Partially Coupled Codes for TB-based Transmission

Author:
Wu, Xiaowei

Publication Date:
2022

DOI:
https://doi.org/10.26190/unsworks/24030

Partially Coupled Codes for TB-based Transmission

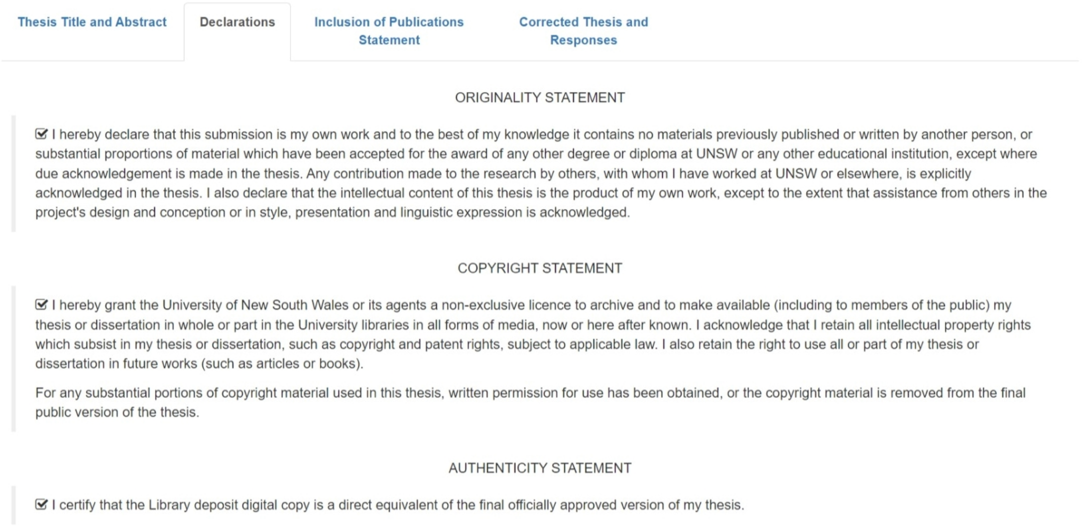
Xiaowei Wu

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

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

January 2022

Originality statement, Copyright and Authenticity statements


Thesis Title and Abstract
Declarations
Inclusion of Publications Statement
Corrected Thesis and Responses

ORIGINALITY STATEMENT

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

COPYRIGHT STATEMENT

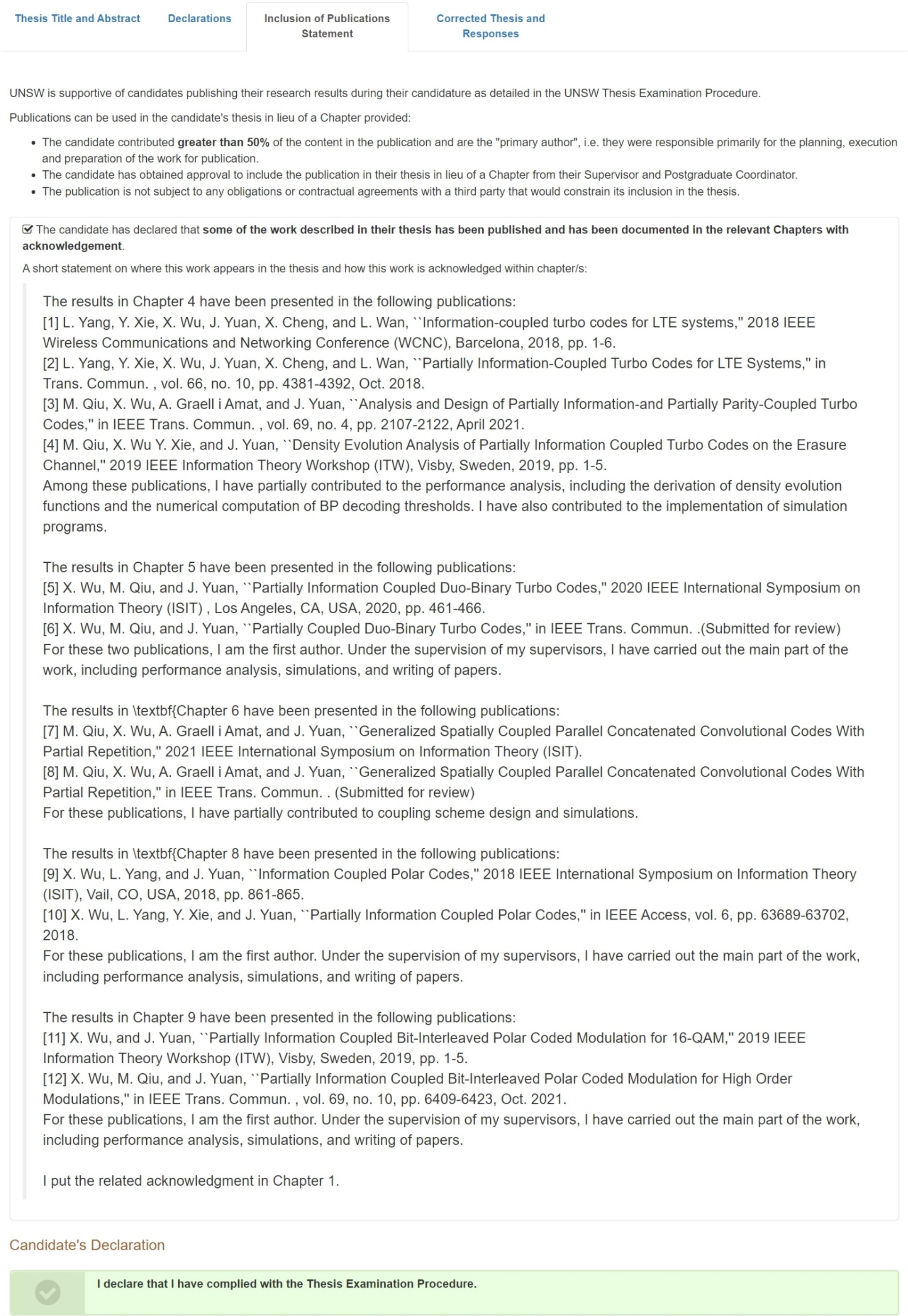
☑ I hereby grant the University of New South Wales or its agents a non-exclusive licence to archive and to make available (including to members of the public) my thesis or dissertation in whole or part in the University libraries in all forms of media, now or here after known. I acknowledge that I retain all intellectual property rights which subsist in my thesis or dissertation, such as copyright and patent rights, subject to applicable law. I also retain the right to use all or part of my thesis or dissertation in future works (such as articles or books).

For any substantial portions of copyright material used in this thesis, written permission for use has been obtained, or the copyright material is removed from the final public version of the thesis.

AUTHENTICITY STATEMENT

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

Inclusion of Publications Statement


Thesis Title and Abstract
Declarations
Inclusion of Publications Statement
Corrected Thesis and Responses

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

Publications can be used in the candidate's thesis in lieu of a Chapter provided:

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

☑ The candidate has declared that **some of the work described in their thesis has been published and has been documented in the relevant Chapters with acknowledgement**.

A short statement on where this work appears in the thesis and how this work is acknowledged within chapter/s:

> The results in Chapter 4 have been presented in the following publications:
> [1] L. Yang, Y. Xie, X. Wu, J. Yuan, X. Cheng, and L. Wan, ``Information-coupled turbo codes for LTE systems,'' 2018 IEEE Wireless Communications and Networking Conference (WCNC), Barcelona, 2018, pp. 1-6.
> [2] L. Yang, Y. Xie, X. Wu, J. Yuan, X. Cheng, and L. Wan, ``Partially Information-Coupled Turbo Codes for LTE Systems,'' in Trans. Commun. , vol. 66, no. 10, pp. 4381-4392, Oct. 2018.
> [3] M. Qiu, X. Wu, A. Graell i Amat, and J. Yuan, ``Analysis and Design of Partially Information-and Partially Parity-Coupled Turbo Codes,'' in IEEE Trans. Commun. , vol. 69, no. 4, pp. 2107-2122, April 2021.
> [4] M. Qiu, X. Wu Y. Xie, and J. Yuan, ``Density Evolution Analysis of Partially Information Coupled Turbo Codes on the Erasure Channel,'' 2019 IEEE Information Theory Workshop (ITW), Visby, Sweden, 2019, pp. 1-5.
> Among these publications, I have partially contributed to the performance analysis, including the derivation of density evolution functions and the numerical computation of BP decoding thresholds. I have also contributed to the implementation of simulation programs.
>
> The results in Chapter 5 have been presented in the following publications:
> [5] X. Wu, M. Qiu, and J. Yuan, ``Partially Information Coupled Duo-Binary Turbo Codes,'' 2020 IEEE International Symposium on Information Theory (ISIT) , Los Angeles, CA, USA, 2020, pp. 461-466.
> [6] X. Wu, M. Qiu, and J. Yuan, ``Partially Coupled Duo-Binary Turbo Codes,'' in IEEE Trans. Commun. .(Submitted for review)
> For these two publications, I am the first author. Under the supervision of my supervisors, I have carried out the main part of the work, including performance analysis, simulations, and writing of papers.
>
> The results in \textbf{Chapter 6 have been presented in the following publications:
> [7] M. Qiu, X. Wu, A. Graell i Amat, and J. Yuan, ``Generalized Spatially Coupled Parallel Concatenated Convolutional Codes With Partial Repetition,'' 2021 IEEE International Symposium on Information Theory (ISIT).
> [8] M. Qiu, X. Wu, A. Graell i Amat, and J. Yuan, ``Generalized Spatially Coupled Parallel Concatenated Convolutional Codes With Partial Repetition,'' in IEEE Trans. Commun. . (Submitted for review)
> For these publications, I have partially contributed to coupling scheme design and simulations.
>
> The results in \textbf{Chapter 8 have been presented in the following publications:
> [9] X. Wu, L. Yang, and J. Yuan, ``Information Coupled Polar Codes,'' 2018 IEEE International Symposium on Information Theory (ISIT), Vail, CO, USA, 2018, pp. 861-865.
> [10] X. Wu, L. Yang, Y. Xie, and J. Yuan, ``Partially Information Coupled Polar Codes,'' in IEEE Access, vol. 6, pp. 63689-63702, 2018.
> For these publications, I am the first author. Under the supervision of my supervisors, I have carried out the main part of the work, including performance analysis, simulations, and writing of papers.
>
> The results in Chapter 9 have been presented in the following publications:
> [11] X. Wu, and J. Yuan, ``Partially Information Coupled Bit-Interleaved Polar Coded Modulation for 16-QAM,'' 2019 IEEE Information Theory Workshop (ITW), Visby, Sweden, 2019, pp. 1-5.
> [12] X. Wu, M. Qiu, and J. Yuan, ``Partially Information Coupled Bit-Interleaved Polar Coded Modulation for High Order Modulations,'' in IEEE Trans. Commun. , vol. 69, no. 10, pp. 6409-6423, Oct. 2021.
> For these publications, I am the first author. Under the supervision of my supervisors, I have carried out the main part of the work, including performance analysis, simulations, and writing of papers.
>
> I put the related acknowledgment in Chapter 1.

Candidate's Declaration

✓ **I declare that I have complied with the Thesis Examination Procedure.**

Abstract

In this thesis, we investigate the design of partially coupled codes for transport block (TB) based transmission protocol adopted in 5G mobile communication network. In this protocol, an information sequence in a TB is segmented into multiple code blocks (CBs) and each CB is protected by a channel codeword independently. This is inefficient in terms of transmit power and spectrum efficiency because any erroneous CB in a TB leads to the retransmission of the whole TB. An important research problem related to this transport block based transmission is how to improve the TB error rate (TBER) performance so that the number of retransmissions reduces.

To tackle this challenge, we present a class of spatial coupling techniques called partial coupling in the TB encoding operation, which has two subclasses: partial information coupling (PIC) and partial parity coupling (PPC). To be specific, the coupling is performed such that a fraction of the information/parity sequence of one component code at the current CB is used as the input of the component encoder at the next CB, leading to improved TBER performance. Each CB is encoded with exactly the same component code. One of the appealing features of partial coupling (both PIC and PPC) is that the coupling can be applied to any component codes without changing their encoding and decoding architectures, making them compatible with the TB-based transmission protocol.

The main body of this thesis consists of two parts. In the first part of our research, we apply both PIC and PPC to turbo-like codes. We investigate various coupling designs and analysis the performance of the partially coupled turbo codes over the binary erasure channel (BEC) via density evolution (DE). Both simulation results and DE analysis show that existing turbo codes can significantly benefit from partial coupling schemes. To be specific, we first introduce the concept of partial coupling by constructing PIC-turbo codes (PIC-TCs) and PPC-turbo codes (PPC-TCs). The encoding structure and the corresponding decoding scheme of PIC-TCs and PPC-TCs are introduced, respectively. Their iterative decoding thresholds are optimized by exploring their exact density evolution function over the BECs. Then, we construct partially coupled duo-binary turbo codes, which overcome the rate-loss issue of partially coupled turbo codes. Various coding schemes within this class are proposed. By analyzing their decoding thresholds and distance properties respectively, we show that each scheme has different characteristics that can be exploited to satisfy different requirements. Last, we introduce a class of generalized spatially coupled turbo coding scheme, where PIC-TCs can be seen as a special subclass of it. Our analysis shows that the decoding threshold of such a generalized scheme can be very close to the BEC capacity.

In the second part of our research, we apply PIC technique to systematic polar codes and construct PIC-polar codes. We investigate the design of PIC-polar codes with respect to the polarization phenomenon. A DE based performance analysis is also conducted. Both performance analysis and simulation results confirm that polar codes can benefit from properly designed PIC schemes. To be specific, we note that due to the polarization phenomenon, the information bits of a polar code have different error probabilities. With this observation, we show that coupling the information bits with higher error probability effectively improves the performance of the underlaying polar codes. In addition, we also explore the design of PIC-polar coding scheme for high order modulations by employing bit-interleaved polar coded modulation (BIPCM) techniques. PIC-BIPCM constructed with different underlaying BIPCM schemes are proposed, and the joint design of coupling and modulation bit-mapping is discussed. Numerical results show that spatial coupling gain can be obtained on top of high order modulations.

Acknowledgment

I would like to thank those people who have provided help and support during my study in UNSW. It is very hard to thank everyone individually, so this acknowledge is just a token of my appreciation.

First of all, I would like to express the deepest appreciation to my supervisor Prof. Jinhong Yuan. I am very fortunate to be advised by Prof. Yuan and I truly appreciate his supervision. This thesis would not have been possible without the guidance and support from him. In particular, he always provides me with constructive feedback and insightful suggestions on my research. His deep understanding of the topics and enthusiasm for the research have inspired me a lot.

Second, I would like to thank Dr. Lei Yang, who has co-supervised me in the first year of my Ph.D studies. He gave me lots of advices on studying channel coding techniques, which was really helpful in the early stage of my research. I would like to thank Dr. Min Qiu, who has co-supervised me for the rest time of my studies. He gave many valuable suggestions on the research problems we discussed together, and helped me a lot on journal writing. I also appreciate Dr. Derrick Wing Kwan Ng for his valuable advices on my research and future career.

I am very grateful to my colleagues in the wireless communication lab of UNSW. Especially, I want to thank Yixuan Xie, Lou Zhao, Zhuo Sun, Zhiqiang Wei, Peng Kang, Bryan Liu, Yihuan Liao, and Shuangyang Li for all the precious

days we have studied together, shared our happiness and frustration, and helped each other. I also would like to express my gratitude to my friends in Sydney, especially to my flatmate Guanting Liu. Because of them, my experience in Sydney becomes so wonderful and unique.

Finally, my deepest appreciation go to my family who are the most important people for me. I especially would like to thank my parents for their support, understanding, and encourage for both my study and my life.

