \\ 1 & x & x & x & x & 1 & 0 & 0 & 0 \\ \hline 0 & x & x & x & x & 0 & 1 & 0 & 0 \\ 0 & x & x & x & x & 0(1) & 0 & 1 & 1 \\ 0 & x & x & x & x & 1(0) & 0 & 1 & 2 \end{array}\right). \tag{5.15}$$

Based on our observations of existing DP-LDPC codes designed at $p_1 = 0.04$, we set the maximum row weight to 8 as to maintain the same decoding complexity as (5.13). By using **Algorithm 4** to search for x's in (5.15), we obtain

$$\mathbf{B}^{\text{opt2}}_{\text{J}_{\text{new\_0.04}}} = \left(\begin{array}{cccc|ccccc} 1 & 0 & 2 & 3 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 3 & 1 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 1 & 3 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3 & 0 & 0 & 1 & 1 \\ 0 & 2 & 0 & 0 & 1 & 1 & 0 & 1 & 2 \end{array}\right). \tag{5.16}$$

Table 5.2 lists the source and channel thresholds of $\mathbf{B}^{\text{opt1}}_{\text{J}_{\text{new\_0.04}}}$, $\mathbf{B}^{\text{opt2}}_{\text{J}_{\text{new\_0.04}}}$, and state-of-the-art DP-LDPC block codes designed at $p_1 = 0.04$ [34,36,37,74]. We can see that $\mathbf{B}^{\text{opt2}}_{\text{J}_{\text{new\_0.04}}}$ achieves the lowest channel threshold and the highest source threshold among all codes. Figure 5.3 plots the SSER performance of these codes. We can see that $\mathbf{B}^{\text{opt2}}_{\text{J}_{\text{new\_0.04}}}$ has the best error performance among all codes. It has gains of about 0.1 dB and 0.3 dB, respectively, over $\mathbf{B}^{0.04}_{\text{J}-\text{new2}}$ [36] and $\mathbf{B}^{\text{opti\_4}}_{\text{J}}$ [37] at an SSER of 10^{-6}. Note that $\mathbf{B}^{\text{opt1}}_{\text{J}_{\text{new\_0.04}}}$ has a worse error performance than

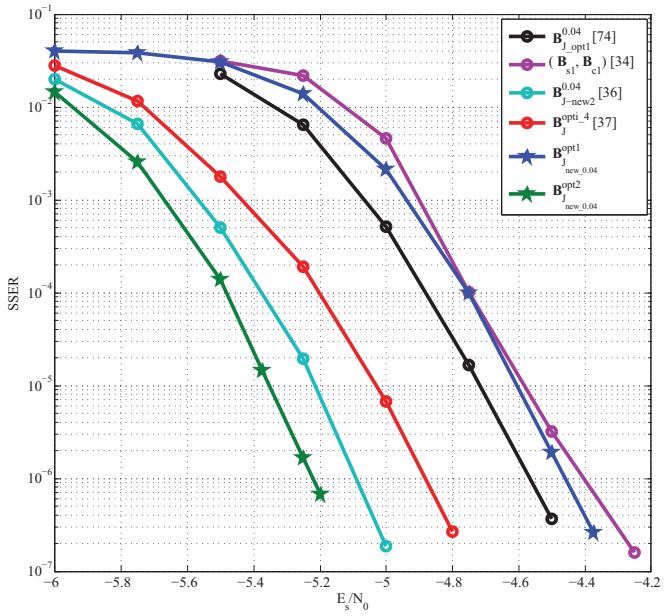


Figure 5.3: SSER performance comparison between $\mathbf{B}^{\text{opt1}}_{\text{J}_{\text{new\_0.04}}}$, $\mathbf{B}^{\text{opt2}}_{\text{J}_{\text{new\_0.04}}}$, and state-of-the-art DP-LDPC block codes at $p_1 = 0.04$. The lifting factor is $z = z_1 z_2 = 4 \times 800 = 3200$.

$\mathbf{B}^{0.04}_{\text{J\_opt1}}$ [74], which is consistent with its theoretical channel threshold being higher than that of $\mathbf{B}^{0.04}_{\text{J\_opt1}}$. The results indicate that by directly adding a non-zero SVCC linking base matrix to an existing DP-LDPC code, the channel threshold and error performance of the resultant code may become worse even though its source threshold becomes higher.

5.1.3.2 High-entropy sources

In this section, we design some codes for high-entropy sources.

Example #3: According to Shannon's coding theory, $RH(p_1) < 1$, where $R = n_s/(n_c - n_p)$ is the overall code rate and $H(p_1) = -p_1 \log_2 p_1 - (1-p_1)\log_2(1-p_1)$ is the source entropy. When $H(p_1)$ is relatively large, R needs to be relatively small. In this section, we set $p_1 = 0.10, 0.20$

and $R = 1$. Firstly, we optimize the SVCC linking base matrix based on (5.13) and obtain

$$\mathbf{B}^{\text{opt1}}_{\text{J}_{\text{new\_0.10}}} = \left(\begin{array}{cccc|ccccc} 1 & 1 & 1 & 2 & 1 & 0 & 0 & 0 & 0 \\ 1 & 2 & 2 & 1 & 1 & 1 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 2 & 2 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 2 & 0 & 0 & 1 & 1 \end{array}\right) \tag{5.17}$$

and

$$\mathbf{B}^{\text{opt1}}_{\text{J}_{\text{new\_0.20}}} = \left(\begin{array}{cccc|ccccc} 1 & 1 & 1 & 2 & 1 & 0 & 0 & 0 & 0 \\ 1 & 2 & 2 & 1 & 1 & 1 & 0 & 0 & 0 \\ \hline 0 & 1 & 1 & 0 & 2 & 2 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 2 & 0 & 0 & 1 & 1 \end{array}\right). \tag{5.18}$$

for $p_1 = 0.10$ and $p_1 = 0.20$, respectively.

Next, by using **Algorithm 4**, we search for x's in (5.15) given $p_1 = 0.10$ and 0.20, respectively, and obtain

$$\mathbf{B}^{\text{opt2}}_{\text{J}_{\text{new\_0.10}}} = \left(\begin{array}{cccc|ccccc} 1 & 0 & 3 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 2 & 1 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 3 & 0 & 1 & 0 & 0 \\ 0 & 2 & 0 & 1 & 2 & 0 & 0 & 1 & 1 \\ 0 & 0 & 2 & 1 & 1 & 1 & 0 & 1 & 2 \end{array}\right) \tag{5.19}$$

and

$$\mathbf{B}^{\text{opt2}}_{\text{J}_{\text{new\_0.20}}} = \left(\begin{array}{cccc|ccccc} 1 & 0 & 2 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 3 & 1 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 3 & 0 & 1 & 0 & 0 \\ 0 & 2 & 0 & 1 & 2 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 2 \end{array}\right). \tag{5.20}$$

Table 5.3: Channel thresholds and source thresholds of different codes at $p_1 = 0.10$ and $p_1 = 0.20$.

p_1	0.10						0.20		
Code	$\mathbf{B}_{\mathrm{J}_{\mathrm{new\_0.10}}}^{\mathrm{opt1}}$	$\mathbf{B}_{\mathrm{J}_{\mathrm{new\_0.10}}}^{\mathrm{opt2}}$	$(\mathbf{B}_{\mathrm{J}3}, \mathbf{B}_{1}^{opt-3})$ [50]	$\mathbf{B}_1$ [29]	$\mathbf{B}_{\mathrm{J}}^{\mathrm{opti\_3}}$ [37]	$(\mathbf{B}_{\mathrm{J-new2}}^{0.04})$ [36, Table I]	$\mathbf{B}_{\mathrm{J}_{\mathrm{new\_0.20}}}^{\mathrm{opt1}}$	$\mathbf{B}_{\mathrm{J}_{\mathrm{new\_0.20}}}^{\mathrm{opt2}}$	$\mathbf{B}_2$ [29]
$(E_s/N_0)_{\mathrm{th}}$ (dB)	-2.333	$\mathbf{-2.595}$	-1.556	-1.581	-1.996	-1.218	1.048	**0.831**	2.322
p_{th}	0.137	**0.171**	0.159	0.160	0.144	0.129	**0.334**	0.288	0.283
Shannon limit (dB)	-3.392						-0.654		

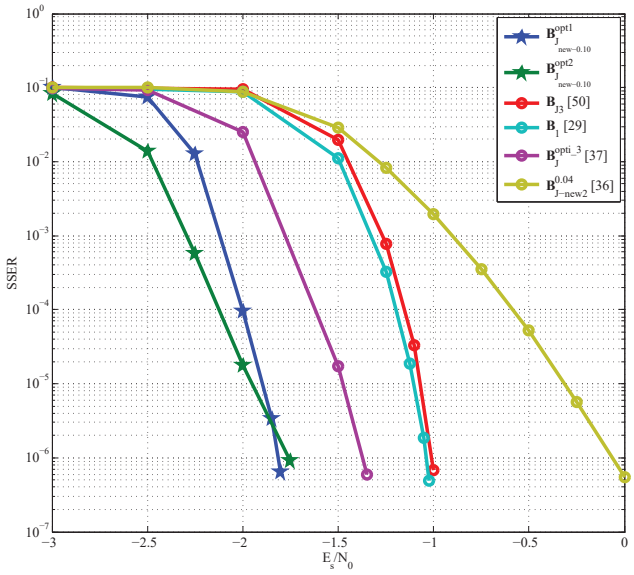


Figure 5.4: SSER performance comparison between $\mathbf{B}_{\mathrm{J}_{\mathrm{new\_0.10}}}^{\mathrm{opt1}}$, $\mathbf{B}_{\mathrm{J}_{\mathrm{new\_0.10}}}^{\mathrm{opt2}}$, and existing DP-LDPC block codes at $p_1 = 0.10$. The lifting factor is $z = z_1 z_2 = 4 \times 800 = 3200$.

Table 5.3 lists the source and channel thresholds of $\mathbf{B}_{\mathrm{J}_{\mathrm{new\_0.10}}}^{\mathrm{opt1}}$, $\mathbf{B}_{\mathrm{J}_{\mathrm{new\_0.10}}}^{\mathrm{opt2}}$, and existing DP-LDPC block codes with non-zero SVCC linking base matrices [29, 36, 37, 50] at $p_1 = 0.10$. It can be seen that given $p_1 = 0.10$, $\mathbf{B}_{\mathrm{J}_{\mathrm{new\_0.10}}}^{\mathrm{opt2}}$ have lower channel thresholds than existing DP-LDPC codes. Figure 5.4 plots the SSER performance of these codes. The two new codes $\mathbf{B}_{\mathrm{J}_{\mathrm{new\_0.10}}}^{\mathrm{opt1}}$ and $\mathbf{B}_{\mathrm{J}_{\mathrm{new\_0.10}}}^{\mathrm{opt2}}$ have a maximum gain of around 1.7 dB and a minimum gain of about 0.5 dB over the existing codes at an SSER of 10^{-6}. The SSER results are also consistent with the theoretical thresholds in Table 5.3.