List of Publications

Journal Articles:

1. L. Yang, Y. Xie, **X. Wu**, J. Yuan, X. Cheng and L. Wan, "Partially Information-Coupled Turbo Codes for LTE Systems," in *IEEE Trans. Commun.*, vol. 66, no. 10, pp. 4381-4392, Oct. 2018.
2. **X. Wu**, L. Yang, Y. Xie and J. Yuan, "Partially Information Coupled Polar Codes," in IEEE Access, vol. 6, pp. 63689-63702, 2018.
3. M. Qiu, **X. Wu**, A. G. i. Amat and J. Yuan, "Analysis and Design of Partially Information-and Partially Parity-Coupled Turbo Codes," in *IEEE Trans. Commun.*, vol. 69, no. 4, pp. 2107-2122, April 2021.
4. **X. Wu**, M. Qiu and J. Yuan, "Partially Information Coupled Bit-Interleaved Polar Coded Modulation for High Order Modulations," in *IEEE Trans. Commun.*, vol. 69, no. 10, pp. 6409-6423, Oct. 2021.
5. **X. Wu**, M. Qiu and J. Yuan, "Partially Coupled Duo-Binary Turbo Codes," in *IEEE Trans. Commun.*. (Submitted for review)
6. M. Qiu, **X. Wu**, A. G. i. Amat and J. Yuan, "Generalized Spatially Coupled Parallel Concatenated Convolutional Codes With Partial Repetition," in *IEEE Trans. Commun.*. (Submitted for review)

Conference Articles:

1. L. Yang, Y. Xie, **X. Wu**, J. Yuan, X. Cheng and L. Wan, "Information-coupled turbo codes for LTE systems," 2018 IEEE Wireless Communications and Networking Conference (WCNC), Barcelona, 2018, pp. 1-6.
2. **X. Wu**, L. Yang and J. Yuan, "Information Coupled Polar Codes," *2018 IEEE International Symposium on Information Theory (ISIT)*, Vail, CO, USA, 2018, pp. 861-865.
3. M. Qiu, **X. Wu**, Y. Xie and J. Yuan, "Density Evolution Analysis of Partially Information Coupled Turbo Codes on the Erasure Channel," *2019 IEEE Information Theory Workshop (ITW)*, Visby, Sweden, 2019, pp. 1-5.

4. **X. Wu** and J. Yuan, “Partially Information Coupled Bit-Interleaved Polar Coded Modulation for 16-QAM,” *2019 IEEE Information Theory Workshop (ITW)*, Visby, Sweden, 2019, pp. 1-5.

5. **X. Wu**, M. Qiu and J. Yuan, “Partially Information Coupled Duo-Binary Turbo Codes,” *2020 IEEE International Symposium on Information Theory (ISIT)*, Los Angeles, CA, USA, 2020, pp. 461-466.

6. M. Qiu, **X. Wu**, A. G. i. Amat and J. Yuan, “Generalized Spatially Coupled Parallel Concatenated Convolutional Codes With Partial Repetition,” *2021 IEEE International Symposium on Information Theory (ISIT)*, Melbourne, Victoria, Australia, 2021.

7. **X. Wu**, M. Qiu, and J. Yuan, “Delayed Bit-Interleaved Polar Coded Modulation with Superposition Gray Labeling,” *IEEE Global Communications Conference Workshop*, 2021.

Abbreviations

2G	The second-generation
3G	The third-generation
3GPP	3rd Generation Partnership Project
4G	The fourth-generation
5G	The fifth-generation
6G	The sixth-generation
APP	A-priori probability
AR	Augmented reality
ARQ	Automatic repeat request
AWGN	Additive white Gaussian noise
BCC	Braided convolutional code
BCH	Bose-Chaudhuri-Hocquenghem
BEC	Binary erasure channel
BSC	Binary symmetric channel
BI-AWGN	Binary-input additive white Gaussian noise
BICM	Bit-interleaved coded modulation
BIPCM	Bit-interleaved polar coded modulation
BLER	Block error rate
BMS	Binary-input memoryless symmetric channel
BP	Belief propagation
bps	bits per second
BPSK	Binary phase-shift keying

CASCL	Cyclic redundancy check aided successive cancellation list
CB	Code block
CBER	Code block error rate
CBG	Code block group
CCSDS	Consultative committee for space data systems
CN	Check node
CRC	Cyclic redundancy check
dB	Decibel
DE	Density evolution
DMC	Discrete memoryless channel
DVB-S2	Second generation digital video broadcasting by satellite
dTC	Duo-binary turbo code
eMBB	Enhanced mobile broadband
GA	Gaussian approximation
HARQ	Hybrid automatic repeat request
IEEE	Institute of electrical and electronics engineers
IoT	Internet of things
KPI	Key performance indicator
LDPC	Low-density parity-check
LLR	Log likelihood ratio
LTE	Long term evolution
M2M	Machine-to-machine
MAP	Maximum-a-posterior
MFD	Maximized free distance
ML	Maximum likelihood
MLC	Multi-level coding
MLPC	Multi-level polar coding
mMTC	Massive machine-type communications
NR	New radio

ODS	Optimal distance spectrum
PAC	Polarization adjusted convolutional code
PAM	Pulse amplitude modulation
PCC	Parallel concatenated convolutional code
PIC	Partial information coupling
PIC-BIPCM	Partially information coupled bit-interleaved polar coded modulation
PIC-dTC	Partially information coupled duo-binary turbo code
PIC-dTC	Partially information coupled duo-binary parallel concatenated convolutional code
PIC-PC	Partially information coupled polar code
PIC-TC	Partially information coupled turbo code
PPC	Partial parity coupling
PPC-dTC	Partially parity coupled duo-binary turbo codes
PPC-dTC	Partially parity coupled duo-binary duo-binary parallel concatenated convolutional code
PPC-TC	Partially parity coupled turbo code
PSK	Phase-shift keying
QAM	Quadrature amplitude modulation
QoS	Quality-of-service
RM	Reed-Muller
RSC	Recursive systematic convolutional
SC	Successive cancellation
SCAN	Soft cancellation
SCC	Serially concatenated convolutional code
SCL	Successive cancellation list
SC-LDPC	Spatially coupled LDPC
SC-PCC	Spatially coupled parallel concatenated code
SC-SCC	Spatially coupled serially concatenated code
SC-TC	Spatially coupled turbo-like code

S-DMC	Symmetric discrete memoryless channel
SISO	Soft-in soft-out
SNR	Signal to noise ratio
SOVA	Soft-output Viterbi algorithm
TB	Transport block
TBCC	Tail-biting convolutional code
TBER	TB error rate
TTI	Transmit time interval
UMTS	Universal mobile telecommunications system
URLLC	Ultra reliable low latency communications
V2X	Vehicle-to-X
VN	Variable node
VR	Virtual reality
WAVA	Warp-around Viterbi algorithm
WiMAX	Worldwide Interoperability for Microwave Access

List of Notations

Scalars, vectors and matrices, set are written in italic, boldface lower-case and upper-case letters, calligraphic, respectively, e.g., x, $\mathbf{x}$ and $\mathbf{X}$.

$\mathbf{X}^{\mathrm{T}}$	Transpose of $\mathbf{X}$
$\mathbf{X}^{-1}$	Inverse of $\mathbf{X}$
$\lvert x\rvert$	Absolute value (modulus) of a complex scalar x
$\lVert\mathbf{x}\rVert$	The Euclidean norm of a vector $\mathbf{x}$
$\lVert\mathcal{X}\rVert$	The cardinality of a set $\mathcal{X}$
$\Pr\{E\}$	The probability of event E occurs
$P_X(x)$	Probability distribution of a random variable X
$P_{X\mid Y}(x\mid y)$	Conditional distribution of X given Y
$P_{X,Y}(x,y)$	Joint distribution of X and Y
$\lceil x\rceil$	Rounds a real number x to the nearest integer greater than or equal to x
$\Re(z)$	The real part of a complex number z
$\Im(z)$	The imaginary part of a complex number z
$\mathbf{0}$	The all-zero vector
$\mathbf{O}$	The all-zero matrix
$\mathbf{I}_N$	The N dimension identity matrix
$\mathbf{A}\otimes\mathbf{B}$	Kronecker product of two matrices $\mathbf{A}$ and $\mathbf{B}$
$\mathbf{A}^{\otimes n}$	n-fold Kronecker power of matrix $\mathbf{A}$
$\mathbf{G}_N$	A matrix of size $N\times N$
$\mathbf{G}_{\mathcal{A}}$	A submatrix of $\mathbf{G}_N$ consisting of rows with indices in $\mathcal{A}$ and $\mathcal{A}\subset\{1,2,\ldots,N\}$

$\mathbf{G}_{\mathcal{A}\mathcal{A}'}$	A submatrix of $\mathbf{G}_N$ consisting of elements $g_{i,j}$ with $i \in \mathcal{A}$, $j \in \mathcal{A}'$, and $\mathcal{A}, \mathcal{A}' \subset \{1, 2, \ldots, N\}$
$\mathbb{E}[\cdot]$	Statistical expectation
ln	Natural logarithm
log	Logarithm in base 2
min	Minimization
max	Maximization
$e^x, \exp(x)$	Natural exponential function
tanh	Hyperbolic tangent function
$\tanh^{-1}$	Inverse hyperbolic tangent function
$\oplus$	Modulo-2 addition
$\mathcal{N}(\mu, \sigma^2)$	Real-valued Gaussian random variable with mean μ and variance σ^2. The probability density function is $\displaystyle f(x\|\mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$
$\mathcal{N}_{\mathrm{c}}(\mu_{\mathrm{x}} + \mathrm{i}\mu_{\mathrm{y}}, \sigma^2)$	Circular symmetric complex Gaussian random variable $\displaystyle \Re\left[\mathcal{N}_{\mathrm{c}}(\mu_{\mathrm{x}} + \mathrm{i}\mu_{\mathrm{y}}, \sigma^2)\right] = \mathcal{N}\left(\mu_{\mathrm{x}}, \frac{\sigma^2}{2}\right)$ $\displaystyle \Im\left[\mathcal{N}_{\mathrm{c}}(\mu_{\mathrm{x}} + \mathrm{i}\mu_{\mathrm{y}}, \sigma^2)\right] = \mathcal{N}\left(\mu_{\mathrm{y}}, \frac{\sigma^2}{2}\right)$
$Q(x)$	Q-function, defined as: $\displaystyle Q(x) \triangleq \Pr\{\mathcal{N}(0,1) > x\} = \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{-\frac{z^2}{2}} \, dz$

Contents

Appendices

Appendices

List of Figures

List of Tables

Chapter 1

Introduction

Channel coding is a fundamental component in wireless communication systems. Since 2G, mobile communication networks have adopted state-of-the-art channel coding techniques. At the present 5G new radio (NR) standard, the performance of NR channel codes can almost approach the Shannon limit over an additive white Gaussian noise (AWGN) channel. The next generation (6G) mobile communication network, which is expected to be commercially available in 2030, will cover many diverse usage scenarios with extreme requirements such as ultra-high data rates, ultra-low latency, and ultra-low power consumption. Evolved channel coding techniques are needed to deliver reliable data transmission for these diverse use cases. In this thesis, we focus on coding techniques for high-speed data transmission. In this chapter, we first briefly present the research background. Then, we state our motivation of this research, and list the main contributions of the thesis.

1.1 Channel Coding for Mobile Communication Networks

Channel coding is a fundamental component of the basic point-to-point communication that provides reliability to all communication systems. In particular, it is essential for the success of various wireless communication systems.

The concept of channel coding was introduced in Claude Shannon's landmark paper [1] in 1948. Shannon showed that a channel can be characterized by its capacity, which measures how much information the channel can convey reliably. In practice, channel coding helps to improve user experience of wireless networks in many aspects. For instance, improved coding gain contributes to extending network coverage and improving quality of service (QoS), especially at cell edges. In addition, as one of the most complex and computationally intense components in the digital baseband processing chain, introducing highly parallelized decoding algorithms is able to improve peak data rates [2–5], while introducing low-complexity decoders substantially reduces power consumption, thereby extending battery life for mobile devices [6–8].

Since the introduction of Shannon's theorem [1], various channel coding schemes have been proposed and implemented in wireless communication systems. From 2G standard to 5G new radio (NR) standard, mobile communication networks have always adopted the state-of-the-art channel coding techniques. To be specific, 2G [9] uses convolutional codes. Both 3G [10, 11] and 4G [12] use convolutional codes and turbo codes for control channels and data channels, respectively. The latest 5G [13], standardized by the 3GPP Release 15 and 16, adopts polar codes and low-density parity-check (LDPC) codes for control channels and data channels, respectively.

1.2 Research Background and Motivation

At the present time, 5G is at an initial stage of large-scale commercial use. The technical specification of 3GPP states that the “full” 5G network would support three usage scenarios [14]: *enhanced mobile broadband* (eMBB), *ultra reliable low latency communications* (URLLC), and *massive machine-type communications* (mMTC). Up to the latest 3GPP release, LDPC codes and polar codes have been adopted by eMBB, while the channel coding techniques for URLLC and mMTC are under discussion.

The next generation (6G) mobile communication network is expected to be commercially available by 2030. 6G will not only make performance improvements in existing 5G usage scenarios, but also cover more diverse scenarios, including autonomous driving, industry verticals, satellite communications, and etc. These diverse use cases bring some extreme requirements such as ultra-high data rates, ultra-low latency, and ultra-low power consumption. To cover these diverse use cases, 6G needs powerful channel coding techniques.

The typical target key performance indicators (KPIs) of 6G are summarized as follows [15–17].

- Tbps throughput: Augmented reality (AR) and virtual reality (VR) will become a key application in various use cases including education, health care, industrial manufacturing, and etc. These applications rely on high-speed data transmission over short distances. To fulfill such a requirement, a peak data rate of 1 Tbps will be a basic KPI. To accomplish this target, the data length of each transmission will increase, which requires long channel codes of near-capacity performance with low decoding latency.

- Ultra reliable short-length codes: Machine type communications typically adopt channel codes with length around or lower than a few hundred bits. To support mission critical machine type communications, such as smart traffic control and remote surgery, 6G channel codes need to have performance approaching finite-length performance bound [18] to provide ultra reliable communication. Meanwhile, such performance should be obtained using a decoder with reasonable decoding complexity.

- Low power consumption decoding: Internet-of-Things (IoT) applications that mainly focus on sensing and data collection, such as environmental monitoring and wildfire prevention, require wireless networks to provide connectivity for a huge number of devices. Such types of devices are resource-limited due to battery constraints, thus low-complexity decoding are required to extend battery life.

In this thesis, we aim to design new spatial coupling schemes that support future high speed data transmission. In this section, we will review the channel coding techniques employed in 5G eMBB, and our motivations of studying spatial coupling schemes.

1.2.1 5G Channel Codes

NR LDPC Codes

LDPC codes were first introduced by R. Gallager in 1962 [19], and were rediscovered in the mid-1990s [20, 21]. Since then LDPC codes have attracted a lot of interest from both academia and industry, and were employed in various communication systems such as DVB-S2 [22], IEEE 802.16e [23] (WiMAX), IEEE 802.11n [24] (Wi-Fi 4), etc.

NR LDPC [13] codes use a quasi-cyclic structure, where the parity-check matrix is defined by a smaller base matrix such that each entry of the base matrix is either an all-zero matrix or a shifted identity matrix. NR LDPC codes employ two base matrices, supporting information length up to 8848 and 3840 bits, respectively. Meanwhile, NR LDPC codes have a rate-compatible structure, which can adjust code rates within a wide range between 1/5 and 5/9.

Compared to LTE turbo codes, NR LDPC codes have similar error performance with LTE turbo codes for rate equal to or below 1/2. At high code rates, such as 5/6, NR LDPC codes have significant performance gain over LTE turbo codes, because LTE turbo codes have performance degradation at high code rates. In addition, NR LDPC codes have advantages on decoding complexity and latency at high code rates. Unlike LTE turbo codes which have constant decoding complexity and latency, regardless of code rates, the decoding complexity and latency of NR LDPC codes decrease with the increment of code rates (see [25,26] and references therein).

NR Polar Codes

Polar codes, proposed by E. Arikan in 2009 [27], have been theoretically proven to be capacity-achieving for discrete memoryless channels. The main idea of polar coding to transform two identical copies of a discrete channel into a set of two correlated bit-channels such that one bit-channel has a higher capacity than the original channel, while the other bit-channel have a worse capacity. Applying such polarizing operation recursively gives a set of bit-channels whose capacity approaches either 0 (completely noisy) or 1 (noise-free). Polar codes take advantage of this polarization phenomenon by only transmitting information through the noise-free bit-channels.

NR polar codes [13] can support a large range of information length between

12 and 1013, and can support codeword length up to 1024. NR polar codes use a pre-transformed construction, where short outer codes are in serial concatenation with polar codes, and different outer codes are used for uplink and downlink transmissions, respectively. For uplink transmission, an information sequence is encoded with a CRC encoder, a parity check encoder, and a polar encoder in a row. For downlink transmission, an information sequence is processed with a CRC encoder, a CRC interleaver, and a polar encoder in a row.

In various research papers(see [28–30] and references therein), it has been shown that polar codes can outperform LDPC codes for blocklengths 1024 or lower. For large blocklengths, although polar codes can achieve a similar performance as LDPC codes, they have significantly higher decoding and hardware implementation complexity and latency at current research stage [25].

1.2.2 High Speed Data Transmission Protocol

The 5G eMBB data channels, as well as the 4G long term evolution (LTE) data channels, employ a *transport block* (TB) *based hybrid automatic repeat request* (HARQ) protocol to support high speed data transmission [12, 13]. In this protocol, a TB refers to a data packet sent in one single transmission. In each TB, the information sequence is segmented into several subsequences of the same length. Each subsequence is called a code block (CB). Then, each CB is encoded with a channel code independently. Later, all CBs are transmitted together as one TB. At the receiver, each CB is decoded independently. Once CB decoding error appears, the receiver will send a retransmission request (NACK) so that the entire TB will be retransmitted.

TB-base HARQ mechanism minimizes the HARQ feedback overhead as it only sends one feedback for a TB rather than sending feedback for each CB. However, it is inefficient in terms of transmit power and spectrum efficiency because any erroneous CB leads to the retransmission of the whole TB so that the successfully received CBs are also retransmitted. In particular, when the number of CBs in a TB increases, the waste of resources due to retransmission gets worse.

To mitigate this problem, CB-based HARQ was proposed and investigated in [31–33]. With this protocol, only erroneous CBs will be retransmitted. Although this approach saves transmission power and improves spectrum efficiency in data channels, it requires an excessive amount of HARQ feedback overhead in control channels, and consequently leads to the waste of control resources.

In addition, 5G introduces CB group (CBG) based HARQ [13] as a compromise between TB-based HARQ and CB-based HARQ. This protocol groups the CBs into CBGs and a TB may have 2, 4, or 8 CBGs. When decoding error appears, the CBGs consisting of erroneous CBs are retransmitted. Compared with TB-based HARQ, it effectively improves the transmission resource utilization efficiency with a small increment in the number of HARQ feedbacks.

1.2.3 Designing New Channel Coding Schemes for TB-based HARQ

For 5G and beyond, as the peak throughput is expected to reach Tbps level, a TB (or CBG) would consist of more CBs. In consequence, the transmission resource utilization problem associated with TB (or CBG) based HARQ would become more serious. In 4G, the downlink peak throughput is 1 Gbps, while a TB can consist of tens of CBs with maximum CB length of 6144 [12]. Moving to 5G, the downlink peak throughput reaches 20 Gbps, while the maximum CB length only increases to 8848 [13]. Thus, in 5G, a TB can consist of tens or even hundreds of CBs, while a CBG can consist of tens of CBs. The number of CBs is likely to keep increasing in future evolution as the downlink peak throughput of 6G is expected to reach 1 Tbps [15], while the maximum CB length cannot increase proportionally due to high complexity and latency for decoding long codes.

As such a transmission resource utilization problem of HARQ always exists, it is crucial to introduce advanced channel coding techniques to reduce the TB error rate (TBER) so that the number of retransmissions decreases. Before we talk about how to reduce TBER, we use a toy example on a simple average downlink latency model for LTE [34] to show how TBER affect transmission resources utilization. Such a model considers that a single transmission takes 4 transmission time intervals (TTI), where 1 TTI is typically 1 ms, and a HARQ round trip takes 8 TTI. For the typical case where there would be at most one retransmission, the average time for a successful TB transmission is $D = 4 + 8 * \text{TBER}$ TTI. When TBER reduces, the average transmission time decreases, so that spectrum efficiency (in terms of the number of TBs transmitted per unit time over the same spectrum) greatly improves.

Generally speaking, there are two approaches to reduce TBER. One straightforward approach is to improve the coding gain of the channel code that encodes each CB so that the error probability of each CB reduces. However, from the finite-length performance bound [18], it is not hard to tell that the potential coding gain of this approach might be limited due to the constraint of blocklength. Another approach is to exploit the benefits of long codes through spatial coupling, where a long codeword chain is constructed by coupling multiple short component channel codes in a certain structure.

Spatial coupling was firstly introduced in [35] for constructing convolutional LDPC codes, also known as spatially coupled LDPC codes. It is a code construction technique that connects a sequence of component codes to form a long codeword chain. Spatially coupled codes can achieve considerable coding gains over their uncoupled counterparts while maintain the decoding complexity in a reasonably low level by using windowed decoders [36,37]. Thus, spatially coupled codes became very popular in a wide range of applications in communications.

In this thesis, we present a class of spatial coupling techniques for communication systems with TB based HARQ, called partial coupling techniques. In particular, we apply partial coupling to turbo codes and polar codes, and investigate the design of coupling schemes, respectively. In the following sections, we will present the thesis outline and our research contributions.

1.3 Thesis Outline

1.3.1 Thesis Organization

There are ten chapters in total, including an overview of channel codes for 5G beyond, the motivation of our research, the related works on channel coding, the details on the conducted researches, and the conclusions.

Chapter 1 provides an overview of 5G and beyond wireless communication networks and the design requirements for channel coding techniques, which motivate our research presented in this thesis. It also presents the outline and the main contributions of this thesis.

Chapter 2 provides the fundamental background on channel coding. First, we review the basics of modern point-to-point digital communication systems. Then, different channel models as well as their capacities are presented. In addition, a general overview on the state-of-the-art channel codes and coding techniques is presented.

Chapter 3 reviews the background on turbo codes, including convolutional codes, encoding and decoding of turbo codes, and performance analysis tools. The relevant works on spatially coupled turbo-like codes are also reviewed. The materials presented in Chapter 3 serve as the necessary background for understanding our work throughout Chapters 4 – 6.

Chapter 4 introduces partial coupling techniques, including partial information coupling (PIC) and partial parity coupling (PPC). The coupling is performed such that a fraction of the information/parity sequence of a CB is used as a part of the information sequence of the succeeding CBs. PIC turbo codes (PIC-TCs) and PPC turbo codes (PPC-TCs) are constructed. We derive the exact density evolution functions of PIC-TCs and PPC-TCs, respectively, and obtain the asymptotic iterative decoding thresholds of the proposed codes over binary erasure channels (BECs). We also present the optimization of decoding thresholds. Error performance simulation is also presented, showing that partial coupling greatly improves the decoding performance over turbo codes.

Chapter 5 proposes the partially coupled duo-binary turbo codes (dTCs) to overcome the rate loss problem of partial coupling for PIC-TCs and PPC-TCs. Specifically, a dTC encoder has two inputs, where one input carries the information bits at the current time instance, while another input carries the coupled information/parity bits from the previous time instances. Depending on whether to couple information bits or parity bits, and whether the coupling is performed on the turbo code level or the convolutional code level, we construct four types of partially coupled dTCs. Each construction within this class has its unique characteristics that can be exploited to satisfy different requirements. Both density evolution analysis and simulation results are shown to demonstrate the performance improvement of partially coupled dTCs over partially coupled turbo codes.

Chapter 6 proposes the generalized spatially coupled parallel concatenated convolutional codes (GSC-PCCs), where a fraction of the information bits is repeated for multiple times before coupling. We show that both PIC-TCs and the spatially coupled parallel concatenated convolutional codes (SC-PCCs) in [38] can be seen as special cases of GSC-PCCs. The density evolution analysis demonstrates that the iterative decoding threshold of GSC-PCCs can be very close to the BEC capacity. Additionally, the design criteria to improve finite-length performance of GSC-PCCs are also discussed.

Chapter 7 reviews the fundamental background on polar codes. First, we review the concept of channel polarization phenomenon. Then, the encoding, decoding, and density evolution analysis of polar codes are presented. The materials presented in this chapter will be used throughout Chapters 8 – 9.

Chapter 8 presents the construction of PIC polar codes. The coupling indices design is presented with respect to the polarization property of finite-length polar codes. To be specific, we choose to couple the information bits which have higher error probability than others. The decoding scheme for PIC polar codes is also introduced, and its TB error performance is analyzed. Both theoretical analysis and simulation results show the proposed design greatly improves the decoding performance over finite-length polar codes.

Chapter 9 proposes the PIC polar coding scheme for high-order modulations to show that spatial coupling gain and coded modulation gain can be jointly attained. The joint design of coupling indices and modulation bit-mapper with respect to both the characteristic of polar codes and bit-interleaved coded modulation are discussed. Both theoretical analysis and simulation results are provided to demonstrate the effectiveness of our design.

Chapter 10 concludes the thesis by summarizing the main ideas of each chapter and the contributions of all the works conducted during my Ph. D. studies. Additionally, several future research topics are presented.

1.3.2 Research Contributions

In what follows, a detailed list of publications and my research contributions in Chapters 4 - 6 and Chapters 8 - 9 are presented.

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

[R1] L. Yang, Y. Xie, **X. Wu**, J. Yuan, X. Cheng, and L. Wan, "Information-coupled turbo codes for LTE systems," 2018 IEEE Wireless Communications and Networking Conference (WCNC), Barcelona, 2018, pp. 1-6.

[R2] L. Yang, Y. Xie, **X. Wu**, J. Yuan, X. Cheng, and L. Wan, "Partially Information-Coupled Turbo Codes for LTE Systems," in *IEEE Trans. Commun.*, vol. 66, no. 10, pp. 4381-4392, Oct. 2018.

[R3] M. Qiu, **X. Wu**, A. Graell i Amat, and J. Yuan, "Analysis and Design of Partially Information-and Partially Parity-Coupled Turbo Codes," in *IEEE Trans. Commun.*, vol. 69, no. 4, pp. 2107-2122, April 2021.

[R4] M. Qiu, **X. Wu**, Y. Xie, and J. Yuan, "Density Evolution Analysis of Partially Information Coupled Turbo Codes on the Erasure Channel," *2019 IEEE Information Theory Workshop (ITW)*, Visby, Sweden, 2019, pp. 1-5.

Among these publications, I have partially contributed to the performance analysis, including the derivation of density evolution functions and the numerical computation of BP decoding thresholds. I have also contributed to the implementation of simulation programs.

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

[R5] **X. Wu**, M. Qiu, and J. Yuan, "Partially Information Coupled Duo-Binary Turbo Codes," *2020 IEEE International Symposium on Information Theory (ISIT)*, Los Angeles, CA, USA, 2020, pp. 461-466.

[R6] **X. Wu**, M. Qiu, and J. Yuan, "Partially Coupled Duo-Binary Turbo Codes," in *IEEE Trans. Commun.*. (Submitted for review)

For these two publications, I am the first author. Under the supervision of my supervisors, I have carried out the main part of the work, including performance analysis, simulations, and writing of papers.

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

[R7] M. Qiu, **X. Wu**, A. Graell i Amat, and J. Yuan, "Generalized Spatially Coupled Parallel Concatenated Convolutional Codes With Partial Repetition," *2021 IEEE International Symposium on Information Theory (ISIT).*

[R8] M. Qiu, **X. Wu**, A. Graell i Amat, and J. Yuan, "Generalized Spatially Coupled Parallel Concatenated Convolutional Codes With Partial Repetition," in *IEEE Trans. Commun.*. (Submitted for review)

For these publications, I have contributed to coupling scheme design and simulations.

The results in **Chapter 8** have been presented in the following publications:

[R9] **X. Wu**, L. Yang, and J. Yuan, "Information Coupled Polar Codes," *2018 IEEE International Symposium on Information Theory (ISIT)*, Vail, CO, USA, 2018, pp. 861-865.

[R10] **X. Wu**, L. Yang, Y. Xie, and J. Yuan, "Partially Information Coupled Polar Codes," in IEEE Access, vol. 6, pp. 63689-63702, 2018.

For these publications, I am the first author. Under the supervision of my supervisors, I have carried out the main part of the work, including performance analysis, simulations, and writing of papers.

The results in **Chapter 9** have been presented in the following publications:

[R11] **X. Wu**, and J. Yuan, "Partially Information Coupled Bit-Interleaved Polar Coded Modulation for 16-QAM," *2019 IEEE Information Theory Workshop (ITW)*, Visby, Sweden, 2019, pp. 1-5.

[R12] **X. Wu**, M. Qiu, and J. Yuan, "Partially Information Coupled Bit-Interleaved Polar Coded Modulation for High Order Modulations," in *IEEE Trans. Commun.*, vol. 69, no. 10, pp. 6409-6423, Oct. 2021.

For these publications, I am the first author. Under the supervision of my supervisors, I have carried out the main part of the work, including performance analysis, simulations, and writing of papers.

Chapter 2

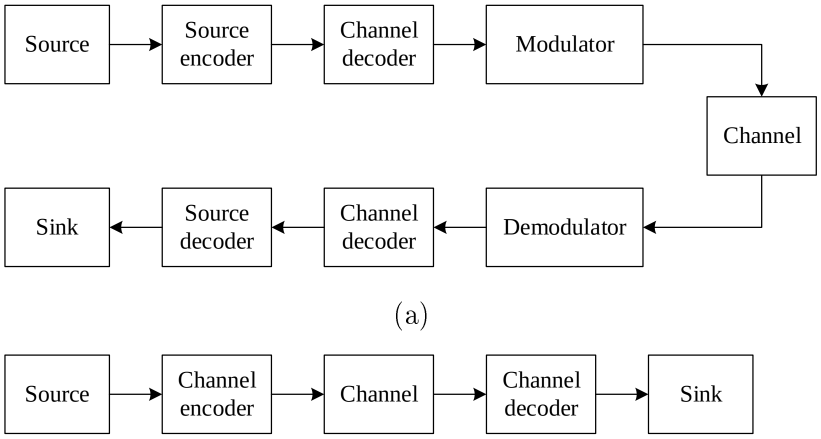
Wireless Communications and Channel Coding

2.1 Introduction

In this chapter, we review the fundamental background on channel coding. We first introduce the system model for digital point-to-point communication. We then review different types of channel models. Lastly, we present background on channel coding and review the state-of-the-art coding techniques. The material in this chapter is a review of the literature, and it serves as the technical guideline to provide the necessary background to understand the works in the later chapters.

2.2 Digital Communication Framework

Consider the digital point-to-point communication scenario, where a *source* transmits its information to a *sink* via a noisy *channel*. In 1948, Shannon established the fundamental point-to-point communication framework [1], depicted in 2.1a. The components of this framework are described as follows.



(a)

(b)

Figure 2.1: Basic point-to-point communication system model. a) Shannon's fundamental system model, and b) simplified channel coding system model.

- *Source encoder* and *source decoder*: The source encoder compresses a sequence of information from the source into an information bit stream. The source decoder is the counterpart of source encoder which recovers the source sequence from the output of channel decoder. This process is called *source coding.*

- *Channel encoder* and *channel decoder*: The channel encoder adds redundancy to the information bit stream and generate the coded bit steam (also called codeword), which is to protect the information against the errors introduced by the channel. The channel decoder is the counterpart of channel encoder, which recovers the information bit stream from the output of demodulator. This process is called *channel coding* (or *error control coding*). The ratio of the length of information bit stream to the length of codeword is called *code rate*, denoted by R, with $0 < R \leq 1$.

- *Modulator* and *demodulator*: The modulator converts a bit stream into the signal which is appropriate for transmission. For a wireless communication channel, the bit steam is converted into analog waveform signal. The demodulator is the modulator's counterpart which recovers a bit steam from received signal.