Table 5.3 also lists the source and channel thresholds of $\mathbf{B}_{\mathrm{J}_{\mathrm{new\_0.20}}}^{\mathrm{opt1}}$, $\mathbf{B}_{\mathrm{J}_{\mathrm{new\_0.20}}}^{\mathrm{opt2}}$, and $\mathbf{B}_2$ (a DP-LDPC block code with a non-zero SVCC linking base matrix in [29]) at $p_1 = 0.20$. It can be seen that given $p_1 = 0.20$, $\mathbf{B}_{\mathrm{J}_{\mathrm{new\_0.20}}}^{\mathrm{opt1}}$ and $\mathbf{B}_{\mathrm{J}_{\mathrm{new\_0.20}}}^{\mathrm{opt2}}$ achieve lower channel thresholds than $\mathbf{B}_2$. Figure 5.5 plots the SSER performance of these codes. The new codes $\mathbf{B}_{\mathrm{J}_{\mathrm{new\_0.20}}}^{\mathrm{opt1}}$ and $\mathbf{B}_{\mathrm{J}_{\mathrm{new\_0.20}}}^{\mathrm{opt2}}$ significantly outperform the existing one $\mathbf{B}_2$. Moreover, $\mathbf{B}_{\mathrm{J}_{\mathrm{new\_0.20}}}^{\mathrm{opt2}}$ has a gain of more than 0.15

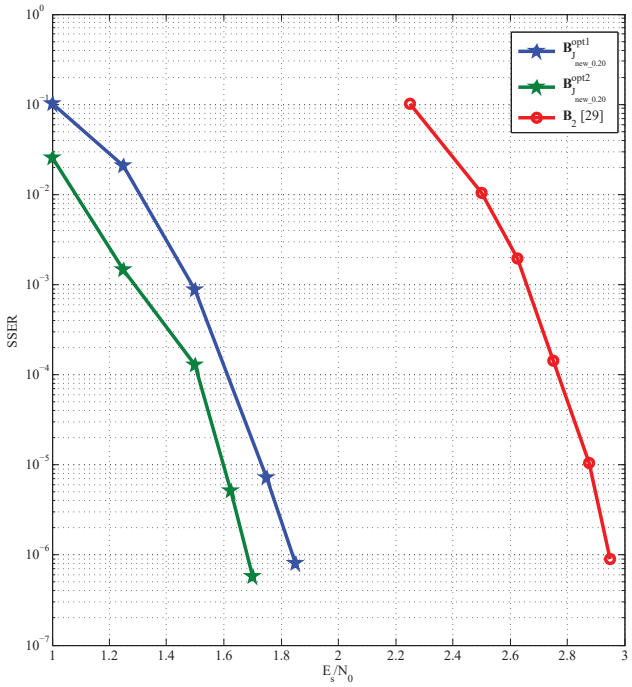


Figure 5.5: SSER performance comparison between $\mathbf{B}^{\text{opt1}}_{\text{J}_{\text{new\_0.20}}}$, $\mathbf{B}^{\text{opt2}}_{\text{J}_{\text{new\_0.20}}}$, and an existing DP-LDPC block code $\mathbf{B}_2$ at $p_1 = 0.20$. The lifting factor is $z = z_1 z_2 = 4 \times 800 = 3200$.

dB over $\mathbf{B}^{\text{opt1}}_{\text{J}_{\text{new\_0.20}}}$ at an SSER of 10^{-6}.

Based on the examples above, we can observe that our proposed codes outperform the other three types of codes. It is because the design of our proposed code structure is more flexible and is therefore more likely to find a code having good thresholds and good error performance.

5.2 Protograph-based spatially-coupled joint source-channel code

In [47], a source spatially coupled protograph-based LDPC (SC-P-LDPC) code and a channel SC-P-LDPC code are concatenated, which has been introduced in Section 2.4. In [76], a source SC-P-LDPC code and a channel SC-P-LDPC code are spatially coupled by using a spatially-coupled SCCV (SC-SCCV) linking (base) matrix, which is proposed in Chapter 4; in other words, the source SC-P-LDPC code and the channel SC-P-LDPC code are not connected in a simple cascading relationship. In the section, we propose a new type of spatially coupled joint

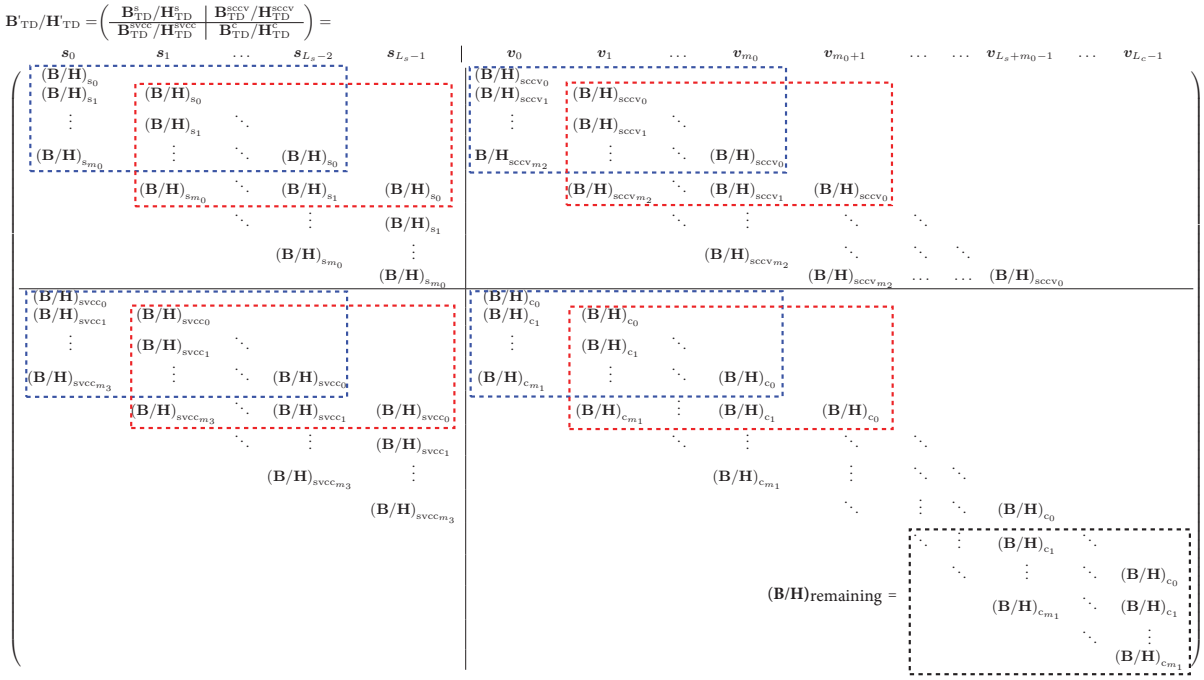


Figure 5.6: Protomatrix and parity-check matrix of proposed spatially coupled joint source-channel codes are denoted by $\mathbf{B}_{\mathrm{TD}}$ and $\mathbf{H}_{\mathrm{TD}}$, respectively. When L_s and L_c are infinite, the corresponding code is called spatially coupled joint source-channel convolutional code

source-channel code (SC-JSCC). In the proposed SC-JSCC, a source SC-P-LDPC code and a channel SC-P-LDPC code are not only spatially coupled by a SC-SCCV linking (base) matrix, but also by a spatially-coupled SVCC (SC-SVCC) linking (base) matrix. By doing this, we aim to improve the source threshold without changing the source compression rate or increasing the syndrome former memory of the source SC-P-LDPC proposed in [47].

Figure 5.6 shows the protomatrix of the proposed SC-JSCC $\mathbf{B}_{\mathrm{TD}}$ and the corresponding parity-check matrix $\mathbf{H}_{\mathrm{TD}}$ formed by lifting $\mathbf{B}_{\mathrm{TD}}$ twice. m_0, m_1, m_2, and m_3, respectively, represent the syndrome former memories of the source SC-P-LDPC code, the channel SC-P-LDPC code, the SC-SVCC linking matrix, and the SC-SVCC linking matrix. L_s and L_c, respectively, denote the coupling lengths of the source and channel SC-P-LDPC codes. When L_s and L_c are finite, the corresponding code is called spatially-coupled joint source-channel (SC-JSC) terminated code. When both L_s and L_c tend to infinity, the SC-JSCC becomes a spatially-coupled joint source-channel convolutional code (SC-JSC-CC). For the SC-JSC-CC, its overall code rate $R_{\mathrm{CC}} = R$, where R is the overall code rate of its corresponding block code.

For the SC-JSC terminated code, $R_{\mathrm{TD}} = L_s/L_c \cdot R_{\mathrm{CC}} < R_{\mathrm{CC}}$.

The sub-base matrices $\mathbf{B}_{\mathrm{s}_i}$ $(i = 0, 1, \ldots, m_0)$, $\mathbf{B}_{\mathrm{c}_i}$ $(i = 0, 1, \ldots, m_1)$, and $\mathbf{B}_{\mathrm{svcc}_i}$ $(i = 0, 1, \ldots, m_3)$ can, respectively, be constructed from $\mathbf{B}_{\mathrm{s}}$, $\mathbf{B}_{\mathrm{c}}$, and $\mathbf{B}_{\mathrm{svcc}}$ in $\mathbf{B}_{\mathrm{J_{new}}}$ (5.1). Moreover, they need to satisfy $\sum_{i=0}^{m_0} \mathbf{B}_{\mathrm{s}_i} = \mathbf{B}_{\mathrm{s}}$, $\sum_{i=0}^{m_1} \mathbf{B}_{\mathrm{c}_i} = \mathbf{B}_{\mathrm{c}}$, and $\sum_{i=0}^{m_3} \mathbf{B}_{\mathrm{svcc}_i} = \mathbf{B}_{\mathrm{svcc}}$ [70]. We further require $\mathbf{B}_{\mathrm{c}_0} = \begin{pmatrix} \mathbf{B}_{m_c \times m_s} & \mathbf{T}_{m_c} \end{pmatrix}$, where $\mathbf{B}_{m_c \times m_s}$ is a base matrix of size $m_c \times m_s$ and $\mathbf{T}_{m_c}$ of size $m_c \times m_c$ denotes a lower or upper triangular protomatrix with "1"s on its diagonal such that linear encoding can be implemented. For the SC-SCCV linking matrix, we require $\mathbf{B}_{\mathrm{sccv}_0}$ to have the same structure as $\mathbf{B}'_{\mathrm{sccv}}$ (i.e., $= \begin{pmatrix} \mathbf{T}_{m_s} & \mathbf{0} \end{pmatrix}$) in $\mathbf{B}_{\mathrm{J_{new}}}$ (5.1) so as to allow linear source compression. We can divide the proposed SC-JSCC into two categories based on the sub-matrices of the SC-SCCV linking matrix:

1. Type-I: All $\mathbf{B}_{\mathrm{sccv}_i}$ $(i = 1, 2, ..., m_2)$ are zero matrices;

2. Type-II: Not all $\mathbf{B}_{\mathrm{sccv}_i}$ $(i = 1, 2, ..., m_2)$ are zero matrices.

Remark: For Type-II SC-JSCCs, the corresponding DP-LDPC block codes may not exist. Given a set of $\mathbf{B}_{\mathrm{sccv}_i}$ $(i = 0, 1, 2, ..., m_2)$ where $\mathbf{B}_{\mathrm{sccv}_0} = \begin{pmatrix} \mathbf{T}_{m_s} & \mathbf{0} \end{pmatrix}$ and not all $\mathbf{B}_{\mathrm{sccv}_i}$ $(i = 1, 2, ..., m_2)$ are zero matrices, $\sum_{i=0}^{m_2} \mathbf{B}_{\mathrm{sccv}_i}$ may not obey the structure of $\mathbf{B}'_{\mathrm{sccv}}$ in $\mathbf{B}_{\mathrm{J_{new}}}$ (5.1), i.e., consisting of a lower or upper triangular protomatrix of size $m_s \times m_s$ with "1"s on the diagonal and a zero matrix. Therefore, we do not base on $\mathbf{B}'_{\mathrm{sccv}}$ to construct the SC-SCCV linking base matrix for Type-II SC-JSCCs.

5.2.1 Encoder

At time t $(= 0, 1, 2, \ldots)$, the source sequence of size $1 \times n_s z$ (z is the overall lifting factor) is denoted by $\boldsymbol{s}_t$. The compressed source sequence of size $1 \times m_s z$ is denoted by $\boldsymbol{c}_t$ at time t. The parity-check bit sequence of size $1 \times m_c z$ for channel encoder is denoted by $\boldsymbol{p}_t$ at time t. The channel codeword sequence of size $1 \times n_c z$ is denoted by $\boldsymbol{v}_t$ at time t. We assume $m_0 = m_1 = m_2 = m_3$ to simplify the explanation.