- *Channel*: The channel is the physical medium (wireless, phone line, optical fiber, etc.) to transmit the signal. Shannon's channel coding theorem shows that a channel can be characterized by a single parameter C, called *channel capacity*, which measures how much information can be transmitted over the channel. In the context of channel coding, Shannon showed that there exists channel codes with code rate R that provide arbitrarily reliable communication over the channel if and only if $R < C$. Conversely, there exists no code that provides reliable communication if $R > C$.

For studying channel coding techniques, this framework is usually simplified into the channel coding system model depicted in Fig. 2.1b. This model omits the source coding system, and considers that modulation and demodulation are a part of the signal transmission process, so that the channel in this framework consists of modulator, demodulator, and the physical medium.

2.3 Common Channel Models for Digital Communication

A communication channel can be modeled as a single-input single-output system. Let $W : \mathcal{X} \rightarrow \mathcal{Y}$ represent a channel with input alphabet $\mathcal{X}$ and output alphabet $\mathcal{Y}$. The *channel transition probability* $P_{Y|X}(y|x)$ is defined as the probability that $y \in \mathcal{Y}$ is received when $x \in \mathcal{X}$ is transmitted. In this thesis, we consider the *symmetric discrete memoryless channel* (S-DMC), which is defined as follows.

- Symmetric: Let $\mathbf{T}$ be the transition probability matrix of a channel, and $T_{i,j} = P_{Y|X}(y_j|x_i)$. A channel is said to be *output symmetric* if all rows of $\mathbf{T}$ are permutations of each other. A channel is said to be *strongly symmetric* if all rows of $\mathbf{T}$ are permutations of each other, and all columns of $\mathbf{T}$ are permutations of each other. In this thesis, a "symmetric channel" refers to an output symmetric channel.
- Discrete: A discrete channel is one whose input alphabet is a discrete set $\mathcal{X} = \{X_1, \ldots, X_l\}$ and output alphabet is a discrete set $\mathcal{Y} = \{Y_1, \ldots, Y_m\}$.
- Memoryless: A channel is memoryless if the channel output at any time instance depends only on the channel input at that time instance.

2.3.1 Channel Capacity

The capacity of a channel W is the highest rate which allows for reliable information transmission over W for sufficiently large code length. Let $P_X(x)$ and $P_Y(y)$ be the probability distribution of channel input x and channel output y respectively. For a DMC channel, the channel capacity is

$$C \triangleq \max_{P_X(x)} I(X;Y), \tag{2.1}$$

where $I(X;Y)$ is the *mutual information* defined as

$$\begin{aligned} I(X;Y) &\triangleq \mathbb{E}\left[\log \frac{P_{Y|X}(y|x)}{P_Y(y)}\right] \\ &= \sum_{y\in\mathcal{Y}} \sum_{x\in\mathcal{X}} P_{Y|X}(y|x) P_X(x) \log \frac{P_{Y|X}(y|x)}{\sum_{a\in\mathcal{X}} P_{Y|X}(y|x) P_X(a)} \end{aligned} \tag{2.2}$$

Note that $I(X;Y)$ is the highest code rate allowing reliable information transmission over W when $P_X(x) = \frac{1}{\|\mathcal{X}\|}$ for all $x \in \mathcal{X}$. It is also called the *symmetric* *capacity* of W, denoted by $I(W)$. For an S-DMC, its symmetric capacity and capacity are the same.

2.3.2 Channel Models and Capacities

Binary Erasure Channel

The *binary erasure channel* (BEC) is a simple model that only considers whether a transmitted bit is received correctly or erased (not received). As illustrated in Fig. 2.2, the BEC has an input alphabet $\mathcal{X} = \{0, 1\}$ and an output alphabet $\mathcal{Y} = \{0, 1, ?\}$, where ? refers to a bit erasure. Let ε be the *erasure probability.* The BEC transition probabilities are $P_{Y|X}(?|0) = P_{Y|X}(?|1) = \varepsilon$ and $P_{Y|X}(0|0) = P_{Y|X}(1|1) = 1 - \varepsilon$. The capacity of the BEC is

$$C_{\mathrm{BEC}}(\varepsilon) = 1 - \varepsilon. \tag{2.3}$$

Binary Symmetric Channel

The *binary symmetric channel* (BSC) has an input alphabet $\mathcal{X} = \{0, 1\}$ and an output alphabet $\mathcal{Y} = \{0, 1\}$. As illustrated in Fig. 2.3, each transmitted bit is either received correctly or flipped to the opposite bit (crossover). Let ε be the *crossover probability.* The BSC transition probabilities are given by $P_{Y|X}(0|1) = P_{Y|X}(1|0) = \varepsilon$ and $P_{Y|X}(0|0) = P_{Y|X}(1|1) = 1 - \varepsilon$. The capacity of the BSC is

$$C_{\mathrm{BSC}}(\varepsilon) = 1 + \varepsilon \log(\varepsilon) + (1 - \varepsilon) \log(1 - \varepsilon). \tag{2.4}$$

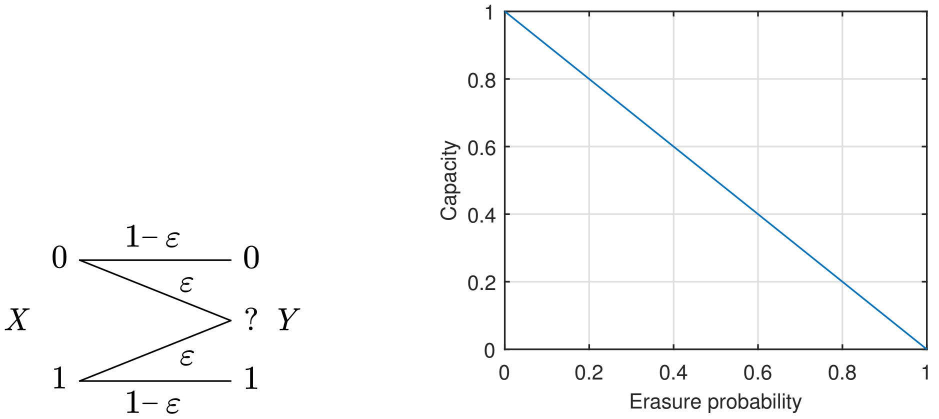


Figure 2.2: Binary erasure channel and its capacity.

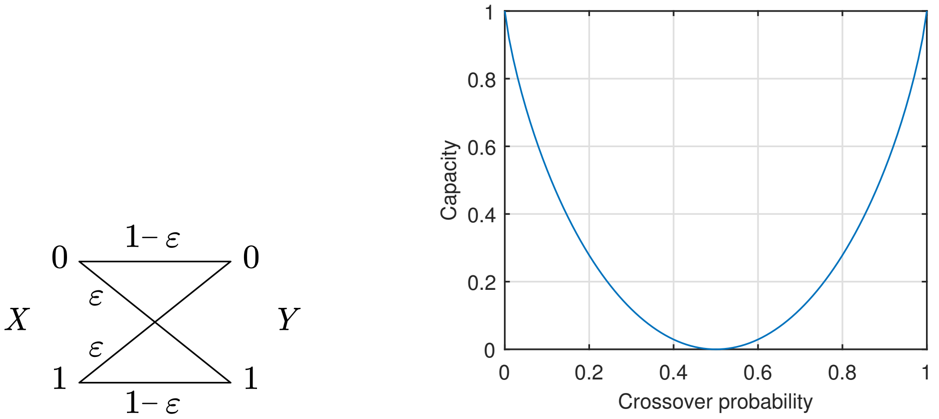


Figure 2.3: Binary symmetric channel and its capacity.

AWGN Channel

The *additive white Gaussian noise* (AWGN) channel is characterized by the *signal-to-noise ratio* (SNR), defined as SNR $\triangleq \frac{E_s}{N_0}$, where E_s is the signal power constrain, and N_0 is the noise spectral density. In this thesis, we consider that the power of transmitted signal is normalized such that $E_s = 1$ and $\frac{E_s}{N_0} = \frac{1}{2\sigma^2}$. The input-output relation of the AWGN channel is

$$y = x + \mathcal{N}_{\mathrm{c}}(0, N_0). \tag{2.5}$$

where $\mathcal{N}_{\mathrm{c}}(0, N_0)$ is a circularly symmetric complex Gaussian random variable with zero mean and variance N_0 such that $\Re\left[\mathcal{N}_{\mathrm{c}}(0, N_0)\right] = \mathcal{N}\left(0, \sigma^2\right)$ and $\Im\left[\mathcal{N}_{\mathrm{c}}(0, N_0)\right] = \mathcal{N}\left(0, \sigma^2\right)$. For an unconstrained AWGN channel, whose input is not restricted to any specific alphabet, the channel capacity is

$$C_{\mathrm{AWGN}}\left(\sigma\right) = \log\left(1 + \frac{1}{2\sigma^2}\right). \tag{2.6}$$

Binary-Input AWGN Channel

The binary-input AWGN (BI-AWGN) channel is also referred to as the *binary phase-shift keying* (BPSK) modulated AWGN channel. As illustrated in Fig. 2.4b, a BI-AWGN channel consists of three components: BPSK modulator, AWGN channel, and BPSK demodulator.

The BI-AWGN channel has an input alphabet $\mathcal{X} = \{0, 1\}$. The channel input $x \in \mathcal{X}$ is mapped to a symbol $s \in \mathcal{S}$ via BPSK modulation, where $\mathcal{S} = \{+1, -1\}$ is the modulation symbol sets. The modulated symbol s is then transmitted via the AWGN channel. The received signal r is demodulated, resulting in a real-valued channel output y. More precisely, the input-output relations are as follows.

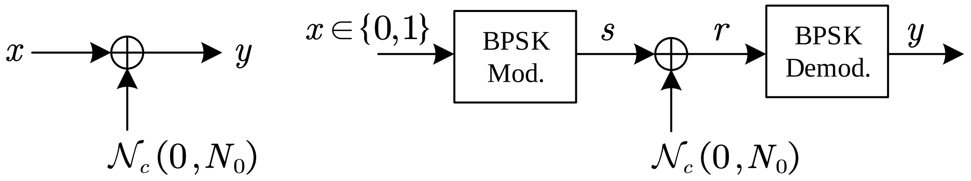
$$\text{BPSK modulation:} \qquad s = 2x - 1, \tag{2.7a}$$

$$\text{AWGN channel:} \qquad r = s + \mathcal{N}_{\mathrm{c}}(0, N_0), \tag{2.7b}$$

$$\text{BPSK demodulation:} \qquad y = \Re\left[r\right] = (2x - 1) + \mathcal{N}(0, \sigma^2). \tag{2.7c}$$

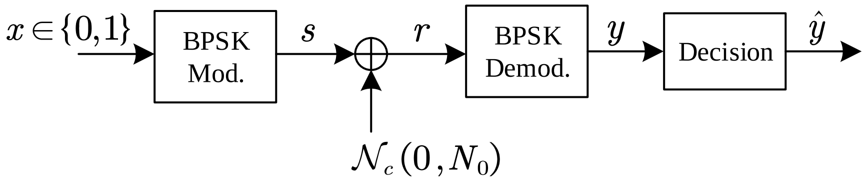
The transition probability of the BI-AWGN channel is

$$P_{Y|X}(y|x) = \Pr\{s + \mathcal{N}(0, \sigma^2)|s\} = \frac{1}{\sigma\sqrt{2\pi}} e^{-(y-s)^2/2\sigma^2}. \tag{2.8}$$



(a) AWGN channel.

(b) Soft-decision BI-AWGN channel.



(c) Hard-decision BI-AWGN channel.

Figure 2.4: AWGN channels.

In most cases, the channel output is expressed in terms of log-likelihood-ratio (LLR), defined as

$$L_{ch}(y) \triangleq \ln \frac{P_{Y|X}(y|x=0)}{P_{Y|X}(y|x=1)} = -\frac{4yE_s}{N_0} = -\frac{2y}{\sigma^2} \tag{2.9}$$

The absolute value of $L_{ch}(y)$ is a Gaussian random variable $|L_{ch}(y)| \sim \mathcal{N}(\frac{2}{\sigma^2}, \frac{4}{\sigma^2})$. The capacity of the BI-AWGN channel is

$$C_{\text{BI-AWGN}}(\sigma) = \frac{1}{\sqrt{2\pi}} \int e^{-z^2/2} \left(1 - \log\left(1 + e^{-(2/\sigma^2)+(2z/\sigma)}\right)\right) dz. \tag{2.10}$$

Hard-Decision Binary-Input AWGN Channel

The hard-decision BI-AWGN channel refers to that the soft-decision output y of an BI-AWGN channel is processed with a decision union, resulted in the hard-decision output $\hat{y}$ according to

$$\hat{y} = \begin{cases} 0 & \text{if } y \leq 0, \\ 1 & \text{if } y > 0. \end{cases} \tag{2.11}$$

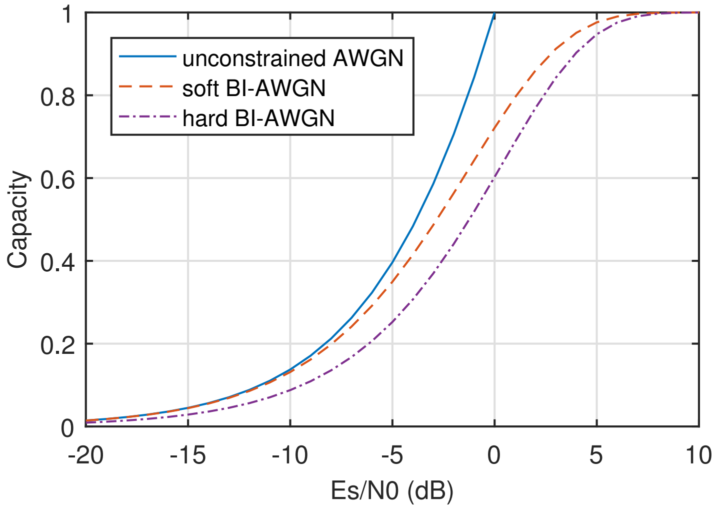


Figure 2.5: AWGN channel capacity.

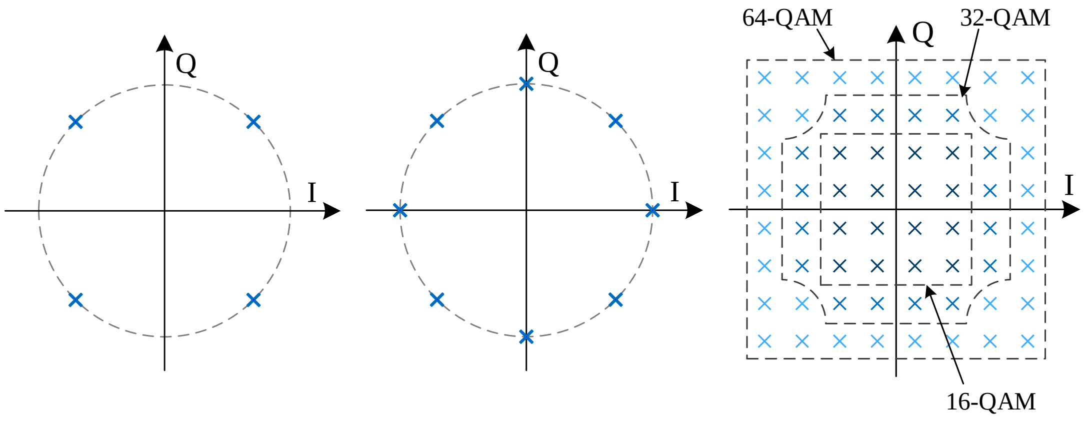
The hard-decision converts the BI-AWGN channel to the BSC with crossover probability $\varepsilon = Q(1/\sigma)$, where the Q-function is defined as

$$Q(x) \triangleq \Pr\{\mathcal{N}(0,1) > x\} = \frac{1}{\sqrt{2\pi}} \int_{x}^{\infty} e^{-z^2/2}\, dz \tag{2.12}$$

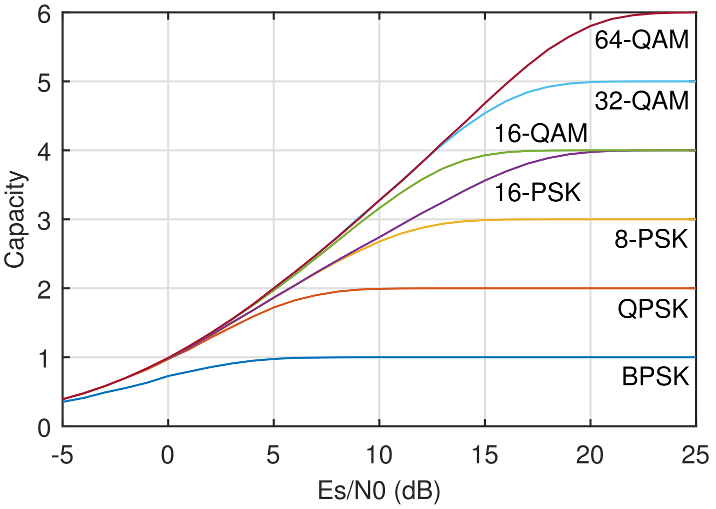
Thus, the capacity of the hard-decision BI-AWGN channel can be obtained from the capacity of its equivalent BSC. Note that converting soft-decision to hard-decision leads to a capacity loss. To demonstrate that, in Fig. 2.5, we compare the capacities of unconstrained AWGN, BI-AWGN, and hard-decision BI-AWGN, respectively.