At $t = 0$, we generate the codeword $\boldsymbol{v}_0$ based on the input $\boldsymbol{s}_0$ and the parity-check matrix

$$\left(\begin{array}{c|c} \boldsymbol{s}_0 & \boldsymbol{v}_0 = [\boldsymbol{c}_0 \ \ \boldsymbol{p}_0] \\ \hline \begin{array}{c|c} \mathbf{H}_{\mathrm{s}_0} & \mathbf{H}_{\mathrm{sccv}_0} \\ \hline \mathbf{H}_{\mathrm{svcc}_0} & \mathbf{H}_{\mathrm{c}_0} \end{array} & \end{array}\right), \tag{5.21}$$

where $\mathbf{H}_{\mathrm{s}_0}$, $\mathbf{H}_{\mathrm{c}_0}$, $\mathbf{H}_{\mathrm{sccv}_0}$, and $\mathbf{H}_{\mathrm{svcc}_0}$ are, respectively, on the 1st block rows of $\mathbf{H}_{\mathrm{TD}}^{\mathrm{s}}$, $\mathbf{H}_{\mathrm{TD}}^{\mathrm{c}}$, $\mathbf{H}_{\mathrm{TD}}^{\mathrm{sccv}}$, and $\mathbf{H}_{\mathrm{TD}}^{\mathrm{svcc}}$ in Fig. 5.6. Since $\mathbf{H}_{\mathrm{sccv}_0}$ is formed by lifting $\mathbf{B}_{\mathrm{sccv}_0} = \begin{pmatrix} \mathbf{T}_{m_s} & \mathbf{0} \end{pmatrix}$ twice, we can obtain $\boldsymbol{v}_0$ by using a similar linear encoding method as described for the JSC-BC in Section 5.1.1. Firstly, $\boldsymbol{c}_0$ is computed based on $\boldsymbol{s}_0$ and $(\mathbf{H}_{\mathrm{s}_0} \mid \mathbf{H}_{\mathrm{sccv}_0})$ by using the linear source compression [74]. Then, $\boldsymbol{p}_0$ is computed based on (i) $\boldsymbol{c}_0$; (ii) the part of $\boldsymbol{s}_0$ corresponding to non-zero columns of $\mathbf{H}_{\mathrm{svcc}_0}$; and (iii) $(\mathbf{H}_{\mathrm{svcc}_0} \mid \mathbf{H}_{\mathrm{c}_0})$ by using linear encoding method. (Recall that $\mathbf{H}_{\mathrm{c}_0}$ is formed by lifting $\mathbf{B}_{\mathrm{c}_0} = \begin{pmatrix} \mathbf{B}_{m_c \times m_s} & \mathbf{T}_{m_c} \end{pmatrix}$ twice.)

At $0 < t < m_0$, the source sequence $\boldsymbol{s}_t$ is input into the encoder. $\boldsymbol{v}_t$ is generated based on the known $\boldsymbol{s}_0, \ldots, \boldsymbol{s}_t$ and $\boldsymbol{v}_0, \ldots, \boldsymbol{v}_{t-1}$ and the following parity-check matrix

$$\left(\begin{array}{cccc|ccc|c} \boldsymbol{s}_0 & \cdots & \boldsymbol{s}_{t-1} & \boldsymbol{s}_t & \boldsymbol{v}_0 & \cdots & \boldsymbol{v}_{t-1} & \boldsymbol{v}_t = [\boldsymbol{c}_t \ \ \boldsymbol{p}_t] \\ \hline \mathbf{H}_{\mathrm{s}_t} & \cdots & \mathbf{H}_{\mathrm{s}_1} & \mathbf{H}_{\mathrm{s}_0} & \mathbf{H}_{\mathrm{sccv}_t} & \cdots & \mathbf{H}_{\mathrm{sccv}_1} & \mathbf{H}_{\mathrm{sccv}_0} \\ \mathbf{H}_{\mathrm{svcc}_t} & \cdots & \mathbf{H}_{\mathrm{svcc}_1} & \mathbf{H}_{\mathrm{svcc}_0} & \mathbf{H}_{\mathrm{c}_t} & \cdots & \mathbf{H}_{\mathrm{c}_1} & \mathbf{H}_{\mathrm{c}_0} \end{array}\right), \tag{5.22}$$

where $\mathbf{H}_{\mathrm{s}_i}$, $\mathbf{H}_{\mathrm{c}_i}$, $\mathbf{H}_{\mathrm{sccv}_i}$, and $\mathbf{H}_{\mathrm{svcc}_i}$ $(i = 0, 1, \ldots, t)$ are, respectively, on the $(t+1)$-th block rows of $\mathbf{H}_{\mathrm{TD}}^{\mathrm{s}}$, $\mathbf{H}_{\mathrm{TD}}^{\mathrm{c}}$, $\mathbf{H}_{\mathrm{TD}}^{\mathrm{sccv}}$, and $\mathbf{H}_{\mathrm{TD}}^{\mathrm{svcc}}$ in Fig. 5.6. Specifically, we first obtain the compressed source symbols $\boldsymbol{c}_t$ based on (i) $\boldsymbol{s}_0, \ldots, \boldsymbol{s}_t$, (ii) $\boldsymbol{v}_0, \ldots, \boldsymbol{v}_{t-1}$, and (iii) the first block row of (5.22) using linear encoding. Then, we generate $\boldsymbol{p}_t$ based on (i) $\boldsymbol{s}_0, \ldots, \boldsymbol{s}_t$, (ii) $\boldsymbol{v}_0, \ldots, \boldsymbol{v}_{t-1}$, (iii) $\boldsymbol{c}_t$ and (iv) the second block row of (5.22) using linear encoding.

At $t \geq m_0$ ($m_0 \leq t \leq L_s - 1$ for SC-JSCCs), the source sequence $\boldsymbol{s}_t$ is input into the encoder. $\boldsymbol{v}_t$ is generated based on the known $\boldsymbol{s}_{t-m_0}, \ldots, \boldsymbol{s}_t$ and $\boldsymbol{v}_{t-m_0}, \ldots, \boldsymbol{v}_{t-1}$ and the following

parity-check matrix

$$\left(\begin{array}{cccc|ccc|c} \boldsymbol{s}_{t-m_0} & \cdots & \boldsymbol{s}_{t-1} & \boldsymbol{s}_t & \boldsymbol{v}_{t-m_0} & \cdots & \boldsymbol{v}_{t-1} & \boldsymbol{v}_t=[\boldsymbol{c}_t\ \ \boldsymbol{p}_t] \\ \hline \mathbf{H}_{\mathrm{s}_{m_0}} & \cdots & \mathbf{H}_{\mathrm{s}_1} & \mathbf{H}_{\mathrm{s}_0} & \mathbf{H}_{\mathrm{sccv}_{m_2}} & \cdots & \mathbf{H}_{\mathrm{sccv}_1} & \mathbf{H}_{\mathrm{sccv}_0} \\ \hline \mathbf{H}_{\mathrm{svcc}_{m_3}} & \cdots & \mathbf{H}_{\mathrm{svcc}_1} & \mathbf{H}_{\mathrm{svcc}_0} & \mathbf{H}_{\mathrm{c}_{m_1}} & \cdots & \mathbf{H}_{\mathrm{c}_1} & \mathbf{H}_{\mathrm{c}_0} \end{array}\right), \tag{5.23}$$

where $\mathbf{H}_{\mathrm{s}_i}$, $\mathbf{H}_{\mathrm{c}_i}$, $\mathbf{H}_{\mathrm{sccv}_i}$, and $\mathbf{H}_{\mathrm{svcc}_i}$ $(i = 0, 1, \ldots, m_0$ and $m_0 = m_1 = m_2 = m_3)$ are, respectively, on the $(t+1)$-th block rows of $\mathbf{H}^{\mathrm{s}}_{\mathrm{TD}}$, $\mathbf{H}^{\mathrm{c}}_{\mathrm{TD}}$, $\mathbf{H}^{\mathrm{sccv}}_{\mathrm{TD}}$, and $\mathbf{H}^{\mathrm{svcc}}_{\mathrm{TD}}$ in Fig. 5.6.

When SC-JSCCs are terminated, $\boldsymbol{v}_{L_s}$, ..., $\boldsymbol{v}_{L_c}$ are generated without any new inputs.

1. Firstly, we continue to generate the channel codewords $\boldsymbol{v}_{L_s}$, ..., $\boldsymbol{v}_{L_s+m_0-1}$. Specifically, we generate $\boldsymbol{v}_{L_s-1+j}$ $(j = 1, ..., m_0)$ based on the known $\boldsymbol{s}_{L_s-m_0-1+j}$, ..., $\boldsymbol{s}_{L_s-1}$, $\boldsymbol{v}_{L_s-m_2-1+j}$, ..., $\boldsymbol{v}_{L_s-2+j}$ and the following parity-check matrix

$$\left(\begin{array}{ccc|ccc|c} \boldsymbol{s}_{L_s-m_0-1+j} & \cdots & \boldsymbol{s}_{L_s-1} & \boldsymbol{v}_{L_s-m_2-1+j} & \cdots & \boldsymbol{v}_{L_s-2+j} & \boldsymbol{v}_{L_s-1+j} \\ \hline \mathbf{H}_{\mathrm{s}_{m_0}} & \cdots & \mathbf{H}_{\mathrm{s}_j} & \mathbf{H}_{\mathrm{sccv}_{m_2}} & \cdots & \mathbf{H}_{\mathrm{sccv}_1} & \mathbf{H}_{\mathrm{sccv}_0} \\ \hline \mathbf{H}_{\mathrm{svcc}_{m_3}} & \cdots & \mathbf{H}_{\mathrm{svcc}_j} & \mathbf{H}_{\mathrm{c}_{m_1}} & \cdots & \mathbf{H}_{\mathrm{c}_1} & \mathbf{H}_{\mathrm{c}_0} \end{array}\right), \tag{5.24}$$

where $(\mathbf{H}_{\mathrm{s}_{m_0}}, ..., \mathbf{H}_{\mathrm{s}_j})$, $(\mathbf{H}_{\mathrm{svcc}_{m_3}}, ..., \mathbf{H}_{\mathrm{svcc}_j})$, $\mathbf{H}_{\mathrm{c}_i}$, and $\mathbf{H}_{\mathrm{sccv}_i}$ $(i = 0, ..., m_0; m_0 = m_1 = m_2 = m_3)$ are, respectively, on the (L_s+j)-th block rows of $\mathbf{H}^{\mathrm{s}}_{\mathrm{TD}}$, $\mathbf{H}^{\mathrm{svcc}}_{\mathrm{TD}}$, $\mathbf{H}^{\mathrm{c}}_{\mathrm{TD}}$, and $\mathbf{H}^{\mathrm{sccv}}_{\mathrm{TD}}$ in Fig. 5.6.

2. Secondly, extra channel codeword sequences $\boldsymbol{v}_{L_s+m_0}$, $\boldsymbol{v}_{L_s+m_0+1}$, $\cdots$, $\boldsymbol{v}_{L_c}$ (consisting of mainly parity-check bits) are derived based on the remaining part of $\mathbf{H}^{\mathrm{c}}_{\mathrm{TD}}$ shown in the

black dashed frame of Fig. 5.6, i.e.,

$$(\mathbf{B}/\mathbf{H})_{\text{remaining}} = \left(\begin{array}{ccc|ccc} & & & \boldsymbol{v}_{L_s+m_0-m_1} & \cdots & \boldsymbol{v}_{L_s+m_0-1} \\ (\mathbf{B}/\mathbf{H})_{c_{m_1}} & \cdots & (\mathbf{B}/\mathbf{H})_{c_1} & (\mathbf{B}/\mathbf{H})_{c_0} & & \\ & \ddots & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & \ddots & (\mathbf{B}/\mathbf{H})_{c_0} \\ & & & \ddots & \ddots & (\mathbf{B}/\mathbf{H})_{c_1} \\ & & & & \ddots & \vdots \\ & & & & & (\mathbf{B}/\mathbf{H})_{c_{m_1}} \end{array}\right). \tag{5.25}$$

We define L_{extra} as the number of block columns added after the block column corresponding to $\boldsymbol{v}_{L_s+m_0-1}$ (i.e., after the (L_s+m_0)-th block column of $\mathbf{H}_{\text{TD}}^{\text{c}}$). To ensure that encoding could be performed based on $\mathbf{H}_{\text{remaining}}$ and the known $\boldsymbol{v}_{L_s+m_0-m_1}$, $\boldsymbol{v}_{L_s+m_0-m_1+1}, \cdots, \boldsymbol{v}_{L_s+m_0-1}$, the number of parity-check equations in $\mathbf{H}_{\text{remaining}}$ should be no more than the number of variable nodes in $\boldsymbol{v}_{L_s+m_0}, \boldsymbol{v}_{L_s+m_0+1}, \cdots, \boldsymbol{v}_{L_c-1}$, i.e.,

$$m_c(L_{\text{extra}}+m_1) - m_{\text{all-zero}} \leq n_c L_{\text{extra}}, \tag{5.26}$$

where $m_{\text{all-zero}}$ denotes the number of all-zero block rows in $\mathbf{B}_{\text{remaining}}$, which denotes the remaining part of $\mathbf{B}_{\text{TD}}^{\text{c}}$ shown in the black dashed frame of Fig. 5.6, i.e., (5.25). In other words, L_{extra} should be the smallest integer selected such that (5.26) is satisfied.

Based on L_{extra} obtained by (5.26), we can calculate the channel coupling length by $L_c = L_s + m_0 + L_{\text{extra}}$.

5.2.2 Sliding window-based decoder and threshold analysis

We apply a sliding window-based joint belief propagation (BP) algorithm to decode the proposed SC-JSCCs and use w to denote the window size. As the blue dashed frames shown in Fig. 5.6, w block rows and block columns of $\mathbf{H}_{\text{TD}}^{\text{s}}$, $\mathbf{H}_{\text{TD}}^{\text{c}}$, $\mathbf{H}_{\text{TD}}^{\text{sccv}}$, and $\mathbf{H}_{\text{TD}}^{\text{svcc}}$ are included in a window. We can regard the protomatrix in a window as a JSC-BC and use the joint BP algo-

rithm [37], which is introduced in Section 2.2.2 to decode the first $n_s z$ source symbols in the window, which are defined as the target symbols. In the next decoding timeslot, the window will slide to the right and downward, i.e., moving from the blue dashed frames to the red dashed frames in Fig. 5.6, and so on. Moreover, all updated log-likelihood ratio (LLR) messages and previously decoded source symbols and channel codewords would be used to facilitate decoding the source symbols and channel codewords in the current window.

As mentioned above, we can regard the protomatrix in a window (w block rows and block columns of $\mathbf{B}_{\mathrm{TD}}^{\mathrm{s}}$, $\mathbf{B}_{\mathrm{TD}}^{\mathrm{c}}$, $\mathbf{B}_{\mathrm{TD}}^{\mathrm{sccv}}$, and $\mathbf{B}_{\mathrm{TD}}^{\mathrm{svcc}}$ in the blue dashed frames in Fig. 5.6) as a JSC-BC. We therefore can use algorithms for calculating the channel and source thresholds of a JSC-BC to calculate the channel and source thresholds of a SC-JSCC, i.e., the JP-EXIT algorithm [33] and UP-EXIT algorithm, respectively. When the MI between the APP-LLR of the first n_s VNs in a window and their corresponding symbols reaches "1" or the maximum number of iterations is reached, the algorithms will stop.

5.2.3 Results and discussions

In this section, we construct some SC-JSCCs from JSC-BCs using the differential evolution (DE) algorithm [29]. When calculating the thresholds for SC-JSCCs, JSC-BCs, and SC-DP-LDPC codes [76], we set the maximum number of iterations to 200.

When simulating the SSER performance of SC-JSCCs and JSC-BCs, we assume that both decoders have the same decoding latency, i.e., both the block decoder and the window decoder need to receive the same number of channel inputs before starting the decoding process [47]. We set the maximum number of iterations for JSC-BCs to 200 while using the syndrome checking to terminate the iteration early. In the decoding of SC-JSCCs and SC-DP-LDPC codes, the iterations will not be stopped until the maximum number of iterations is reached. Thus, we set the maximum number of iterations to 100 for SC-JSCCs and SC-DP-LDPC codes in order to reduce the simulation time. Note that adjusting the maximum number of iterations from 100 to 200 for SC-JSCCs and SC-DP-LDPC codes may result in improved error performance compared to the simulation results presented in this section.

5.2.3.1 Low-entropy sources

We design some SC-JSCCs for low-entropy sources.

Example #4: When $p_1 = 0.01$, we first construct a SC-JSCC based on $\mathbf{B}^{\text{opt1}}_{\text{J}_{\text{new\_0.01}}}$ (5.10). We start by setting $m_0 = m_1 = m_3 = 1$ and $m_2 = 0$. The source SC-P-LDPC protomatrix and the channel SC-P-LDPC protomatrix are to be the same as the SC-DP-LDPC code $\mathbf{B}^{0.01}_{\text{TD}_{\text{new}}}$ [76] proposed in Section 4.5, which are obtained based on $\mathbf{B}^{0.01}_{\text{J3\_opt3}}$ (5.9). We thus obtain

$$\mathbf{B}^{0.01}_{\text{s}'_1} = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 0 & 2 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 \end{pmatrix} ; \mathbf{B}^{0.01}_{\text{s}'_0} = \begin{pmatrix} 1 & 1 & 2 & 0 & 2 & 1 & 1 & 0 \\ 1 & 2 & 1 & 1 & 0 & 1 & 1 & 1 \end{pmatrix} ;$$

$$\mathbf{B}^{0.01}_{\text{c}'_1} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 \\ 2 & 0 & 0 & 1 & 0 \end{pmatrix} ; \mathbf{B}^{0.01}_{\text{c}'_0} = \begin{pmatrix} 3 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{pmatrix} ; \mathbf{B}^{0.01}_{\text{sccv}'_0} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 3 & 1 & 0 & 0 & 0 \end{pmatrix} ; \tag{5.27}$$

$$\mathbf{B}^{0.01}_{\text{svcc}'_1} = \begin{pmatrix} 0 & 0 & 0 & 0 & 2 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} ; \mathbf{B}^{0.01}_{\text{svcc}'_0} = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} ;$$

and we denote the corresponding SC-JSCC by $\mathbf{B}^{0.01}_{\text{TD}_{11}}$. Next, we set $m_2 = 1$ and use the DE algorithm (introduced in Section 2.5) to construct the SC-SCCV sub-protomatrices based on $\mathbf{B}^{0.01}_{\text{s}'_i}$, $\mathbf{B}^{0.01}_{\text{c}'_i}$, and $\mathbf{B}^{0.01}_{\text{svcc}'_i}$ $(i = 0, 1)$ shown in (5.27) to form a new SC-JSCC. We set the maximum row weight of the SC-JSCC no larger than that of $\mathbf{B}^{\text{opt1}}_{\text{J}_{\text{new\_0.01}}}$ to maintain the same decoding complexity as $\mathbf{B}^{\text{opt1}}_{\text{J}_{\text{new\_0.01}}}$. The SC-SCCV linking sub-protomatrices obtained are

$$\mathbf{B}^{0.01}_{\text{sccv}'_1} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} ; \mathbf{B}^{0.01}_{\text{sccv}'_0} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 & 0 \end{pmatrix} \tag{5.28}$$

and the corresponding SC-JSCC is denoted as $\mathbf{B}^{0.01}_{\text{TD}_{12}}$.

Secondly, we start by setting $m_0 = m_1 = m_3 = 1$ and $m_2 = 0$, and construct source sub-protomatrices $\mathbf{B}^{0.01}_{\text{s}_0}$ and $\mathbf{B}^{0.01}_{\text{s}_1}$, channel sub-protomatrices $\mathbf{B}^{0.01}_{\text{c}_0}$ and $\mathbf{B}^{0.01}_{\text{c}_1}$, and SC-SVCC

Table 5.4: Channel thresholds and source thresholds of different codes at $p_1 = 0.01$. Shannon limit equals -12.02 dB.

Code	$\mathbf{B}^{0.01}_{\mathrm{TD}_{11}}$		$\mathbf{B}^{0.01}_{\mathrm{TD}_{12}}$		$\mathbf{B}^{0.01}_{\mathrm{TD}_{21}}$		$\mathbf{B}^{0.01}_{\mathrm{TD}_{22}}$		$\mathbf{B}^{\mathrm{opt2}}_{\mathrm{J}_{\mathrm{new\_0.01}}}$	$\mathbf{B}^{0.01}_{\mathrm{TD}_{\mathrm{new}}}$ [76]	
w	6	8	6	8	6	8	6	8	None	6	8
$(E_s/N_0)_{\mathrm{th}}$ (dB)	-9.999	-10.795	-11.087	-11.112	-10.646	-10.850	-11.104	-11.153	-10.170	-11.022	-11.112
p_{th}	0.045	0.046	0.045	0.046	0.091	0.095	0.091	0.095	0.084	0.031	0.032

sub-protomatrices $\mathbf{B}^{0.01}_{\mathrm{svcc}_0}$ and $\mathbf{B}^{0.01}_{\mathrm{svcc}_1}$ based on the JSC-BC $\mathbf{B}^{\mathrm{opt2}}_{\mathrm{J}_{\mathrm{new\_0.01}}}$ (5.12). We obtain

$$\mathbf{B}^{0.01}_{\mathrm{s}_1} = \begin{pmatrix} 0 & 1 & 2 & 2 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 2 & 0 & 1 & 0 & 0 \end{pmatrix} ; \ \mathbf{B}^{0.01}_{\mathrm{s}_0} = \begin{pmatrix} 1 & 2 & 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 2 & 0 & 1 & 1 & 2 & 1 \end{pmatrix} ;$$

$$\mathbf{B}^{0.01}_{\mathrm{c}_1} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} ; \ \mathbf{B}^{0.01}_{\mathrm{c}_0} = \begin{pmatrix} 2 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 2 & 0 & 0 & 2 & 1 \end{pmatrix} ;$$

$$\mathbf{B}^{0.01}_{\mathrm{svcc}_1} = \begin{pmatrix} 0 & 0 & 2 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix} ; \ \mathbf{B}^{0.01}_{\mathrm{svcc}_0} = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} ; \quad (5.29)$$

$$\mathbf{B}^{0.01}_{\mathrm{sccv}_0} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 3 & 1 & 0 & 0 & 0 \end{pmatrix} ;$$

and we denote the corresponding SC-JSCC as $\mathbf{B}^{0.01}_{\mathrm{TD}_{21}}$. Then, we set $m_2 = 1$ and use the DE algorithm to construct the SC-SCCV sub-protomatrices based on $\mathbf{B}^{0.01}_{\mathrm{s}_i}$, $\mathbf{B}^{0.01}_{\mathrm{c}_i}$, and $\mathbf{B}^{0.01}_{\mathrm{svcc}_i}$ ($i = 0, 1$) shown in (5.29) to form a new SC-JSCC. We set the maximum row weight of the SC-JSCC no larger than that of $\mathbf{B}^{\mathrm{opt2}}_{\mathrm{J}_{\mathrm{new\_0.01}}}$ to maintain the same decoding complexity as $\mathbf{B}^{\mathrm{opt2}}_{\mathrm{J}_{\mathrm{new\_0.01}}}$. The SC-SCCV linking sub-protomatrices obtained are

$$\mathbf{B}^{0.01}_{\mathrm{sccv}_1} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix} ; \ \mathbf{B}^{0.01}_{\mathrm{sccv}_0} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 & 0 \end{pmatrix} \quad (5.30)$$

and the corresponding SC-JSCC code is denoted as $\mathbf{B}^{0.01}_{\mathrm{TD}_{22}}$.

Table 5.4 lists the source thresholds and channel thresholds of the new SC-JSCCs, $\mathbf{B}^{\mathrm{opt2}}_{\mathrm{J}_{\mathrm{new\_0.01}}}$ (with best error performance among block codes), and $\mathbf{B}^{0.01}_{\mathrm{TD}_{\mathrm{new}}}$ [76]. We have the following observations.

i) The channel thresholds of Type-II SC-JSCCs, namely $\mathbf{B}_{\mathrm{TD}_{12}}^{0.01}$ and $\mathbf{B}_{\mathrm{TD}_{22}}^{0.01}$ differ within 0.05 dB between window size $w = 6$ and $w = 8$.

ii) The channel thresholds of Type-I SC-JSCCs, i.e., $\mathbf{B}_{\mathrm{TD}_{11}}^{0.01}$ and $\mathbf{B}_{\mathrm{TD}_{21}}^{0.01}$, as well as the SC-DP-LDPC code $\mathbf{B}_{\mathrm{TD}_{\mathrm{new}}}^{0.01}$ [76], exhibit a larger difference between window size $w = 6$ and $w = 8$, ranging from 0.09 dB to 0.8 dB.

iii) $\mathbf{B}_{\mathrm{TD}_{22}}^{0.01}$ at $w = 8$ has a channel threshold of 0.041 dB lower than $\mathbf{B}_{\mathrm{TD}_{\mathrm{new}}}^{0.01}$ at $w = 8$.

iv) All SC-JSCCs with $w = 8$ have lower channel thresholds than the JSC-BC code $\mathbf{B}_{\mathrm{J}_{\mathrm{new\_0.01}}}^{\mathrm{opt2}}$.

v) $\mathbf{B}_{\mathrm{TD}_{21}}^{0.01}$ and $\mathbf{B}_{\mathrm{TD}_{22}}^{0.01}$ have the highest source thresholds among these codes.