AWGN Channel with 2^m-ary Modulations

Let m be a positive integer and $M = 2^m$. Let $\mathcal{S} = \{S_1, \ldots, S_M\}$ be the constellation symbol set of an M-ary modulator whose modulation order is m. The modulator maps an m-bit sequence $[x_1, \ldots, x_m]$ to one symbol $y \in \mathcal{Y}$. The mapping from bit-sequence to symbol can be described by the constellation labeling.



(a) QPSK (b) 8-PSK (c) 16-QAM – 64-QAM



(d) Capacities of AWGN channels with PSK and QAM.

Figure 2.6: Examples on PSK and QAM.

The high-order modulations that are commonly used in wireless communication systems include *phase-shift keying* (PSK), which uses phase (in terms of polar coordinate) to define symbols, and *quadrature amplitude modulation* (QAM), which uses both phase and amplitude to define symbols. Examples of PSK and QAM are illustrated in Figs. 2.6a – 2.6c, while the capacities of the AWGN channels with PSK and QAM are shown in Fig. 2.6d.

2.4 Channel Coding

Channel encoding can be described as an injective mapping. Consider a binary (N, K) channel code with information length K and codeword length N. The encoder maps a length-K information sequence $\mathbf{u} \in \{0, 1\}^K$ to a codeword $\mathbf{x} \in \mathcal{C}$. The output alphabet $\mathcal{C} \subset \{0, 1\}^N$ is called the codebook of this code, and $\|\mathcal{C}\| = 2^K$. In practice, it is considered that information bits are randomly generated with $\Pr\{u_i = 0\} = \Pr\{u_i = 1\}$ for any $u_i \in \mathbf{u}$ so that the codewords are equally likely, i.e, $P_X(x) = 1/2^K$.

Some terms to describe the property of channel codes are list in the following.

- *Linear code*: A (N, K) linear code is a vector subspace with dimension K of the vector space $\mathbb{F}_q^N$, where $\mathbb{F}_q$ is the finite field with q elements. The linear codes with $q = 2$ are called binary codes. Linear codes exhibit the following properties:
    - The sum of two codewords is another codeword (closed under vector addition).
    - The zero vector is a codeword.
    - The number of codewords in a (N, K) linear code under $\mathbb{F}_q^N$ is q^K.
- *Systematic code*: A code is said to be in systematic form if its codeword $\mathbf{x}$ for any input sequence $\mathbf{u}$ consists of two parts $\mathbf{x} = [\mathbf{u}, \mathbf{p}]$. The first part $\mathbf{u}$ is called the systematic bits, and the second part $\mathbf{p}$ is called the parity bits (or redundant bits).

Before we process to the overview of channel coding, we would to emphasize that we only consider binary linear codes in this thesis.

2.4.1 Error Control Operation

Channel coding techniques may fall into two categories: *automatic repeat request* (ARQ) schemes and *forward error correction* (FEC) schemes. In ARQ schemes, the receiver detects whether a received packet contains errors with an *error-detection code.* When errors are detected, the receiver requests for transmitting the same packet. This process may be repeated until no error is detected or the maximum time of retransmission is reached. In FEC schemes, the receiver tries to correct the errors in the received packet through the decoding process. The codes devised for this case are called *error-correction codes.* There also exist *hybrid-ARQ* (HARQ) schemes which employ FEC and ARQ jointly. In such scheme, the receiver carries out error correction first and then detects whether all errors are corrected. When decoder fails to correct the errors, the receiver requests for the retransmission.

TB-based HARQ

TB based HARQ is employed in both 4G [39] and 5G [40] standards. The information sequence in a TB is segmented into multiple sub-sequences of the same length. Each sub-sequence is encoded by a finite-length channel code referred to as the code block (CB). Once errors appear in any of the CBs, the entire TB will be retransmitted.

Error Performance Measures

One of the most commonly used error performance measures is the *bit error rate* (BER), defined as the ratio of number of bits received in error to the total number of bits transmitted. Another commonly used error performance measure is the *block error rate* (BLER), defined as the ratio of number of blocks (i.e., information sequences) received with error to the total number of blocks transmitted.

When considering the BEC, *bit erasure rate* is commonly used to measure the error performance instead of bit error rate. It is defined as the ratio of number of erased bits to the total number of bits transmitted. General speaking, bit error rate is half of the bit erasure rate as 0 and 1 are generated with equal probabilities.

In the context of TB-based transmission, *TB error rate* (TBER), defined as the ratio of number of TBs received in error to the total number of TBs transmitted, is also considered as a performance measure.

2.4.2 Decoder Design Criteria

The decoder of a (N, K) channel code inputs the received signal $\mathbf{y} = [y_1, \ldots, y_N]$ and output the hard-decision estimation $\hat{\mathbf{u}} = [\hat{u}_1, \ldots, \hat{u}_K]$. There exist two commonly used decoder design criteria: *maximum a-posteriori* (MAP) and *maximum-likelihood* (ML) decoding. Both of them are optimal design criteria minimizing the probability of codeword error, but their optimization objectives are distinct. Here, a codeword error refers to that the channel decoder faills to recover the correct codeword.

As its name implies, MAP decoding is to maximize the a-posteriori probability $P_{X|Y}(\mathbf{x}|\mathbf{y}) = \prod_{i=1}^{N} P_{X|Y}(x_i|y_i)$ that $\mathbf{x}$ is transmitted when $\mathbf{y}$ is received, i.e., $\hat{\mathbf{u}} = \arg\max_{\mathbf{u}} P_{X|Y}(\mathbf{x}|\mathbf{y})$, where $\mathbf{u} \in \{0,1\}^K$, and $\mathbf{x} = \mathrm{enc}(\mathbf{u})$ is the codeword corresponding to $\mathbf{u}$. ML decoding is to maximize the transition probability $P_{Y|X}(\mathbf{y}|\mathbf{x}) = \prod_{i=1}^{N} P_{Y|X}(y_i|x_i)$ that $\mathbf{y}$ is received when $\mathbf{x}$ is transmitted. That is, $\hat{\mathbf{u}} = \arg\max_{\mathbf{u}} P_{Y|X}(\mathbf{y}|\mathbf{x})$, $\mathbf{u} \in \{0,1\}^K$, and $\mathbf{x} = \mathrm{enc}(\mathbf{u})$. Following Bayes' rule, the a-posteriori probability can be obtained by $P_{X|Y}(\mathbf{x}|\mathbf{y}) = \frac{P_{Y|X}(\mathbf{y}|\mathbf{x})P_X(\mathbf{x})}{P_Y(\mathbf{y})}$. Due to the fact that $P_X(\mathbf{x})$ and $P_Y(\mathbf{y})$ are both independent of $\mathbf{u}$, MAP decoding rule can be rewritten as

$$\hat{\mathbf{u}} = \arg\max_{\mathbf{u}} P_{X|Y}(\mathbf{x}|\mathbf{y}) = \arg\max_{\mathbf{u}} \frac{P_{Y|X}(\mathbf{y}|\mathbf{x})P_X(\mathbf{x})}{P_Y(\mathbf{y})} = \arg\max_{\mathbf{u}} P_{Y|X}(\mathbf{y}|\mathbf{x}). \tag{2.13}$$

This implies that MAP decoding and ML decoding have the same decoding performance.

In addition, bit-wise MAP decoding is also widely adopted as a criterion which minimizes the probability of bit error. It is given by

$$\hat{u}_i = \arg\max_{u_i \in \{0,1\}} P(u_i|\mathbf{y}), \tag{2.14}$$

where $P(u_i|\mathbf{y})$ is the a-posteriori probability that the i-th bit of the information sequence equals u_i.

2.4.3 State-of-the-Art Channel Codes

Linear Block Codes

Linear block codes are a class of codes which take a length-K input sequence and generate a length-N codeword via linear mapping. Many well known codes such as *Hamming codes* [41] and *Bose-Chaudhuri-Hocquenghem* (BCH) codes [42, 43] are in this class. A linear block code $\mathcal{C}(N, K)$ can be described by its *generator matrix* $\mathbf{G}$ of size $K \times N$. Let $\mathbf{u} = (u_1, \ldots, u_K)$ represent an input sequence, and $\mathbf{x} = (x_1, \ldots, x_N)$ represent a codeword. The encoding process can be described as

$$\mathbf{x} = \mathbf{u} \cdot \mathbf{G}, \tag{2.15}$$

which means that the codeword of $\mathcal{C}$ is given by a linear combination of the rows of $\mathbf{G}$. This code can also be described by its *parity-check matrix* $\mathbf{H}$ of size $(N-K)\times N$ such that

$$\mathbf{G}\cdot\mathbf{H}^{\mathrm{T}}=\mathbf{O}, \tag{2.16}$$

where $\mathbf{O}$ is a all-zero matrix of size $K\times(N-K)$.

Cyclic Redundancy Checks Codes

Cyclic redundancy checks (CRCs) [44] codes refer to a class of linear block codes proposed for error detection propose. At the transmitter, check bits are produced based on the remainder of a binary division between the information bits and a predetermined divisor. At the receiver, the calculation is repeated. An error is claimed when the check bits produced by the receiver do not agree with that produced by the transmitter.

Convolutional Codes

Introduced by Elias in 1955 [45], *convolutional codes* refer to a class of codes generated by the encoder with memory. Consider a convolutional code encoder with memory order m. At any time instance, the encoder output depends on the current input as well as the inputs in the previous m time instances. Such a structural property makes convolutional codes applicable for arbitrary information lengths. Viterbi algorithm [46] and BCJR algorithm [47] are widely adopted to decode convolutional codes. Convolutional codes bring a breakthrough in coding theory as these codes have simpler encoder structure than generic linear blocks codes while having same average performance as randomly chosen linear blocks codes.

As we use convolutional codes as one of the components to build our proposed codes in Chapter 4 - 6, we will give a detailed literature review on convolutional codes in Chapter 3.

Turbo Codes

Turbo codes are a class of high-performance error correction codes introduced by Berrou et.al in 1993 [48]. They were the first class of practical codes that closely approach the channel capacity. Turbo codes are constructed by concatenating two convolutional codes in parallel. One key factor making turbo codes capacity approaching is the adoption of iterative decoding, where the component codes exchange soft decoding messages between each other over iterations. As the number of iterations increases, the estimated sequence gets closer to the original information sequence.

As we use turbo codes to build our proposed codes in Chapters 4 – 6, we will present the detailed literature review on turbo codes in Chapter 3.

LDPC Codes

Low-density parity-check (LDPC) codes, introduced by R. Gallager in 1962 [19], are a class of linear block codes which allow iteratively decoding. In mid-1990s, MacKay et.al [20, 21] showed that LDPC codes have some features similar with the turbo codes and that LDPC codes can approach channel capacity. As their name implies, the parity-check matrix of a LDPC code contains only a very small number of non-zero entries. Such characteristic suppresses the iterative decoding complexity so that it only grows linearly with the increment of code length.

Polar Codes

Polar codes are a class of linear block codes proposed by E. Arikan in 2009 [27]. This class of codes have been theoretically proved to be capacity-achieving for S-DMC while having low-complexity construction, encoding, and decoding algorithms. Polar codes convert N copies of a channel into a set of N correlated bit-channels whose capacity approaches either 0 or 1. By transmitting information via the perfect bit-channels, polar codes approach the capacity of an S-DMC when N approaches infinity.

In Chapter 7, we will present the literature review on polar codes. In Chapters 8 – 9, We will use polar codes to build our proposed codes.

Coded Modulation

Coded modulation refers to the techniques that combine channel coding and higher order modulations. A typical example of coded modulation is the *trellis-coded modulation* (TCM) [49, 50], which combines the design of convolutional encoder and modulation labeling. Apart from TCM, there are two broad categories of code modulation schemes which can use various channel codes as the constituent code. One is *multilevel coding* (MLC) [51], where multiple codewords with different code rates are modulated simultaneously such that each codeword is transmitted via one of the bit-positions of the modulated symbols. Another is *bit-interleaved coded modulation* (BICM) [52], which only uses a single channel code and each codeword is independently modulated and transmitted. In recent literature [53], *delayed BICM* (D-BICM) was proposed as a generalization of BICM. Unlike BICM where each codeword is independently modulated and transmitted in a single time slot, D-BICM divides each codeword into multiple subblocks and modulates the subblocks from both the previous time slots and the current time slot onto the same signal sequence.

2.4.4 Codes on Graphs

The field "*codes on graphs*" is not a specific class of codes but a class of graphical representations to describe the structure of various codes. In [54], which is the work founded this field, *Tanner graph* was introduced, providing a bipartite graphical model for LDPC codes. A Tanner graph consists of two types of nodes:

1. *Variable nodes* (VN) (or *bit nodes*): each VN represents a coded bit (a bit in the codeword), plotted as a filled circle in the graph.
2. *Check nodes* (CN): each CN represent a parity check function, plotted as a square in the graph.

Edges are used to connect VNs to CNs. Multiple VNs connecting with a CN means that the sum of their corresponding coded bits is 0.

The number of edges connected to a node is called the degree of that node. Based on the degree distribution, LDPC codes can be classified into *regular codes* and *irregular codes*. A regular code is one that all VNs have the same degree d_v and all CNs have the same degree d_c. An irregular code is one that nodes of the same type may have different degrees.

Example 2.1. Consider the parity-check matrix of a regular LDPC code with $(d_v, d_c) = (2, 4)$.

$$\mathbf{H} = \begin{bmatrix} \mathbf{h}_1 \\ \mathbf{h}_2 \\ \mathbf{h}_3 \\ \mathbf{h}_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \end{bmatrix}. \tag{2.17}$$

Given a codeword $\mathbf{x} = [x_1, \ldots, x_8]$, $\mathbf{H}$ satisfies that $\mathbf{x} \cdot \mathbf{H}^{\mathrm{T}} = \mathbf{0}$. This constraint

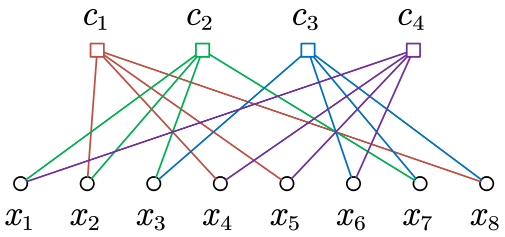


Figure 2.7: Tanner graph of a LDPC code.

can also be written as a group of parity check functions $\{c_1, \ldots, c_4\}$, where

$$c_i : \mathbf{x} \cdot \mathbf{h}_i^{\mathrm{T}} = \sum_{j=1}^{8} x_j \cdot h_{i,j} = 0 \tag{2.18}$$

for $1 \leq i \leq 4$. The corresponding tanner graph is shown in Fig. 2.7. The columns and rows in **H** correspond to VNs and CNs respectively. If $h_{i,j} = 1$, then there is an edge connecting the VN representing x_j and the CN representing c_i. ■

Subsequently, *factor graph* was introduced in [55] as an extension of Tanner graphs. By introducing state nodes, factor graphs can describe trellis-based codes such as convolutional codes and turbo codes. With the help of factor graphs, it was shown that turbo codes are related to LDPC codes.