Figure 5.7 shows the SSER performance of the SC-JSCCs and the SC-DP-LDPC code $\mathbf{B}_{\mathrm{TD}_{\mathrm{new}}}^{0.01}$ when $w = 6$ and $w = 8$. For $w = 8$, we set $z = 4 \cdot 50 = 200$. For $w = 6$, we set $z = 4 \cdot 67 = 268$. w and z are set to maintain a similar code length in each window and to achieve a similar decoding latency as the JSC-BC $\mathbf{B}_{\mathrm{J}_{\mathrm{new\_0.01}}}^{\mathrm{opt2}}$. Figure 5.7 also presents the SSER performance of $\mathbf{B}_{\mathrm{J}_{\mathrm{new\_0.01}}}^{\mathrm{opt2}}$ for comparison. We have the following observations.

i) Type-I SC-JSCCs, i.e., $\mathbf{B}_{\mathrm{TD}_{11}}^{0.01}$ and $\mathbf{B}_{\mathrm{TD}_{21}}^{0.01}$, as well as the SC-DP-LDPC code $\mathbf{B}_{\mathrm{TD}_{\mathrm{new}}}^{0.01}$ [76] with $w = 8$ have better error performance than that with $w = 6$. Their channel thresholds with $w = 8$ are lower than those with $w = 6$. Type-II SC-JSCCs, namely $\mathbf{B}_{\mathrm{TD}_{12}}^{0.01}$ and $\mathbf{B}_{\mathrm{TD}_{22}}^{0.01}$ at $w = 6$ have better error performance than those at $w = 8$. Their channel thresholds at $w = 6$ and $w = 8$ are very close. Additionally, the lifting factor at $w = 6$ is larger than that at $w = 8$. This suggests that the performance difference is not solely dependent on the channel threshold and indicates the importance of considering the lifting factor along with the channel threshold for a given window size so as to achieve good error performance in scenarios where the code lengths are similar within a window.

ii) Type-II SC-JSCCs, i.e., $\mathbf{B}_{\mathrm{TD}_{12}}^{0.01}$ and $\mathbf{B}_{\mathrm{TD}_{22}}^{0.01}$ at $w = 6$ have better error performance than $\mathbf{B}_{\mathrm{TD}_{\mathrm{new}}}^{0.01}$ at $w = 6$ and $w = 8$ and the JSC-BC $\mathbf{B}_{\mathrm{J}_{\mathrm{new\_0.01}}}^{\mathrm{opt2}}$ when E_s/N_0 is large.

iii) $\mathbf{B}_{\mathrm{TD}_{22}}^{0.01}$ at $w = 6$ has the best error performance among all codes.

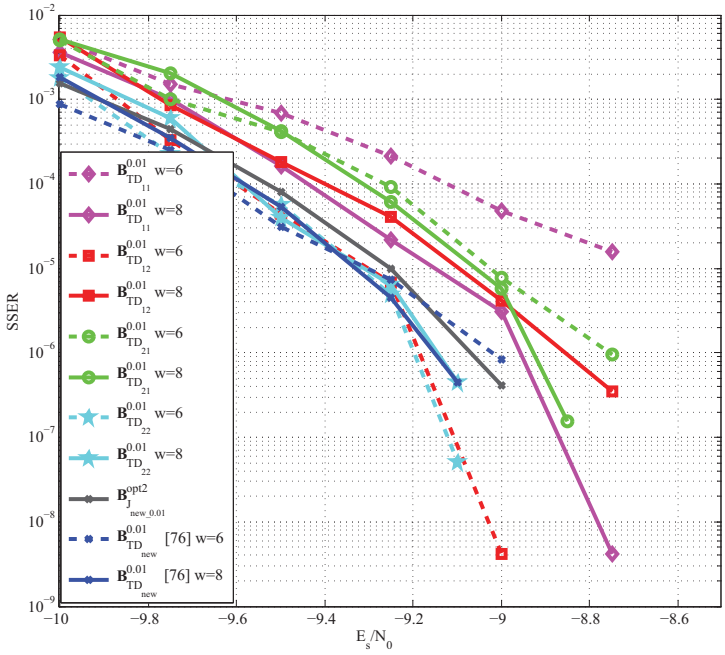


Figure 5.7: SSER performance comparison between new SC-JSCCs, $\mathbf{B}_{\mathrm{J}_{\mathrm{new\_0.01}}}^{\mathrm{opt2}}$ and SC-DP-LDPC codes at $p_1 = 0.01$. The lifting factors for SC-JSCCs are $z = z_1 z_2 = 4 \cdot 67 = 268$ and $z = z_1 z_2 = 4 \cdot 50 = 200$ when $w = 6$ and $w = 8$, respectively. $L_s = 128$ and $L_c = 130$. $R_{\mathrm{TD}} = 1.969$ for SC-JSCCs.

iv) All codes do not suffer from an error floor caused by the source compression because of their large source thresholds relative to $p_1 = 0.01$.

Example #5: When $p_1 = 0.04$, we first construct a SC-JSCC based on $\mathbf{B}_{\mathrm{J}_{\mathrm{new\_0.04}}}^{\mathrm{opt1}}$ (5.14). We begin by setting $m_0 = m_1 = m_3 = 1$ and $m_2 = 0$. The source SC-P-LDPC protomatrix and the channel SC-P-LDPC protomatrix are set to be the same as the SC-DP-LDPC code $\mathbf{B}_{\mathrm{TD}_{\mathrm{new}}}^{0.04}$ [76] given in Section 4.5, which are obtained based on $\mathbf{B}_{\mathrm{J}_{\mathrm{opt1}}}^{0.04}$ (5.13). The sub-protomatrices obtained

are

$$\begin{aligned}
\mathbf{B}_{\mathrm{s}_1'}^{0.04} &= \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \end{pmatrix} \; ; \; \mathbf{B}_{\mathrm{s}_0'}^{0.04} = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{pmatrix} \; ; \\
\mathbf{B}_{\mathrm{c}_1'}^{0.04} &= \begin{pmatrix} 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \; ; \; \mathbf{B}_{\mathrm{c}_0'}^{0.04} = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 1 \end{pmatrix} \; ; \\
\mathbf{B}_{\mathrm{svcc}_1'}^{0.04} &= \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \; ; \; \mathbf{B}_{\mathrm{svcc}_0'}^{0.04} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \; ; \\
\mathbf{B}_{\mathrm{sccv}_0'}^{0.04} &= \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \end{pmatrix} \; ;
\end{aligned} \tag{5.31}$$

and the corresponding SC-JSCC is denoted as $\mathbf{B}_{\mathrm{TD}_{11}}^{0.04}$. Next, we set $m_2 = 1$ and use the DE algorithm (introduced in Section 2.5) to construct the SC-SCCV sub-protomatrices based on $\mathbf{B}_{\mathrm{s}_i'}^{0.04}$, $\mathbf{B}_{\mathrm{c}_i'}^{0.04}$, and $\mathbf{B}_{\mathrm{svcc}_i'}^{0.04}$ $(i = 0, 1)$ shown in (5.31) to form a new SC-JSCC. We set the maximum row weight of the SC-JSCC no larger than that of $\mathbf{B}_{\mathrm{J}_{\mathrm{opt\_new1}}}^{0.04}$ to maintain the same decoding complexity as $\mathbf{B}_{\mathrm{J}_{\mathrm{opt\_new1}}}^{0.04}$. The SC-SCCV linking sub-protomatrices obtained are

$$\mathbf{B}_{\mathrm{sccv}_1'}^{0.04} = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \; ; \; \mathbf{B}_{\mathrm{sccv}_0'}^{0.04} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \end{pmatrix} . \tag{5.32}$$

and the corresponding SC-JSCC code is denoted as $\mathbf{B}_{\mathrm{TD}_{12}}^{0.04}$.

Secondly, we begin by setting $m_0 = m_1 = m_3 = 1$ and $m_2 = 0$, and construct source sub-protomatrices $\mathbf{B}_{\mathrm{s}_0}^{0.04}$ and $\mathbf{B}_{\mathrm{s}_1}^{0.04}$, channel sub-protomatrices $\mathbf{B}_{\mathrm{c}_0}^{0.04}$ and $\mathbf{B}_{\mathrm{c}_1}^{0.04}$, and SC-SVCC

Table 5.5: Channel thresholds and source thresholds of different codes at $p_1 = 0.04$. Shannon limit equals -7.00 dB.

Code	$\mathbf{B}^{0.04}_{\mathrm{TD}_{11}}$		$\mathbf{B}^{0.04}_{\mathrm{TD}_{12}}$		$\mathbf{B}^{0.04}_{\mathrm{TD}_{21}}$		$\mathbf{B}^{0.04}_{\mathrm{TD}_{22}}$		$\mathbf{B}^{\mathrm{opt2}}_{\mathrm{J}_{\mathrm{new\_0.04}}}$	$\mathbf{B}^{0.04}_{\mathrm{TD}_{\mathrm{new}}}$ [76]	
w	6	8	6	8	6	8	6	8	None	6	8
$(E_s/N_0)_{\mathrm{th}}$ (dB)	-5.386	-6.120	-6.390	-6.430	-5.547	-6.365	-6.442	-6.476	-5.880	-6.311	-6.441
p_{th}	0.129	0.150	0.129	0.150	0.394	0.394	0.394	0.394	0.217	0.084	0.091

sub-protomatrices $\mathbf{B}^{0.04}_{\mathrm{svcc}_0}$ and $\mathbf{B}^{0.04}_{\mathrm{svcc}_1}$ based on the JSC-BC $\mathbf{B}^{\mathrm{opt2}}_{\mathrm{J}_{\mathrm{new\_0.04}}}$ (5.16). We obtain

$$
\begin{aligned}
\mathbf{B}^{0.04}_{\mathrm{s}_1} &= \begin{pmatrix} 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & 1 \end{pmatrix} ; \ \mathbf{B}^{0.04}_{\mathrm{s}_0} = \begin{pmatrix} 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix} ; \\
\mathbf{B}^{0.04}_{\mathrm{c}_1} &= \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \end{pmatrix} ; \ \mathbf{B}^{0.04}_{\mathrm{c}_0} = \begin{pmatrix} 3 & 0 & 1 & 0 & 0 \\ 2 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 \end{pmatrix} ; \\
\mathbf{B}^{0.04}_{\mathrm{svcc}_1} &= \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} ; \ \mathbf{B}^{0.04}_{\mathrm{svcc}_0} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \end{pmatrix} ; \\
\mathbf{B}^{0.04}_{\mathrm{sccv}_0} &= \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 3 & 1 & 0 & 0 & 0 \end{pmatrix} ;
\end{aligned}
\tag{5.33}
$$

and we denote the corresponding SC-JSCC as $\mathbf{B}^{0.04}_{\mathrm{TD}_{21}}$.

Next, we set $m_2 = 1$ and use the DE algorithm to construct the SC-SCCV sub-protomatrices based on $\mathbf{B}^{0.04}_{\mathrm{s}_i}$, $\mathbf{B}^{0.04}_{\mathrm{c}_i}$, and $\mathbf{B}^{0.04}_{\mathrm{svcc}_i}$ $(i = 0, 1)$ shown in (5.33) to form a new SC-JSCC. We set the maximum row weight of the SC-JSCC no larger than that of $\mathbf{B}^{\mathrm{opt2}}_{\mathrm{J}_{\mathrm{new\_0.04}}}$ to maintain the same decoding complexity as $\mathbf{B}^{\mathrm{opt2}}_{\mathrm{J}_{\mathrm{new\_0.04}}}$. The SC-SCCV linking sub-protomatrices obtained are

$$
\mathbf{B}^{0.04}_{\mathrm{sccv}_1} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 \end{pmatrix} ; \ \mathbf{B}^{0.04}_{\mathrm{sccv}_0} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \end{pmatrix}
\tag{5.34}
$$

and the corresponding SC-JSCC code is denoted as $\mathbf{B}^{0.04}_{\mathrm{TD}_{22}}$.

Table 5.5 lists the source thresholds and channel thresholds of the new SC-JSCCs, $\mathbf{B}^{\mathrm{opt2}}_{\mathrm{J}_{\mathrm{new\_0.04}}}$

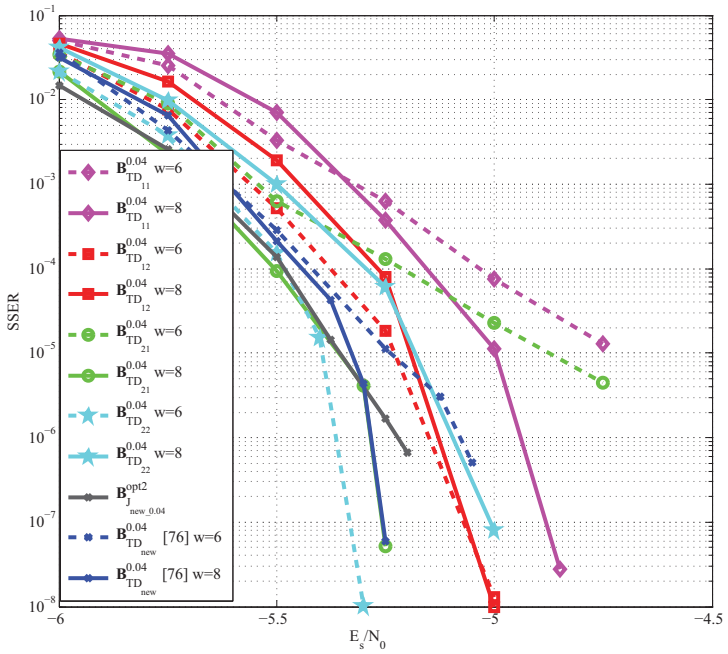


Figure 5.8: SSER performance comparison between new SC-JSCCs, $\mathbf{B}^{\mathrm{opt2}}_{\mathrm{J}_{\mathrm{new\_0.04}}}$ and SC-DP-LDPC codes at $p_1 = 0.04$. The lifting factors for SC-JSCCs are $z = z_1 z_2 = 4 \cdot 134 = 536$ and $z = z_1 z_2 = 4 \cdot 134 = 536$ when $w = 6$ and $w = 8$, respectively. $L_s = 128$ and $L_c = 130$. $R_{\mathrm{TD}} = 0.985$ for SC-JSCCs.

(with best error performance among block codes), and $\mathbf{B}^{0.04}_{\mathrm{TD}_{\mathrm{new}}}$ [76]. We have the following observations.

i) The thresholds of Type-II SC-JSCCs, namely $\mathbf{B}^{0.04}_{\mathrm{TD}_{12}}$ and $\mathbf{B}^{0.04}_{\mathrm{TD}_{22}}$, only differ within 0.04 dB between window size $w = 6$ and $w = 8$. Type-I SC-JSCCs, i.e., $\mathbf{B}^{0.04}_{\mathrm{TD}_{11}}$ and $\mathbf{B}^{0.04}_{\mathrm{TD}_{21}}$, as well as the SC-DP-LDPC code $\mathbf{B}^{0.04}_{\mathrm{TD}_{\mathrm{new}}}$ [76], exhibit a larger difference in their channel thresholds between window size $w = 6$ and $w = 8$, ranging from around 0.10 dB to 0.80 dB. This phenomenon is also observed in Table 5.4.

ii) $\mathbf{B}^{0.04}_{\mathrm{TD}_{22}}$ at $w = 8$ has a channel threshold of 0.035 dB lower than $\mathbf{B}^{0.04}_{\mathrm{TD}_{\mathrm{new}}}$ at $w = 8$.

iii) All SC-JSCCs with $w = 8$ have lower channel thresholds than the JSC-BC code $\mathbf{B}^{\mathrm{opt2}}_{\mathrm{J}_{\mathrm{new\_0.04}}}$.

iv) $\mathbf{B}^{0.04}_{\mathrm{TD}_{21}}$ and $\mathbf{B}^{0.04}_{\mathrm{TD}_{22}}$ have the highest source thresholds among these codes.

Figure 5.8 shows the SSER performance of the SC-JSCCs and the SC-DP-LDPC code $\mathbf{B}^{0.04}_{\mathrm{TD}_{\mathrm{new}}}$ when $w = 6$ and $w = 8$. For $w = 8$, we set $z = 4 \cdot 100 = 400$. For $w = 6$,

we set $z = 4 \cdot 134 = 536$. Figure 5.8 also presents the SSER performance of $\mathbf{B}_{\mathrm{J}_{\mathrm{new\_0.04}}}^{\mathrm{opt2}}$ for comparison. We can observe the followings.

a) Type-I SC-JSCCs, i.e., $\mathbf{B}_{\mathrm{TD}_{11}}^{0.04}$ and $\mathbf{B}_{\mathrm{TD}_{21}}^{0.04}$, as well as the SC-DP-LDPC code $\mathbf{B}_{\mathrm{TD}_{\mathrm{new}}}^{0.04}$ [76] with $w = 8$ have better error performance than that with $w = 6$. Their channel thresholds at $w = 8$ are lower than those at $w = 6$. Type-II SC-JSCCs, namely $\mathbf{B}_{\mathrm{TD}_{12}}^{0.04}$ and $\mathbf{B}_{\mathrm{TD}_{22}}^{0.04}$ at $w = 6$ have better error performance than that at $w = 8$. Their channel thresholds at $w = 6$ and $w = 8$ are very close;

b) $\mathbf{B}_{\mathrm{TD}_{22}}^{0.04}$ at $w = 6$ has the best error performance among those codes;

c) No error floor is observed due to the large source thresholds of these codes relative to $p_1 = 0.04$.

5.2.3.2 High-entropy sources

We design some SC-JSCCs for high-entropy sources. Based on the observations above, we can conclude that Type-II SC-JSCCs constructed based on the optimized JSC-BCs, i.e., $\mathbf{B}_{\mathrm{TD}_{22}}^{0.01}$ and $\mathbf{B}_{\mathrm{TD}_{22}}^{0.04}$, possess excellent theoretical thresholds and simulated SSER results at $w = 6$. Therefore in the following, we construct only Type-II SC-JSCCs based on the optimized JSC-BCs.

Example #6: When $p_1 = 0.10$, we consider $\mathbf{B}_{\mathrm{J}_{\mathrm{new\_0.10}}}^{\mathrm{opt2}}$ in (5.19). We set $m_0 = m_1 = m_2 = m_3 = 1$ and construct source sub-protomatrices $\mathbf{B}_{\mathrm{s}_0}^{0.10}$ and $\mathbf{B}_{\mathrm{s}_1}^{0.10}$, channel sub-protomatrices $\mathbf{B}_{\mathrm{c}_0}^{0.10}$ and $\mathbf{B}_{\mathrm{c}_1}^{0.10}$, SC-SVCC sub-protomatrices $\mathbf{B}_{\mathrm{svcc}_0}^{0.10}$ and $\mathbf{B}_{\mathrm{svcc}_1}^{0.10}$, and SC-SCCV sub-protomatrices $\mathbf{B}_{\mathrm{sccv}_0}^{0.10}$

and $\mathbf{B}^{0.10}_{\mathrm{sccv}_1}$, and we obtain

$$\begin{aligned}
\mathbf{B}^{0.10}_{\mathrm{s}_1} &= \begin{pmatrix} 0 & 0 & 2 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} \; ; \; \mathbf{B}^{0.10}_{\mathrm{s}_0} = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 \end{pmatrix} \; ; \\
\mathbf{B}^{0.10}_{\mathrm{c}_1} &= \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \; ; \; \mathbf{B}^{0.10}_{\mathrm{c}_0} = \begin{pmatrix} 2 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 2 & 1 \end{pmatrix} \; ; \\
\mathbf{B}^{0.10}_{\mathrm{svcc}_1} &= \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix} \; ; \; \mathbf{B}^{0.10}_{\mathrm{svcc}_0} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \; ; \\
\mathbf{B}^{0.10}_{\mathrm{sccv}_1} &= \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \; ; \; \mathbf{B}^{0.10}_{\mathrm{sccv}_0} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 & 0 \end{pmatrix} \; .
\end{aligned} \tag{5.35}$$

We denote the corresponding SC-JSCC as $\mathbf{B}^{0.10}_{\mathrm{TD}_1}$.

When $p_1 = 0.20$, we consider $\mathbf{B}^{\mathrm{opt2}}_{\mathrm{J}_{\mathrm{new\_0.20}}}$ in (5.20). We also set $m_0 = m_1 = m_2 = m_3 = 1$, and construct source sub-protomatrices $\mathbf{B}^{0.20}_{\mathrm{s}_0}$ and $\mathbf{B}^{0.20}_{\mathrm{s}_1}$, channel sub-protomatrices $\mathbf{B}^{0.20}_{\mathrm{c}_0}$ and $\mathbf{B}^{0.20}_{\mathrm{c}_1}$, SC-SVCC sub-protomatrices $\mathbf{B}^{0.20}_{\mathrm{svcc}_0}$ and $\mathbf{B}^{0.20}_{\mathrm{svcc}_1}$, and SC-SCCV sub-protomatrices $\mathbf{B}^{0.20}_{\mathrm{sccv}_0}$

Table 5.6: Channel thresholds and source thresholds of different codes at $p_1 = 0.10$ and $p_1 = 0.20$.

p_1	0.10			0.20		
Code	$\mathbf{B}_{\mathrm{TD}_1}^{0.10}$		$\mathbf{B}_{\mathrm{J}_{\mathrm{new\_0.10}}}^{\mathrm{opt2}}$	$\mathbf{B}_{\mathrm{TD}_1}^{0.20}$		$\mathbf{B}_{\mathrm{J}_{\mathrm{new\_0.20}}}^{\mathrm{opt2}}$
Shannon limit (dB)	-3.39			-0.65		
$(E_s/N_0)_{\mathrm{th}}$ (dB)	-2.958 ($w=6$)	-2.995 ($w=8$)	-2.595	0.259 ($w=6$)	0.209 ($w=8$)	0.831
p_{th}	0.434 ($w=6$)	0.434 ($w=8$)	0.171	0.318($w=6$)	0.318($w=8$)	0.288

and $\mathbf{B}_{\mathrm{sccv}_1}^{0.20}$, and we obtain

$$
\begin{aligned}
\mathbf{B}_{\mathrm{s}_1}^{0.20} &= \begin{pmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 \end{pmatrix} \; ; \; \mathbf{B}_{\mathrm{s}_0}^{0.20} = \begin{pmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \end{pmatrix} \; ; \\
\mathbf{B}_{\mathrm{c}_1}^{0.20} &= \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 2 & 0 & 0 & 0 & 0 \end{pmatrix} \; ; \; \mathbf{B}_{\mathrm{c}_0}^{0.20} = \begin{pmatrix} 3 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 \end{pmatrix} \; ; \\
\mathbf{B}_{\mathrm{svcc}_1}^{0.20} &= \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix} \; ; \; \mathbf{B}_{\mathrm{svcc}_0}^{0.20} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 1 \end{pmatrix} \; ; \\
\mathbf{B}_{\mathrm{sccv}_1}^{0.20} &= \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix} \; ; \; \mathbf{B}_{\mathrm{sccv}_0}^{0.20} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 & 0 \end{pmatrix} \; .
\end{aligned}
\tag{5.36}
$$

We denote the corresponding SC-JSCC as $\mathbf{B}_{\mathrm{TD}_1}^{0.20}$.