In [56], *protograph* representation of a code ensemble was introduced. A protograph is a Tanner graph with a relatively small number of nodes. Additionally, multiple parallel edges are permitted. Moreover, a “copy-and-permute” operation was proposed to use protograph to design LDPC codes of arbitrary size.

Belief Propagation Decoding

The outstanding error performance of codes on graphs comes from the *message passing decoding* algorithms, where the decoding messages are exchanged between the CNs and VNs iteratively via the edge connected between them. A typical algorithm in this class is the *belief propagation* [55, 57] (BP) algorithm, which is also called sum-product algorithm.

In a Tanner graph, each node (VN/CN) can be seen as one single constituent decoder. The VNs and CNs on each side of the tanner graph can be seen as a set of VN decoders concatenated through an interleaver to a set of CN decoders. The VNs and CNs work cooperatively to estimate the LLR for each coded bit.

The LLR of a bit x is defined as

$$L(x) \triangleq \ln \frac{\Pr\{x=0\}}{\Pr\{x=1\}} \tag{2.19}$$

The a-posteriori LLR $L(x)$ (i.e., the soft-decision estimation) of a bit x given by a VN/CN is the summation of three terms: a-priori LLR $L_a(x)$, channel LLR $L_{ch}(x)$, and extrinsic LLR $L_e(x)$, written as

$$L(x) = L_a(x) + L_{ch}(x) + L_e(x). \tag{2.20}$$

In this expression, $L_a(x)$ and $L_{ch}(x)$ are the decoder inputs, while $L_e(x)$ is the decoder output. To be precise, $L_a(x)$ is the estimation of x produced by other decoders, $L_{ch}(x)$ is the channel transition probability of x, and $L_e(x)$ is the estimation of x made by the current decoder. That is, the extrinsic LLR of one decoder will become the a-priori LLR of the other decodes.

In the iterative decoding process, each iteration starts with the VN side. First, the VNs process their inputs (the a-priori LLRs from CNs and the channel LLRs) and pass its extrinsic LLR to the CNs they are connected with. Consider a VN x_i that is connected with CNs $\{c_j : j \in \mathcal{S}\}$ and let L_i be the channel LLR of x_i, $L_{i \leftarrow j}$ be the LLR sent from c_j to x_i. The extrinsic LLR from x_i to c_j is updated by

$$L_{i\to j} = L_i + \sum_{j' \in \mathcal{S}\setminus j} L_{i\leftarrow j'}. \tag{2.21}$$

Next, the CNs process their inputs (the a-priori LLRs from VNs) and pass its extrinsic LLR to the VNs they are connected with. Consider a CN c_j that is connected with VNs $\{x_i : i \in \mathcal{S}\}$ and let $L_{i\to j}$ be the LLR sent from x_i to c_j. The extrinsic LLR from c_j to x_i is updated by

$$L_{i\leftarrow j} = \underset{i' \in \mathcal{S}\setminus i}{\boxplus} L_{i'\to j} = 2\tanh^{-1}\left(\prod_{i' \in \mathcal{S}\setminus i} \tanh\left(\frac{L_{i'\to j}}{2}\right)\right), \tag{2.22}$$

where box-plus operation is defined as $a \boxplus b = 2\tanh^{-1}\left(\tanh\left(\frac{a}{2}\right) \cdot \tanh(\frac{b}{2})\right)$, and can be approximated by $a \boxplus b \approx \operatorname{sign}(a, b) \cdot \min(|a|, |b|)$.

The VN-CN process repeats until a preset maximum number of iterations or a stopping criterion is met. Finally, the VNs compute the a-posteriori LLRs of the coded bits using

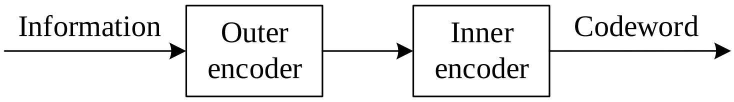
$$L(x_i) = L_i + \sum_{j \in \mathcal{S}} L_{i\leftarrow j}. \tag{2.23}$$

2.4.5 Coding Techniques for Constructing Long Codes

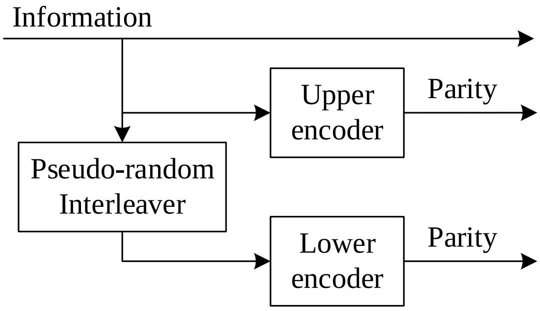
In this section, we present several widely adopted coding techniques to construct powerful long codes from short component codes.

Concatenated Codes

Concatenated codes are codes build from two or more simple codes in order to achieve good performance with reasonable complexity. There exist two methods



(a) Serial concatenated codes.



(b) Parallel concatenated codes.

Figure 2.8: Encoder block diagram of concatenated codes.

to build concatenated codes: serial concatenation and parallel concatenation.

Fig. 2.8a depicts the encoder block diagram of a serial concatenated code, where the output of the outer encoder becomes the input of the inner encoder, and the output of the inner encoder is the codeword to be transmitted. One representative design is to concatenate an outer Reed-Solomon code to an inner convolutional code, which was firstly used in Voyager 2 space probe [58] and is widely adopted for deep space communication [59].

Fig. 2.8b shows the encoder block diagram of a parallel concatenated code. The input of the upper encoder is interleaved and becomes the input of the lower encoder. The codeword consists of the information bits as well as the parity bits from both component encoder. One typical class of parallel concatenated codes is turbo codes, which are uses in 3G and 4G mobile communication standards [10–12].

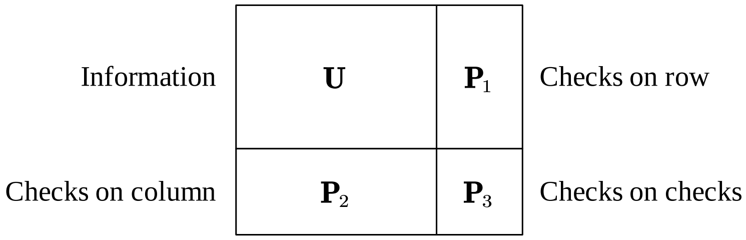


Figure 2.9: Structure of product codes.

Product Codes

A *product code* $\mathcal{C}(N, K)$ is constructed by two smaller linear block codes $\mathcal{C}_1 = (n_1, k_1)$ and $\mathcal{C}_2 = (n_2, k_2)$, where $N = n_1 n_2$ and $K = k_1 k_2$. Encoding of product codes can be accomplished in two stages. As shown in Fig. 2.9, the information bits are arranged as a $k_2 \times k_1$ matrix $\mathbf{U}$. At the first stage, each row of $\mathbf{U}$ is encoded with C_1. The resulting parity bits forms a matrix $\mathbf{P}_1$ of size $k_2 \times (n_1 - k_1)$, referred to as the *checks on rows*. At the second stage, each column of $\mathbf{U}$ is encoded with $\mathcal{C}_2$, giving the parity bits matrix $\mathbf{P}_2$ of size $k_1 \times (n_2 - k_2)$, referred to as the *checks on rows*. Meanwhile, each column of P_1 is encoded with $\mathcal{C}_2$, giving the parity bits matrix $\mathbf{P}_3$ of size $(n_2 - k_2) \times (n_1 - k_1)$, referred to as the *checks on checks*.

Spatially Coupled Codes

Spatial coupling is a code construction technique that connects a sequence of component codes to form a long codeword chain. It was firstly introduced in [35] for constructing convolutional LDPC codes, also known as spatially coupled LDPC (SC-LDPC) codes. Extensive research has been conducted on SC-LDPC codes (see [60–62] and the references therein). Most notably, it has been theoretically proven in [60] that SC-LDPC codes under belief propagation decoding can achieve the MAP decoding threshold of their underlying component codes, namely the threshold saturation phenomenon. Another advantage of spatially

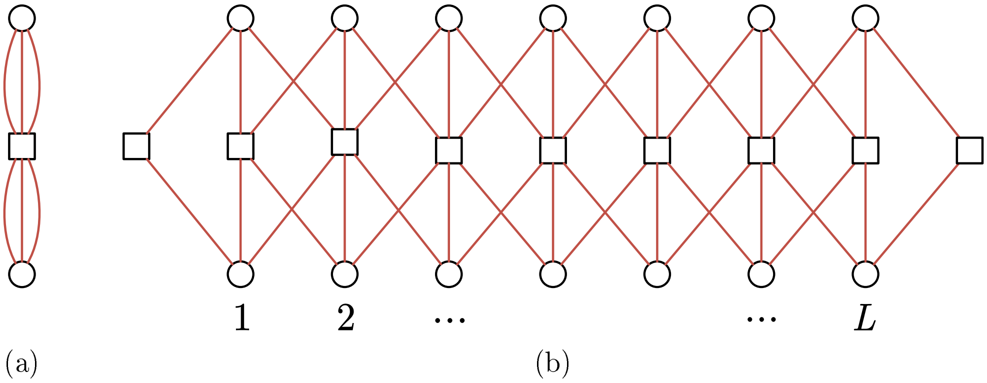


Figure 2.10: Protograph of a) a regular $(3, 6)$ LDPC code and b) a terminated SC-LDPC code constructed with L copies of regular $(3, 6)$ LDPC code.

coupled codes is that they can be decoded by a sliding window decoder [36] with a lower decoding latency compared to decoding the whole codeword block. Thus, spatially coupled codes have become very popular in a wide range of applications in communications.

Fig. 2.10 shows an example of constructing SC-LDPC codes using a regular $(3, 6)$ LDPC code. Given L copies of the $(3, 6)$ LDPC code, the edge at the t-th copy are spread to the $t - 1$-th copy and $t + 1$-th copy to form a long code. At each end of the coupling chain, the edges are terminated so that the CNs at two ends have a smaller degree. In the context of BP decoding, CNs with a smaller degree can pass more reliable messages to its neighboring VNs [60, 61]. As decoding iteration increases, the reliable messages propagate throughout the nodes in coupling chain.

The edge-spreading (also called unwrapping) procedure can be described by splitting a parity-check matrix $\mathbf{H}$ into m sub-matrices $\mathbf{H}_0, \ldots, \mathbf{H}_{m-1}$ such that

$\mathbf{H} = \sum_{i=0}^{m} \mathbf{H}_i$. The convolutional parity-check matrix becomes

$$\mathbf{H}_{cc} = \begin{bmatrix} \mathbf{H}_0 & & & \\ \mathbf{H}_1 & \mathbf{H}_0 & & \\ \vdots & \mathbf{H}_1 & \ddots & \\ \mathbf{H}_{m-1} & \vdots & \ddots & \mathbf{H}_0 \\ & \mathbf{H}_{m-1} & & \mathbf{H}_1 \\ & & \ddots & \vdots \\ & & & \mathbf{H}_{m-1} \end{bmatrix}. \tag{2.24}$$

For the case shown in Fig. 2.10, where the parity-check matrix can be written as $\mathbf{H} = [3,\ 3]$, the edge spreading is given by $\mathbf{H} = \mathbf{H}_0 + \mathbf{H}_1 + \mathbf{H}_2 = [1,\ 1] + [1,\ 1] + [1,\ 1]$. The convolutional parity-check matrix is

$$\mathbf{H}_{cc} = \begin{bmatrix} 1 & 1 & & & & & \\ 1 & 1 & 1 & 1 & & & \\ 1 & 1 & 1 & 1 & \ddots & & \\ & & 1 & 1 & \ddots & 1 & 1 \\ & & & & \ddots & 1 & 1 \\ & & & & & 1 & 1 \end{bmatrix}. \tag{2.25}$$

Apart from LDPC codes, spatial coupling techniques have also been applied to turbo-like codes [38, 63–65] and polar codes [66–69]. In Chapters 4 – 6, we will present our proposed spatially coupled turbo-like codes. In Chapters 8 – 9, we will present our proposed spatially coupled polar codes.

Chapter 3

Introduction to Turbo Codes

Turbo codes [48] are widely employed in various communication systems including the 3G and 4G mobile communications [10–12], WiMAX [23], and deep space communications [70]. In this chapter, we present the fundamental background on turbo codes, including convolutional codes, encoding and decoding of turbo codes, and performance analysis tools. The relevant spatially coupled turbo-like codes [38] are also reviewed. The material in this chapter is summarized from the literature and will be used as necessary background for the work throughout Chapters 4 – 6.

3.1 Convolutional Codes

We first start with convolutional codes [45], which are the component codes of turbo codes.

3.1.1 Encoding

Consider a rate-$\frac{k}{n}$ convolutional code with memory order m. Let $\mathbf{u} = (\mathbf{u}_1, \mathbf{u}_2, \ldots)$ be the information sequence and $\mathbf{v} = (\mathbf{v}_1, \mathbf{v}_2, \ldots)$ be the coded sequence. At time

t, the encoder inputs a k-bit information sequence $\mathbf{u}_t = (u_t^{(1)}, \ldots, u_t^{(k)})$. Then, it outputs an n-bit coded sequence $\mathbf{v}_t = (v_t^{(1)}, \ldots, v_t^{(n)})$, which is produced depending on not only the current input $\mathbf{u}_t$ but also the previous inputs $\mathbf{u}_{t-m}, \ldots, \mathbf{u}_{t-1}$ over m time instants. The convolutional code can be described by the generator matrix

$$\mathbf{G} = \begin{bmatrix} \mathbf{g}_1^{(1)} & \mathbf{g}_1^{(2)} & \cdots & \mathbf{g}_1^{(n)} \\ \mathbf{g}_2^{(1)} & \mathbf{g}_2^{(2)} & \cdots & \mathbf{g}_2^{(n)} \\ \vdots & \vdots & \cdots & \vdots \\ \mathbf{g}_k^{(1)} & \mathbf{g}_k^{(2)} & \cdots & \mathbf{g}_k^{(n)} \end{bmatrix}, \tag{3.1}$$

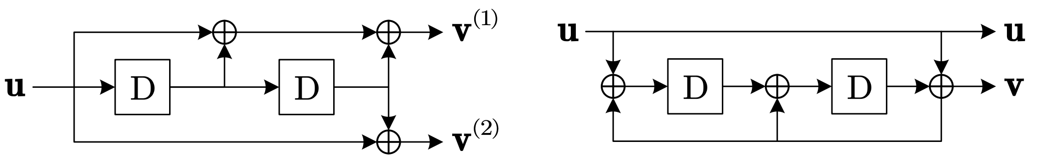
where $\mathbf{g}_i^{(j)} = [g_{i,1}^{(j)}, \ldots, g_{i,m}^{(j)}]$ represents the generator vector corresponding to the i-th input bit and j-th output bit, $1 \leq i \leq k$, and $1 \leq j \leq n$. The input-output relation at time t can then be expressed by

$$v_t^{(j)} = \sum_{i=1}^{k} \sum_{\delta=0}^{m} u_{t-\delta}^{(i)} \cdot g_{i,\delta}^{(j)} \tag{3.2}$$

Recursive systematic convolutional (RSC) codes are a class of convolutional codes commonly employed for constructing turbo codes. The definitions are as follows.

- A convolutional code is said to be in *systematic* form if its code sequences satisfy $v_t^{(i)} = u_t^{(i)}$ for $1 \leq i \leq k$.
- A convolutional code is said to be in *recursive* form if its encoder consists of a feedback component.

Example 3.1. Fig. 3.1a shows the encoder block diagram of a rate-$\frac{1}{2}$ convolutional code with generator $\mathbf{G} = [\mathbf{g}_1^{(1)}, \mathbf{g}_1^{(2)}] = [(1\ 1\ 1), (1\ 0\ 1)]$. At time t, the



(a) Non-systematic realization. (b) Systematic realization.

Figure 3.1: Convolutional code encoder block diagram for $\mathbf{G} = [7\ 5]$.

encoder implies the convolution operation

$$v_t^{(1)} = u_t + u_{t-1} + u_{t-2} \tag{3.3a}$$

$$v_t^{(2)} = u_t + u_{t-2} \tag{3.3b}$$

We may also express the generator in octal-form [7 5] or in the polynomial form $[1 + D + D^2, 1 + D^2]$, where D represents the discrete-time delay operator.

Fig. 3.1b shows the systematic encoder realization of [7 5], which results in an RSC code. In this case, the generator can be rewritten as $\mathbf{G}_{\text{sys}} = [1\ \frac{\mathbf{g}_f}{\mathbf{g}_b}] = [1\ \frac{5}{7}]$, where $\mathbf{g}_f$ is the forward generator vector, $\mathbf{g}_b$ is the feedback generator vector. ■

3.1.2 Trellis Diagram

Mathematically, a convolutional encoder is modeled as a finite state machine. The encoder state is defined as the bit sequence stored in the delay units at a time instant, so that an encoder with memory order m has a total of $M = 2^m$ states $\mathcal{S} = \{S_i = \text{de2bi}(i) : 0 \leq i \leq M - 1\}$, where de2bi($i$) is the function to convert i to the m-bit binary expression. The input-output relation and state transition of an encoder can be described by a *state-transition table* or a *state diagram*. The time evolution of the encoder given an input sequence $\mathbf{u}$ can be described by a *trellis diagram*.

Example 3.2. Consider the convolutional encoder given by $\mathbf{G} = [7\ 5]$, whose

Table 3.1: State-transition table for $\mathbf{G} = [7\ 5]$.

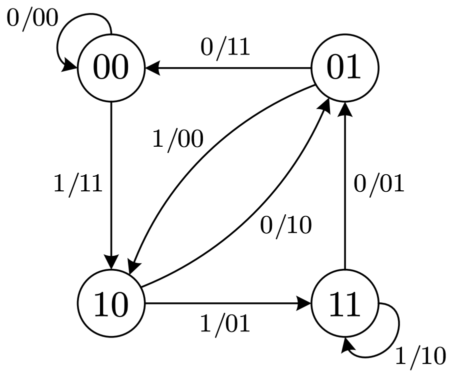
Input u_t	Current state $s_{t,1}s_{t,2}$	Next State $s_{t+1,1}s_{t+1,2}$	Output $v_t^{(1)}v_t^{(2)}$
0	00	00	00
1	00	10	11
0	01	00	11
1	01	10	00
0	10	01	10
1	10	11	01
0	11	01	01
1	11	11	10

encoder block diagram is shown in Fig. 3.1a. Let $s_t = [s_{t,1}, s_{t,2}]$ be the encoder state at time t, when encoder input u_t, the encoder state will transit to $s_{t+1} = [s_{t+1,1}, s_{t+1,2}]$ at time $t+1$, where $s_{t+1,1} = u_t$ and $s_{t+1,2} = s_{t,1}$. Knowing the relation between the encoder states and the encoder input, we may produce the state-transition table, shown in Table 3.1. The corresponding state-transition diagram is depicted in Fig. 3.2a, where edge labels show the input-output relation $u_t/v_t^{(1)}v_t^{(2)}$ for each state transition.

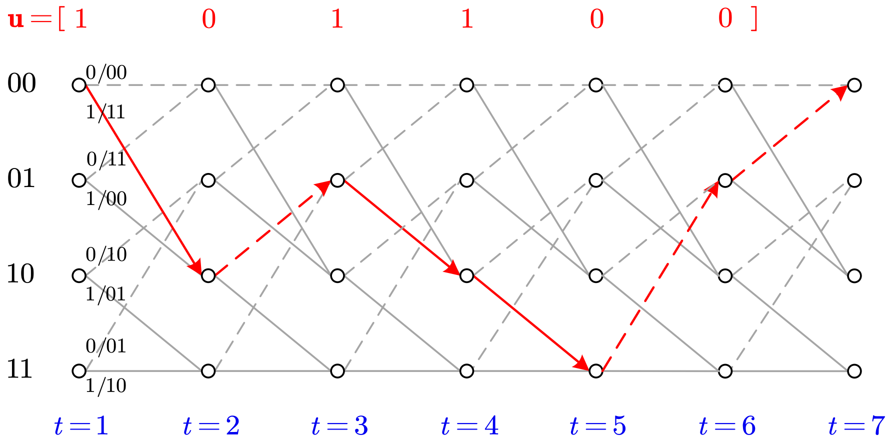
Fig. 3.2b shows the trellis diagram. The nodes in each column represent the set of encoder states $\mathcal{S}$ at a time instant. A solid/dashed edge from node S_i at a time t to S_j time $t+1$ means that encoder state transits from S_i to S_j when encoder has input $u_t = 1/u_t = 0$, respectively. As an example, we highlight the state transition for the information sequence $\mathbf{u} = [1, 0, 1, 1, 0, 0]$ with red arrows. ■

3.1.3 Decoding

There exist two optimal decoding algorithms for convolutional codes: *Viterbi algorithm* [46] and *BCJR algorithm* [47]. Viterbi algorithm performs sequential



(a) State-transition diagram.



(b) Trellis disgram.

Figure 3.2: Graphical representation for convolutional encoder $\mathbf{G} = [7\ 5]$.

ML decoding, which minimizes the block error probability, while BCJR algorithm performs bit-wise MAP decoding, which minimizes the bit error probability. Generally speaking, these two algorithms have similar error performance characteristics, in terms of either block error rate or bit error rate. In this subsection, we briefly review the BCJR algorithm, as it is commonly used in turbo decoders.

BCJR Algorithm

BCJR algorithm is used in turbo decoders to decode the constituent RSC codes, because it has soft-decision output, while Viterbi algorithm only has hard-decision output. Although soft-output Viterbi algorithm (SOVA) [71] also have soft-decision output, BCJR has slightly better performance than SOVA [72].

Here, we review details of the BCJR algorithm in the log domain, which is also called log-MAP algorithm [72]. The criterion of BCJR is to minimize the bit error probability by maximizing APP $\Pr(u_t|\mathbf{y})$. Let $\mathbf{y} = [\mathbf{y}_1, \ldots, \mathbf{y}_K]$ be the received sequence such that $\mathbf{y}_t$ is received when $\mathbf{v}_t$ is transmitted, $1 \leq t \leq K$. The soft-decision output $L(u_t)$, in terms of LLR, is given by

$$\begin{aligned} L(u_t) &= \ln \left[\frac{\Pr(u_t = 0|\mathbf{y})}{\Pr(u_t = 1|\mathbf{y})} \right] \\ &= \ln \left[\max_{(s',s)^0}{}^{*} \alpha_{t-1}(s')\gamma_t(s',s)\beta_{t+1}(s) \right] - \ln \left[\max_{(s',s)^1}{}^{*} \alpha_{t-1}(s')\gamma_t(s',s)\beta_{t+1}(s) \right]. \end{aligned} \tag{3.4}$$

The hard-decision output $\hat{u}_t$ is given by

$$\hat{u}_t = \begin{cases} 0, & \text{if } L(u_t) > 0, \\ 1, & \text{if } L(u_t) < 0. \end{cases} \tag{3.5}$$

In this expression, $(s', s)^0$ represents the set of state transitions when encoder input is 0, $(s', s)^1$ represents the set of state transitions when encoder input is 1, and (s', s) represents the state transition for an arbitrary input, respectively. The max-star function is [1]

$$\max{}^*(x, y) \triangleq \ln(e^x + e^y) = \max(x, y) + \ln\left(1 + e^{-|x-y|}\right). \tag{3.6}$$

The branch metric $\gamma_t(s', s) \triangleq \ln \Pr(s_t = s, \mathbf{y}_t | s_{t-1} = s')$, the forward metric $\alpha_t(s) \triangleq \ln \Pr(s_t = s, \mathbf{y}_1^t)$, and the backward metric $\beta_{t-1}(s') \triangleq \ln \Pr(\mathbf{y}_t^K | s_{t-1} = s')$

---

[1] There are various methods to approximate $\max^*(x, y)$, leading to simplified algorithms including max-log-MAP, constant-log-MAP, linear-log-MAP, etc. (see [73] and the references therein)

are computed as

$$\gamma_t(s', s) = -\ln p(y_t|u_t) \tag{3.7a}$$

$$\alpha_t(s) = \max_{s'}{}^* \left[\alpha_{t-1}(s') + \gamma(s', s)\right] \tag{3.7b}$$

$$\beta_{t-1}(s') = \max_{s}{}^* \left[\beta_t(s) + \gamma_t(s', s)\right] \tag{3.7c}$$

Consider that the encoder is initialized to the all-zero state and eventually terminates at the all-zero state, the initial condition of BCJR decoding is

$$\alpha_0(s) = \beta_K(s) = \begin{cases} 0, & \text{if } s = 0. \\ -\infty, & \text{if } s \neq 0. \end{cases} \tag{3.8}$$

3.2 Turbo Codes

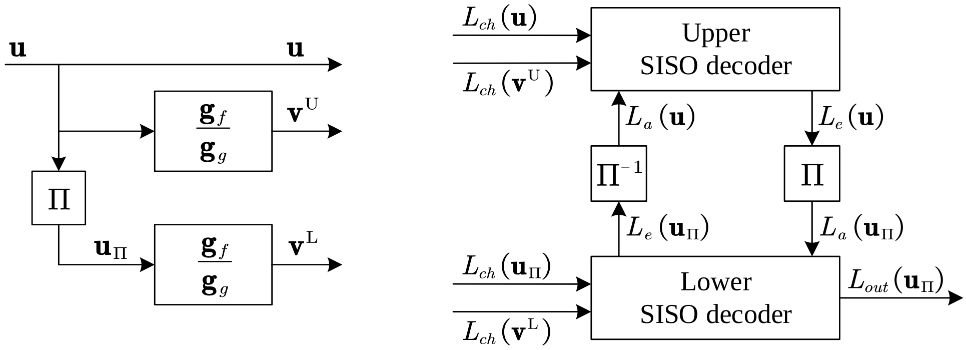
Turbo codes, also called parallel-concatenated convolutional codes (PCCs), consist of two or more identical RSC codes concatenated in parallel. In most cases, a widely used "turbo code" refers to a rate-$\frac{1}{3}$ turbo code consisting of two rate-$\frac{1}{2}$ RSC codes with generator matrix

$$\mathbf{G} = \begin{bmatrix} 1 & \frac{\mathbf{g}_f}{\mathbf{g}_b} \end{bmatrix}. \tag{3.9}$$

The encoder and decoder block diagrams of such a code are depicted in Fig. 3.3.

3.2.1 Encoding

As shown in Fig. 3.3a, the turbo encoder consists of two RSC encoders separated by an interleaver Π. The constituent RSC encoders are referred to as the upper/lower RSC encoder respectively. For a K-bit information sequence $\mathbf{u}$, the


u
u
$\frac{\mathbf{g}_f}{\mathbf{g}_g}$
$\mathbf{v}^{\mathrm{U}}$
Π
$\mathbf{u}_{\Pi}$
$\frac{\mathbf{g}_f}{\mathbf{g}_g}$
$\mathbf{v}^{\mathrm{L}}$
$L_{ch}(\mathbf{u})$
$L_{ch}(\mathbf{v}^{\mathrm{U}})$
Upper SISO decoder
$L_a(\mathbf{u})$
$L_e(\mathbf{u})$
Π^{-1}
Π
$L_e(\mathbf{u}_{\Pi})$
$L_a(\mathbf{u}_{\Pi})$
$L_{ch}(\mathbf{u}_{\Pi})$
$L_{ch}(\mathbf{v}^{\mathrm{L}})$
Lower SISO decoder
$L_{out}(\mathbf{u}_{\Pi})$

(a) Encoder block diagram. (b) Decoder block diagram

Figure 3.3: Turbo code block diagram.

encoder generates the codeword $\mathbf{x} = [\mathbf{u}, \mathbf{v}^{\mathrm{U}}, \mathbf{v}^{\mathrm{L}}]$, where $\mathbf{v}^{\mathrm{U}} = \mathbf{u} \cdot \frac{\mathbf{g}_f}{\mathbf{g}_b}$ is the parity sequence from the upper encoder, $\mathbf{v}^{\mathrm{L}} = \mathbf{u}_{\Pi} \cdot \frac{\mathbf{g}_f}{\mathbf{g}_b}$ is the parity sequence from the lower encoder, $\mathbf{u}_{\Pi} = \Pi(\mathbf{u})$, and $\|\mathbf{v}^{\mathrm{U}}\| = \|\mathbf{v}^{\mathrm{L}}\| = K$.

3.2.2 Decoding

As shown in Fig. 3.3b, the turbo decoder consists of two soft-in soft-out (SISO) decoders matched to the upper and lower RSC encoders, and a pair of interleaver Π and deinterleaver Π^{-1}. The constituent SISO decoders are typically realized by the BCJR algorithm [47] or the SOVA algorithm [71].

Recall that $L_{ch}(\cdot)$ represents channel LLR, $L_a(\cdot)$ represents the a-priori LLR, and $L_e(\cdot)$ represents extrinsic LLR. Let $L_{out}(\mathbf{u}) = \mathrm{DEC}(L_{in}(\mathbf{u}), L_{in}(\mathbf{v}))$ represents the underlying decoding algorithm for the constituent SISO decoder, where $L_{in}(\mathbf{u})$ and $L_{in}(\mathbf{v})$ are the input LLRs for information sequence $\mathbf{u}$ and parity sequence $\mathbf{v}$, and $L_{out}(\mathbf{u})$ is the soft-decision of $\mathbf{u}$.

Upon receiving a noisy codeword, the upper decoder receives $[L_{ch}(\mathbf{u}), L_{ch}(\mathbf{v}^{\mathrm{U}})]$, while the lower decoder receives $[L_{ch}(\mathbf{u}_{\Pi}), L_{ch}(\mathbf{v}^{\mathrm{L}})]$, where $L_{ch}(\mathbf{u}_{\Pi})$ is obtained via interleaving $L_{ch}(\mathbf{u})$. The a-priori LLRs $L_a(\mathbf{u})$ and $L_a(\mathbf{u}_{\Pi})$ are initialized as zero.