Table 5.6 lists the source thresholds and channel thresholds of the new SC-JSCCs and the JSC-BCs $\mathbf{B}_{\mathrm{J}_{\mathrm{new\_0.10}}}^{\mathrm{opt2}}$ and $\mathbf{B}_{\mathrm{J}_{\mathrm{new\_0.20}}}^{\mathrm{opt2}}$, at $p_1 = 0.10$ and $p_1 = 0.20$, respectively. We can see that the SC-JSCCs have lower channel thresholds and higher source thresholds than the JSC-BCs at both p_1 values. $\mathbf{B}_{\mathrm{TD}_1}^{0.10}$ and $\mathbf{B}_{\mathrm{TD}_1}^{0.20}$ have close channel thresholds at $w = 6$ and $w = 8$. When $w = 6$, their channel thresholds lie within 0.45 dB and 0.91 dB, respectively, of the Shannon limits. Figure 5.9 shows the SSER performance of $\mathbf{B}_{\mathrm{TD}_1}^{0.10}$ and $\mathbf{B}_{\mathrm{TD}_1}^{0.20}$ at $w = 6$ and the SSER performance of $\mathbf{B}_{\mathrm{J}_{\mathrm{new\_0.10}}}^{\mathrm{opt2}}$ and $\mathbf{B}_{\mathrm{J}_{\mathrm{new\_0.20}}}^{\mathrm{opt2}}$. The proposed SC-JSCCs have similar error performance as the JSC-BCs in the low to medium E_s/N_0 region, but outperform them in the high E_s/N_0 region.

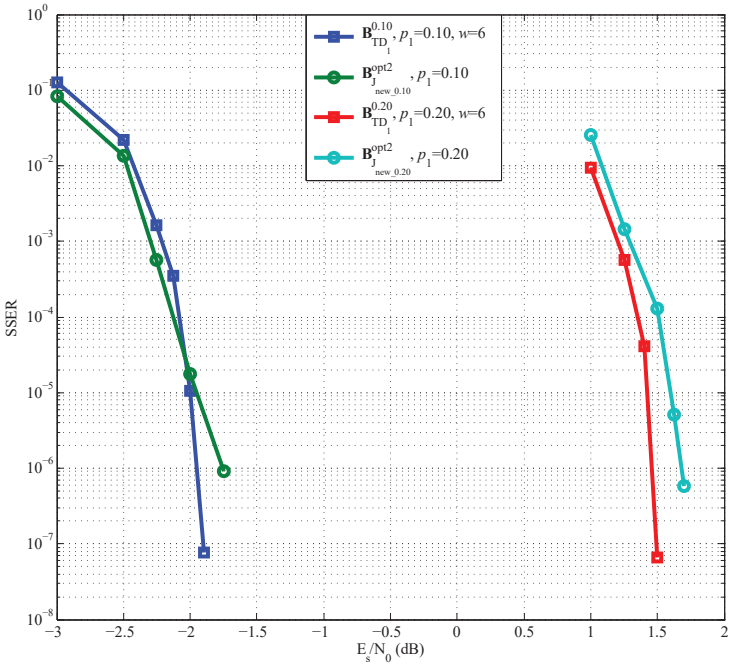


Figure 5.9: SSER performance comparison between new SC-JSCCs and new JSC-BC codes at $p_1 = 0.10$ and $p_1 = 0.20$. The lifting factors for SC-JSCCs are $z = z_1 z_2 = 4 \cdot 100 = 400$ when $w = 6$. $L_s = 128$ and $L_c = 130$. $R_{\mathrm{TD}} = 0.985$ for SC-JSCCs.

5.3 Conclusions

In this chapter, we first proposed a joint source-channel block code (JSC-BC), whose SVCC linking base matrix is a non-zero matrix and SCCV linking base matrix consists of a zero matrix and a lower or upper triangular base matrix with "1"s on its diagonal. Next, we propose an efficient UP-EXIT algorithm, where only untransmitted VNs and their connected nodes are considered, to calculate the source threshold of a JSC-BC. For both low-entropy and high-entropy sources, we construct JSC-BCs having good source and channel thresholds. SSER simulation results show that the new JSC-BCs outperform the existing DP-LDPC codes. We also propose a new type of spatially-coupled joint source-channel code (SC-JSCC). Moreover, with 200 iterations, the channel threshold of a SC-JSCC can be as close as within 0.45 dB of the Shannon limit. Theoretical analyses and simulation results further show that the new SC-JSCCs can achieve better error performance than SC-DP-LDPCs and JSC-BCs. Since our proposed JSCC schemes can achieve good error performance for both low-entropy and high-entropy sources, we will consider applying them to multi-source transmission scenarios in the

future.

Chapter 6

Conclusion and Future Work

In this chapter, we conclude the thesis and provide potential future research directions.

6.0.1 Conclusions

Utilizing the joint source-channel coding technique, which exploits both the source statistics and channel information, can yield superior error performance compared to separate source-channel designs in the case of finite code lengths. A well-established joint source-channel coding system involves the utilization of DP-LDPC codes, wherein two P-LDPC codes are employed as the source code and the channel code, respectively. A traditional joint source-channel coding system based on DP-LDPC codes establishes a one-to-one connection between the check nodes in the source P-LDPC code and the variable nodes in the channel P-LDPC code. This connection can be denoted by a source-check-channel-variable (SCCV) linking base matrix consisting of an identity matrix and a zero matrix. In our research, we concentrate on enhancing the error performance of the joint source-channel coding system by designing new systems based on the structure of traditional DP-LDPC codes. We propose several novel joint source-channel systems as follows.

1. In Chapter 3, we propose a novel type of DP-LDPC codes by replacing the identity matrix in the SCCV linking matrix of traditional DP-LDPC codes with a lower or upper triangular matrix with "1"s on its diagonal. By doing this, we allow for more versatility in the

code design. Furthermore, the linear source compression feature is preserved. Theoretical analyses have demonstrated that the proposed new DP-LDPC codes have lower channel thresholds. Error rate simulation results further reinforce the superiority of the proposed DP-LDPC codes compared with the traditional DP-LDPC codes and are consistent with the channel thresholds.

2. In Chapter 4, we build upon the advantages of spatially coupled protograph-based LDPC (SC-P-LDPC) codes over P-LDPC block codes and propose a new joint source-channel coding scheme called spatially coupled double protograph-based LDPC (SC-DP-LDPC) codes. In this design, two SC-P-LDPC codes are spatially coupled using a spatially coupled SCCV (SC-SCCV) linking matrix. Unlike a simple cascading relationship between the source SC-P-LDPC code and the channel SC-P-LDPC code, our design establishes a more intricate connection. The proposed SC-DP-LDPC codes enhance the correlation between codewords generated at different time instances. The current compressed source symbols are not only influenced by the previous source symbols, but also affected by the previously generated channel codewords. Both theoretical analyses and simulation results demonstrate the superiority of the proposed SC-DP-LDPC codes compared to concatenated SC-P-LDPC codes and state-of-the-art DP-LDPC block codes under the same decoding latency and similar overall code rate.

3. Studies have proved that source thresholds can be improved by adding connections between variable nodes (VNs) in the source P-LDPC and check nodes (CNs) in the channel P-LDPC, i.e., adding a source-variable-channel-check (SVCC) linking matrix in the traditional joint source-channel coding system based on DP-LDPC codes. In Chapter 5, we propose a novel joint source-channel block code (JSC-BC) in which two P-LDPC block codes are connected by a SVCC linking (base) matrix and a SCCV linking (base) matrix that consists of a zero matrix and a lower or upper triangular (base) matrix with “1”s on its diagonal, aiming to obtain good error performance for both low-entropy and high-entropy sources.

4. Moreover, in Chapter 5, we simplify the traditional joint protograph extrinsic information transfer (JP-EXIT) algorithm and propose an "untransmitted protograph-based EXIT (UP-EXIT) algorithm" for calculating the source threshold of a JSC-BC. Compared to the JP-EXIT algorithm, the proposed UP-EXIT algorithm is more efficient because a smaller protograph consisting of only the untransmitted VNs (i.e., the source VNs and the punctured channel VNs) and their connected check nodes need to be considered. By using the UP-EXIT algorithm, we first search for candidate codes with the proposed code structure and high source thresholds. Then we select those among the candidate codes also with low channel thresholds by using the JP-EXIT algorithm. Theoretical and simulation results show that the newly constructed codes outperform state-of-the-art DP-LDPC block codes.

5. In Chapter 5, we also spatially couple the joint source-channel block code and obtain a spatially coupled joint source-channel code (SC-JSCC). Theoretical analyses and simulation results show that even with a smaller window size and lower decoding complexity, the SC-JSCC with the spatially coupled structure for each sub-block (source protomatrix, channel protomatrix, SCCV linking matrix, and SVCC linking matrix) can achieve better error performance than existing spatially-coupled DP-LDPC codes.

6.0.2 Future work

Based on the research conducted in this thesis, the following future work can be considered to expand the findings.

1. For SC-DP-LDPC codes and SC-JSCCs, there exists an error propagation phenomenon. In addition, the decoding iterations will stop only when the maximum number of iterations is reached. The sliding window joint belief propagation decoding algorithm can be improved to mitigate error propagation and reduce the number of iterations without worsening error performance.

2. We can explore hardware-friendly implementations of the JSC-BC and SC-JSCC codes.

We can consider the practical constraints of hardware platforms and develop optimized implementations for specific applications or communication systems.

3. Due to the excellent error performance of JSC-BCs and SC-JSCCs for both low-entropy and high-entropy sources, we can consider applying them in multi-source scenarios, where multiple independent sources of data need to be transmitted simultaneously or sequentially.

4. For high reliability and low latency communication, a joint source-channel anytime coding scheme has been proposed by others [78, 79], where two anytime spatially coupled repeat-accumulate (SC-RA) codes are concatenated. We can also investigate the change of their concatenated relationship by changing the SCCV linking matrix to improve the error performance. Compared with SC-P-LDPC codes, the connections between blocks vary over time for the anytime SC-RA codes.

Bibliography

[1] C. E. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J.*, vol. 27, no. 3, pp. 379–423, Oct. 1948.

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

[3] H. P. Medeiros, M. C. Maciel, R. Demo Souza, and M. E. Pellenz, "Lightweight data compression in wireless sensor networks using Huffman coding," *International Journal of Distributed Sensor Networks*, vol. 10, no. 1, pp. 672–921, Jan. 2014.

[4] S. S. M. Than, "Secure data transmission in video format based on LSB and Huffman coding," *International Journal of Image, Graphics and Signal Processing*, vol. 12, no. 1, pp. 10–17, Feb. 2020.

[5] J. Rissanen, "A universal data compression system," *IEEE Trans. Inf. Theory.*, vol. 29, no. 5, pp. 656–664, Sep. 1983.

[6] I. H. Witten, R. M. Neal, and J. G. Cleary, "Arithmetic coding for data compression," *Communications of the ACM*, vol. 30, no. 6, pp. 520–540, Jun. 1987.

[7] R. W. Hamming, "Error detecting and error correcting codes," *The Bell system technical journal*, vol. 29, no. 2, pp. 147–160, May. 1950.

[8] I. S. 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, Jun. 1960.

[9] P. Elias, "Error-free coding," *IRE Trans. Inform. Theory*, vol. PGIT-4, pp. 29–37, Sep.1954.

[10] E. R. Berlekamp, "Non-binary BCH decoding," North Carolina State University. Dept. of Statistics, Tech. Rep., 1966.

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

[12] R. Gallager, "Low-density parity-check codes," *IRE Trans. Inf. Theory.*, vol. 8, no. 1, pp. 21–28, Sep. 1962.

[13] E. Arıkan, "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, Jan. 2009.

[14] K. Sayood and J. C. Borkenhagen, "Use of residual redundancy in the design of joint source/channel coders," *IEEE Trans. Commun.*, vol. 39, no. 6, pp. 838–846, June. 1991.

[15] J. Hagenauer, "Source-controlled channel decoding," *IEEE Trans. Commun.*, vol. 43, no. 9, pp. 2449–2457, Sep. 1995.

[16] M. Jeanne, J. C. Carlach, and P. Siohan, "Joint source-channel decoding of variable-length codes for convolutional codes and turbo codes," *IEEE Trans. Commun.*, vol. 53, no. 1, pp. 10–15, Jan. 2005.

[17] W. Xu, J. Hagenauer, and J. Hollmann, "Joint source-channel decoding using the residual redundancy in compressed images," in *Proc. 1996 Int. Conf. Communications*. Dallas, USA: IEEE, Jun. 1996, pp. 142–148.

[18] J. P. Woodard and L. Hanzo, "Comparative study of turbo decoding techniques: An overview," *IEEE Trans. Veh. Technol.*, vol. 49, no. 6, pp. 2208–2233, Nov. 2000.

[19] 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.

[20] L. Guivarch, J.-C. Carlach, and P. Siohan, "Joint source-channel soft decoding of Huffman codes with turbo-codes," in *Proc. IEEE Data Compression Conf. DCC*. Snowbird, USA: IEEE, Mar. 2000, pp. 83–92.

[21] A. Zribi, R. Pyndiah, S. Zaibi, F. Guilloud, and A. Bouallegue, "Low-complexity soft decoding of Huffman codes and iterative joint source channel decoding," *IEEE Trans. Commun.*, vol. 60, no. 6, pp. 1669–1679, June. 2012.

[22] M. Fresia, F. Perez-Cruz, and H. V. Poor, "Optimized concatenated LDPC codes for joint source-channel coding," in *Proc. ISIT*. Seoul, Korea (South): IEEE, Jun. 2009.

[23] M. Fresia, F. Perez-Cruz, H. V. Poor, and S. Verdú, "Joint source and channel coding," *IEEE Signal Process. Mag.*, vol. 27, no. 6, pp. 104–113, Nov. 2010.

[24] J. G. He, L. Wang, and P. Chen, "A joint source and channel coding scheme base on simple protograph structured codes," in *Proc. Int. Symp. Commun. Inform. Tech.*, Gold Coast, Australia, Oct. 2012, pp. 65–69.

[25] D. Divsalar, S. Dolinar, C. Jones, and K. Andrews, "Capacity approaching protograph codes," *IEEE J. Sel. Areas Commun.*, vol. 27, no. 6, pp. 876–888, Aug. 2009.

[26] Y. Fang, G. Bi, Y. L. Guan, and F. C. M. Lau, "A survey on protograph LDPC codes and their applications," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 4, pp. 1989–2016, 4th Quart., 2015.

[27] D. Divsalar, S. Dolinar, and C. Jones, "Low-rate LDPC codes with simple protograph structure," in *Proc. IEEE Int. Symp. Inf. Theory*. Adelaide, Australia: IEEE, Sep. 2005, pp. 1622–1626.

[28] G. Liva and M. Chiani, "Protograph LDPC codes design based on EXIT analysis," in *Proc. IEEE GLOBECOM*. Washington, DC, USA: IEEE, Nov. 2007, pp. 3250–3254.

[29] Q. Chen, F. C. M. Lau, H. Wu, and C. Chen, "Analysis and improvement of error-floor performance for JSCC scheme based on double protograph LDPC codes," *IEEE Trans. Veh. Technol.*, vol. 69, no. 12, pp. 14 316–14 329, Dec. 2020.

[30] L. Wang, L. Xu, and S. H. Hong, "New results on radiography image transmission with unequal error protection using protograph double LDPC codes," in *Proc. Int. Symp. on Medical Information and Communication Technology (ISMICT)*, Firenze, April 2014, pp. 1–4.

[31] J. G. He, Y. Li, and G. F. Wu, "Performance improvement of joint source-channel coding with unequal power allocation," *IEEE Wireless Commun. Lett.*, vol. 6, no. 5, pp. 582–585, 2017.

[32] C. Chen, L. Wang, and Z. Xiong, "Matching criterion between source statistics and source coding rate," *IEEE Commun. Lett.*, vol. 19, no. 9, pp. 1504–1507, Sep. 2015.

[33] Q. Chen, L. Wang, S. Hong, and Z. Xiong, "Performance improvement of JSCC scheme through redesigning channel code," *IEEE Commun. Lett.*, vol. 20, no. 6, pp. 1088–1091, Jun. 2016.

[34] C. Chen, L. Wang, and F. C. M. Lau, "Joint optimization of protograph LDPC code pair for joint source and channel coding," *IEEE Trans. Commun.*, vol. 66, no. 8, pp. 3255–3267, Aug. 2018.

[35] Q. Chen, L. Wang, S. Hong, and Y. Chen, "Integrated design of JSCC scheme based on double protograph LDPC codes system," *IEEE Commun. Lett.*, vol. 23, no. 2, pp. 218–221, Feb. 2019.

[36] S. Hong, J. Ke, and L. Wang, "Global design of double protograph LDPC codes for joint source-channel coding," *IEEE Commun. Lett.*, vol. 27, no. 2, pp. 424–427, Feb. 2023.

[37] S. Liu, L. Wang, J. Chen, and S. Hong, "Joint component design for the JSCC system based on DP-LDPC codes," *IEEE Trans. Commun.*, vol. 68, no. 9, pp. 5808–5818, Jun. 2020.

[38] S. Liu, C. Chen, L. Wang, and S. Hong, "Edge connection optimization for JSCC system based on DP-LDPC codes," *IEEE Wireless Commun. Lett*, vol. 8, no. 4, pp. 996–999, Aug. 2019.

[39] J. Kim, A. Ramamoorthy, and S. W. McLaughlin, "The design of efficiently-encodable rate-compatible LDPC codes," *IEEE Trans. Commun.*, vol. 57, no. 2, pp. 365–375, Feb. 2009.

[40] T. J. Richardson and R. L. Urbanke, "Efficient encoding of low-density parity-check codes," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 638–656, Feb. 2001.

[41] A. E. Pusane, A. J. Feltström, A. Sridharan, M. Lentmaier, K. S. Zigangirov, and J. D. J. Costello, "Implementation aspects of LDPC convolutional codes," *IEEE Trans. Commun.*, vol. 56, no. 7, pp. 1060–1069, Jul. 2008.

[42] A. E. Pusane, R. Smarandache, P. O. Vontobel, and D. J. Costello, "Deriving good LDPC convolutional codes from LDPC block codes," *IEEE Trans. Inf. Theory*, vol. 57, no. 2, pp. 835–857, Feb. 2011.

[43] 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.

[44] S. Kumar, A. J. Young, N. Macris, and H. D. Pfister, "Threshold saturation for spatially coupled LDPC and LDGM codes on BMS channels," *IEEE Trans. Inf. Theory*, vol. 60, no. 12, pp. 7389–7415, Dec. 2014.

[45] K. Klaiber, S. Cammerer, L. Schmalen, and S. T. Brink, "Avoiding burst-like error patterns in windowed decoding of spatially coupled LDPC codes," in *Proc. IEEE 10th Int. Symp.*

Turbo Codes Iterative Inf. Process. (ISTC). Hong Kong, China: IEEE, Dec. 2018, pp. 1–5.

[46] A. Golmohammadi and D. G. Mitchell, "Concatenated spatially coupled LDPC codes for joint source-channel coding," in *IEEE Int. Symp. Inf. Theory (ISIT)*. Vail, CO, USA: IEEE, Jun. 2018.

[47] A. Golmohammadi and D. G. M. Mitchell, "Concatenated spatially coupled LDPC codes with sliding window decoding for joint source-channel coding," *IEEE Trans. Commun.*, vol. 70, no. 2, pp. 851–864, Feb. 2021.

[48] H. V. B. Neto and W. Henkel, "Multi-edge optimization of low-density parity-check codes for joint source-channel coding," in *Proc. 9th Int. ITG Conf. Syst., Commun. Coding (SCC)*, Munich, Germany, Jan. 2013, pp. 1–6.

[49] S. Hong, Q. Chen, and L. Wang, "Performance analysis and optimization for edge connection of JSCC system based on double protograph LDPC codes," *IET Commun.*, vol. 12, no. 2, pp. 214–219, Jan. 2018.

[50] Q. Chen, S. Hong, and Y. Chen, "Design of linking matrix in JSCC scheme based on double protograph LDPC codes," *IEEE Access*, vol. 7, pp. 92 176–92 183, July. 2019.

[51] R. M. Tanner, "A recursive approach to low complexity codes," *IEEE Trans. Inf. Theory.*, vol. 27, no. 5, pp. 533–547, Sep. 1981.

[52] J. Thorpe, "Low-density parity-check (LDPC) codes constructed from protographs," *in IPN progress report*, vol. 42, no. 154, pp. 42–154, Aug. 2003.

[53] T. Richardson, R. Urbanke *et al.*, "Multi-edge type LDPC codes," in *Workshop honoring Prof. Bob McEliece on his 60th birthday*. California Institute of Technology, Pasadena, California: Citeseer, 2002.

[54] 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.

[55] A. Abbasfar, D. Divsalar, and K. Yao, "Accumulate repeat accumulate codes," *IEEE Trans. Commun.*, vol. 55, no. 4, pp. 692–702, Apr. 2007.

[56] D. Divsalar, C. Jones, S. Dolinar, and J. Thorpe, "Protograph based LDPC codes with minimum distance linearly growing with block size," in *Proc. IEEE Global Telecomm. Conf. (GLOBECOM)*, St Louis, MO, USA, Nov. 2005, pp. 1152–1156.

[57] S. Myung, K. Yang, and J. Kim, "Quasi-cyclic LDPC codes for fast encoding," *IEEE Trans. Inf. Theory.*, vol. 51, no. 8, pp. 2894–2901, Aug. 2005.

[58] Z. Li, L. Chen, L. Zeng, S. Lin, and W. H. Fong, "Efficient encoding of quasi-cyclic low-density parity-check codes," *IEEE Trans. Commun.*, vol. 54, no. 1, pp. 71–81, Jan. 2006.

[59] K. Zhang, X. Huang, and Z. Wang, "High-throughput layered decoder implementation for quasi-cyclic LDPC codes," *IEEE J. Sel. Area. Comm.*, vol. 27, no. 6, pp. 985–994, Aug. 2009.

[60] X. Y. 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.

[61] F. C. M. Lau and W. M. Tam, "A fast searching method for the construction of QC-LDPC codes with large girth," in *Proc. IEEE Symp. Comput. Commun. (ISCC)*, Cappadocia, Turkey, Jul. 2012, pp. 125–128.

[62] C. Chen, L. Wang, and S. Liu, "The design of protograph LDPC codes as source codes in a JSCC system," *IEEE Commun. Lett.*, vol. 22, no. 4, pp. 672–675, Apr. 2018.

[63] A. J. Felström 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.

[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] G. D. Forney, "The viterbi algorithm," *Proceedings of the IEEE*, vol. 61, no. 3, pp. 268–278, Mar. 1973.

[66] M. Lentmaier, A. Sridharan, D. J. Costello, and K. S. Zigangirov, "Iterative decoding threshold analysis for LDPC convolutional codes," *IEEE Trans. Inf. Theory.*, vol. 56, no. 10, pp. 5274–5289, Oct. 2010.

[67] 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.

[68] 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.

[69] K. Huang, D. G. 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.

[70] 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.

[71] A. R. Iyengar, P. H. Siegel, R. L. Urbanke, and J. K. Wolf, "Windowed decoding of spatially coupled codes," *IEEE Trans. Inf. Theory*, vol. 59, no. 4, pp. 2277–2292, Apr. 2013.

[72] 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.

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

[74] J. Zhan and F. C. M. Lau, "Joint design of source-channel codes with linear source encoding complexity and good channel thresholds based on double-protograph LDPC codes," *IEEE Commun. Lett.*, vol. 27, no. 11, pp. 2909–2913, Sep. 2023.

[75] Q. Lu, J. Fan, C.-W. Sham, W. M. Tam, and F. C. M. Lau, "A 3.0 Gb/s throughput hardware-efficient decoder for cyclically-coupled QC-LDPC codes," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 63, no. 1, pp. 134–145, Jan. 2016.

[76] J. Zhan and F. C. M. Lau, "Design of joint source-channel coding scheme based on spatially-coupled DP-LDPC codes," *IEEE Commun. Lett.*, vol. 28, no. 4, pp. 749–753, Apr. 2024.

[77] Q. Chen, Z. Xu, H. Wu, and G. Cai, "Analysis and optimization of a general linking matrix for JSCC scheme based on double LDPC codes," *Entropy*, vol. 25, no. 2, p. 382, Feb. 2023.

[78] L. Deng, Y. Wang, X. Yu, M. Noor-A-Rahim, Y. L. Guan, and Z. Shi, "Joint source channel anytime coding," in *Proc. IEEE Global Telecomm. Conf. (GLOBECOM)*. Taipei, Taiwan, China: IEEE, Dec. 2020, pp. 1–6.

[79] L. Deng, X. Yu, Y. Wang, M. Noor-A-Rahim, Y. L. Guan, Z. Shi, and Z. Zhang, "Joint source channel anytime coding based on spatially coupled repeat-accumulate codes," *IEEE Trans. Commun.*, vol. 71, no. 11, pp. 6215–6230, Nov. 2023.