In each decoding iteration, the upper decoder performs

$$L_{in}(\mathbf{u}) = L_a(\mathbf{u}) + L_{ch}(\mathbf{u}), \tag{3.10a}$$

$$L_{out}(\mathbf{u}) = \mathrm{DEC}(L_{in}(\mathbf{u}), L_{in}(\mathbf{v}^{\mathrm{U}})), \tag{3.10b}$$

$$L_e(\mathbf{u}) = L_{out}(\mathbf{u}) - L_{in}(\mathbf{u}), \tag{3.10c}$$

and outputs the extrinsic LLR $L_e(\mathbf{u})$. Then, the a-priori LLR of the lower encoder is updated as $L_a(\mathbf{u}_\Pi) = \Pi(L_e(\mathbf{u}))$. The lower decoder performs

$$L_{in}(\mathbf{u}_\Pi) = L_a(\mathbf{u}_\Pi) + L_{ch}(\mathbf{u}_\Pi), \tag{3.11a}$$

$$L_{out}(\mathbf{u}_\Pi) = \mathrm{DEC}(L_{in}(\mathbf{u}_\Pi), L_{in}(\mathbf{v}^{\mathrm{L}})), \tag{3.11b}$$

$$L_e(\mathbf{u}_\Pi) = L_{out}(\mathbf{u}_\Pi) - L_{in}(\mathbf{u}_\Pi), \tag{3.11c}$$

and outputs the extrinsic LLR $L_e(\mathbf{u}_\Pi)$. The a-priori LLR of the upper encoder is updated as $L_a(\mathbf{u}) = \Pi^{-1}(L_e(\mathbf{u}_\Pi))$. The turbo decoding runs iteratively until it meets a predetermined exit condition or the maximum number of iterations is reached. After iteration terminates, the hard-decision estimation $\hat{\mathbf{u}}$ is produced base on the a-posteriori LLR

$$L_{out}(\mathbf{u}) = \Pi^{-1}(L_{out}(\mathbf{u}_\Pi)) = L_{ch}(\mathbf{u}) + L_e(\mathbf{u}) + \Pi^{-1}(L_e(\mathbf{u}_\Pi)). \tag{3.12}$$

3.2.3 Duo-Binary Turbo Codes

Duo-binary turbo codes (dTCs) refer to the turbo codes which encode two information sequences [74–76]. Compared with the conventional turbo codes, which encode one information sequence, dTCs have better convergence performance during decoding and better robustness to the simplified decoding algorithms such

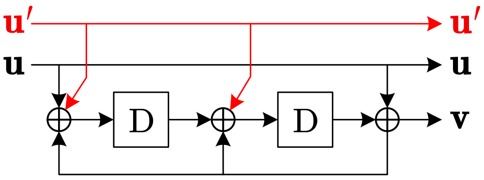


Figure 3.4: Encoder block diagram for $\mathbf{G}_{2/3} = \left[1\ 0\ \frac{5}{7};\ 0\ 1\ \frac{3}{7}\right]$.

as constant-log-Map algorithm and max-log-Map algorithm [75]. A rate-$\frac{1}{2}$ dTC is a parallel concatenation of two identical rate-$\frac{2}{3}$ RSC codes, whose generator matrix is

$$\mathbf{G}_{2/3} = \begin{bmatrix} 1 & 0 & \frac{\mathbf{g}_f}{\mathbf{g}_b} \\ 0 & 1 & \frac{\mathbf{g}'_f}{\mathbf{g}_b} \end{bmatrix}, \tag{3.13}$$

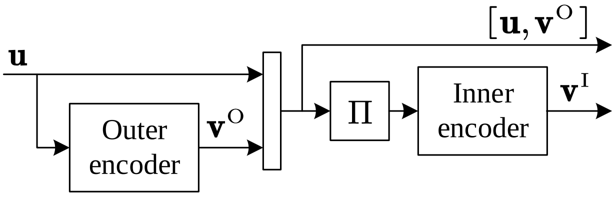
where $\mathbf{g}_f$ and $\mathbf{g}'_f$ are the forward generator polynomials for two information sequences $\mathbf{u}$ and $\mathbf{u}'$, respectively, and $\mathbf{g}_b$ is the feedback generator polynomial. The codeword of the rate-$\frac{2}{3}$ RSC code is given by $[\mathbf{u}\ \mathbf{u}'] \cdot \mathbf{G}_{2/3} = [\mathbf{u}\ \mathbf{u}'\ \mathbf{v}]$, where $\mathbf{v} = \frac{\mathbf{u}\cdot\mathbf{g}_f}{\mathbf{g}_b} + \frac{\mathbf{u}'\cdot\mathbf{g}'_f}{\mathbf{g}_b}$ is the parity sequence. As an example, Fig. 3.4 depicts the encoder block diagram of a rate-$\frac{2}{3}$ RSC code with $\mathbf{G}_{2/3} = \left[1\ 0\ \frac{5}{7};\ 0\ 1\ \frac{3}{7}\right]$.

3.2.4 Other Turbo-Like Codes

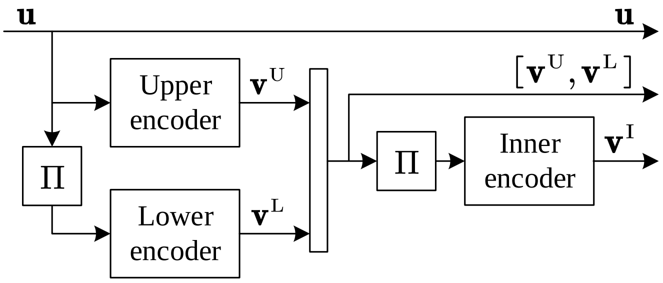
Apart from turbo codes, there are various forms of concatenated convolutional codes, referred to as turbo-like codes. In the following, we have a brief review of two typical classes of turbo-like codes: serially concatenated convolutional codes (SCCs) [77] and hybrid concatenated convolutional codes (HCCs) [78].

Serially Concatenated Convolutional Codes

SCCs are constructed by two component RSC codes placed in the cascade of each other. Fig. 3.5a shows the encoder structure of a rate-$\frac{1}{4}$ SCC constituting


u
Outer encoder
$\mathbf{v}^{\mathrm{O}}$
$[\mathbf{u},\mathbf{v}^{\mathrm{O}}]$
Π
Inner encoder
$\mathbf{v}^{\mathrm{I}}$

(a) Encoder block diagram of SCC.


u
u
Upper encoder
$\mathbf{v}^{\mathrm{U}}$
Π
Lower encoder
$\mathbf{v}^{\mathrm{L}}$
$[\mathbf{v}^{\mathrm{U}},\mathbf{v}^{\mathrm{L}}]$
Π
Inner encoder
$\mathbf{v}^{\mathrm{I}}$

(b) Encoder block diagram of HCC.

Figure 3.5: Example of turbo-like codes.

two rate-$\frac{1}{2}$ RSC codes. First, the outer encoder firstly encodes the information sequence $\mathbf{u}$ of length K, and generates the outer parity sequence $\mathbf{v}^{\mathrm{O}}$ of length K. Then, the interleaver permutes $[\mathbf{u},\mathbf{v}^{\mathrm{O}}]$ received from the outer encoder and passes the interleaved sequence to the inner encoder. Lastly, inner encoder generate the inner parity sequence $\mathbf{v}^{\mathrm{I}}$ of length $2K$, and the SCC codeword is $\mathbf{x} = [\mathbf{u},\mathbf{v}^{\mathrm{O}},\mathbf{v}^{\mathrm{I}}]$.

Hybrid Concatenated Convolutional Codes

HCCs are constructed by concatenating three RSC codes. Fig. 3.5b shows the encoder structure of a rate-$\frac{1}{5}$ HCC. The information sequence $\mathbf{u}$ of length K is firstly encoded by two parallel concatenated RSC encoders, giving the parity sequences $\mathbf{v}^{\mathrm{U}}$ and $\mathbf{v}^{\mathrm{L}}$ respectively, and $\|\mathbf{v}^{\mathrm{U}}\| = \|\mathbf{v}^{\mathrm{L}}\| = K$. Later, the inner encoder inputs the interleaved sequence $[\mathbf{v}^{\mathrm{U}},\mathbf{v}^{\mathrm{L}}]_{\Pi}$ and produce the parity sequence $\mathbf{v}^{\mathrm{I}}$ of length $2K$, and the HCC codeword is $[\mathbf{u},\mathbf{v}^{\mathrm{U}},\mathbf{v}^{\mathrm{L}},\mathbf{v}^{\mathrm{I}}]$.

3.3 Tools for Performance Analysis

The error performance of a turbo code (or a turbo-like code) has two different behaviors in two different regions. One is the waterfall region in the low SNR, where the error rate drops sharply when SNR increases. The other is the error floor region in the high SNR, where the error rate reduces very slowly. In this section, we first review the density evolution (DE) analysis, which describes the performance in the waterfall region. We then review the distance property analysis, which describes the performance in the error floor region.

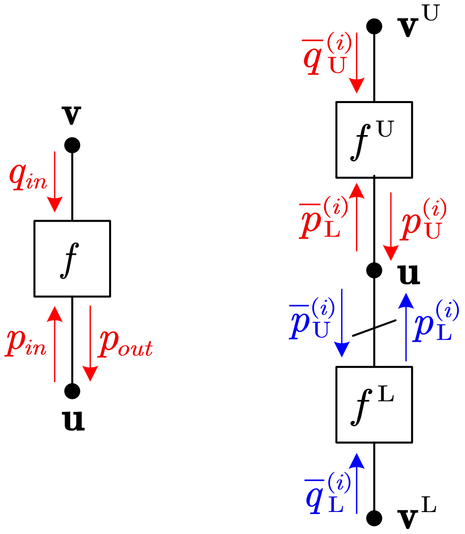
3.3.1 Density Evolution Analysis

In this section, we review the DE analysis of turbo-like codes over the BEC[2]. DE is a technique to describe the performance of iterative decoders when the codeword length is infinite. For the BEC, DE describes how the erasure probability of decoding messages evolves during the iterative decoding process. Due to the simplicity of BECs, we can express DE explicitly by a set of equations.

To explain how DE describes the iterative decoding performance, we review the DE analysis for turbo codes in this section. We first review the compact graph of turbo codes. Then, we show how to derive the exact DE equations and obtain the iterative decoding threshold, also called belief propagation (BP) decoding threshold using the DE analysis.

---

[2] Unlike BECs where the input-output relations of an RSC decoder can be expressed by a set of exact transfer functions, in general binary-input memoryless channels, the transfer functions cannot be expressed in closed form. Although the Monte-Carlo method [79] and the Gaussian approximation method [80] can approximate decoding thresholds over the AWGN channel, they require large computational efforts and are very time-consuming. Alternatively, it was shown in [79, 81] that the decoding thresholds over the AWGN channel can be predicted from the decoding thresholds over the BEC, and the predicted thresholds are very close to the Monte-Carlo thresholds.


$\mathbf{v}$
q_{in}
f
p_{in}
p_{out}
$\mathbf{u}$
$\mathbf{v}^{\mathrm{U}}$
$\overline{q}_{\mathrm{U}}^{(i)}$
f^{U}
$\overline{p}_{\mathrm{L}}^{(i)}$
$p_{\mathrm{U}}^{(i)}$
$\mathbf{u}$
$\overline{p}_{\mathrm{U}}^{(i)}$
$p_{\mathrm{L}}^{(i)}$
f^{L}
$\overline{q}_{\mathrm{L}}^{(i)}$
$\mathbf{v}^{\mathrm{L}}$

(a) (b)

Figure 3.6: Compact graph of a) a rate-$\frac{1}{2}$ RSC code, and b) a rate-$\frac{1}{3}$ turbo code.

Compact Graph

In the compact graph [38] representation of a convolutional code, each information sequence is treated as one variable node (VN), and each parity sequence is treated as one VN. The SISO decoder of an RSC code is treated as one factor node. Fig. 3.6a shows the compact graph of a rate $\frac{1}{2}$ RSC code, where the decoder is represented by $\boxed{f}$.

Fig. 3.6b shows the compact graph of a rate-$\frac{1}{3}$ turbo code. In this graph, we use three variable nodes (represented by filled circles) to represent the information sequence $\mathbf{u}$, the upper parity sequence $\mathbf{v}^{\mathrm{U}}$, and the lower parity sequence $\mathbf{v}^{\mathrm{L}}$. The upper and the lower decoders are represented by $\boxed{f^{\mathrm{U}}}$ and $\boxed{f^{\mathrm{L}}}$. The interleaver is represented by a slash on the edge between $\mathbf{u}$ and $\boxed{f^{\mathrm{L}}}$.

Density Evolution Functions

We first consider the RSC decoder shown in Fig. 3.6a. The DE of the decoding process $L_e(\mathbf{u}) = \mathrm{DEC}(L_{in}(\mathbf{u}), L_{in}(\mathbf{v})) - L_{in}(\mathbf{u})$ can be expressed by the transfer function [82]

$$p_{out} = F(p_{in}, q_{in}), \tag{3.14}$$

where p_{in} and q_{in} are the erasure probability of the decoder inputs $L_{in}(\mathbf{u})$ and $L_{in}(\mathbf{v})$, while p_{out} is the erasure probability of the decoder output $L_e(\mathbf{u})$.

Now consider the turbo decoder shown in Fig. 3.6b. For simplicity, we consider that the upper and lower decoder have the same transfer function. The notations for the DE functions at the i-th decoding iteration is as follows:

Notation	Definition
$\bar{p}_{\mathrm{L}}^{(i)}$	Information input erasure probability of f^{U}
$\bar{q}_{\mathrm{U}}^{(i)}$	Parity input erasure probability of f^{U}
$p_{\mathrm{U}}^{(i)}$	Information output erasure probability of f^{U}
$\bar{p}_{\mathrm{U}}^{(i)}$	Information input erasure probability of f^{L}
$\bar{q}_{\mathrm{L}}^{(i)}$	Parity input erasure probability of f^{L}
$p_{\mathrm{L}}^{(i)}$	Information output erasure probability of f^{L}

At upper decoder and the i-th decoding iteration, the input of the upper decoder information sequence $L_{in}(\mathbf{u})$ denpends on both the channel LLR and the extrinsic information from the lower decoder, so that its erasure probability is $\bar{p}_{\mathrm{L}}^{(i)} = \varepsilon \cdot p_{\mathrm{L}}^{(i-1)}$. The input of the upper decoder parity sequence $L_{in}(\mathbf{v}^{\mathrm{U}})$ only depends on the channel LLR, so that its erasure probability is $\bar{q}_{\mathrm{U}}^{(i)} = \varepsilon$. Therefore, the DE function of the upper encoder is

$$p_{\mathrm{U}}^{(i)} = F(\bar{p}_{\mathrm{L}}^{(i)}, \bar{q}_{\mathrm{U}}^{(i)}) = F(\varepsilon \cdot p_{\mathrm{L}}^{(i-1)}, \varepsilon). \tag{3.15}$$

Likewise, the DE function of the lower encoder is

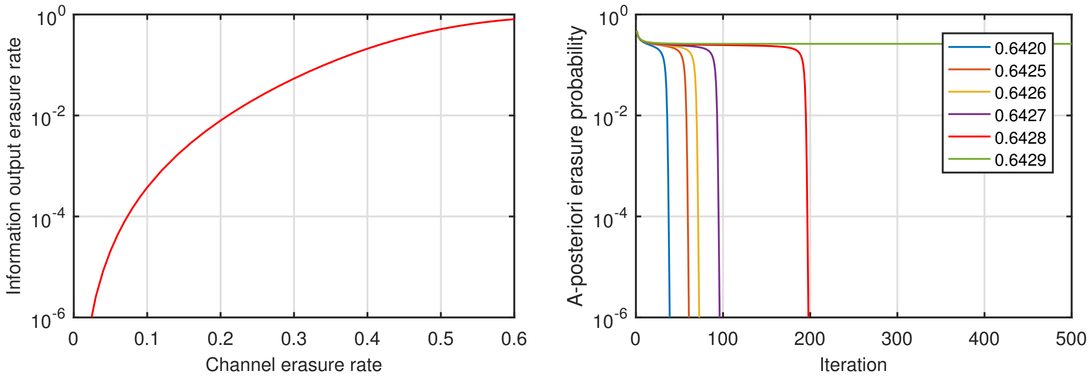
$$p_{\mathrm{L}}^{(i)} = F(\bar{p}_{\mathrm{U}}^{(i)}, \bar{q}_{\mathrm{L}}^{(i)}) = F(\varepsilon \cdot p_{\mathrm{U}}^{(i-1)}, \varepsilon). \tag{3.16}$$

The a-posteriori LLR depends on both the upper and the lower extrinsic LLR, as well as the channel LLR. Thus, the a-posteriori erasure probability is

$$p_{\mathbf{u}}^{(i)} = \varepsilon \cdot p_{\mathrm{U}}^{(i)} \cdot p_{\mathrm{L}}^{(i)}. \tag{3.17}$$

For a turbo-like code and at a given BEC erasure rate, if the a-posteriori erasure probability drops to zero as the number of iterations increases, it means that any codeword of infinite length received over the BEC can be successfully decoded when BP decoding is used. Otherwise, if the a-posteriori erasure probability becomes a non-zero constant as the number of iterations increases, it means that the decoding of any received codeword would fail. The maximum BEC erasure rate that the a-posteriori erasure probability drops to zero is referred to as the BP decoding threshold. General speaking, a large BP decoding threshold leads to a good finite-length error performance in the waterfall regime.

Example 3.3. Consider the turbo code with component RSC generator $\mathbf{G} = [1, \frac{5}{7}]$. Fig. 3.7a plots the transfer function of the component RSC code, while 3.7b plots the evolution of a-posteriori erasure probability over decoding iterations for various BEC erasure rates. When $\varepsilon = 0.6428$, the a-posteriori erasure probability drops to zero for around 200 iterations. When $\varepsilon = 0.6429$, the a-posteriori erasure probability stays at 0.2622. Thus, the BP decoding threshold of the turbo code is $\varepsilon_{\mathrm{BP}} = 0.6428$. ■



(a) Transfer function of the component RSC code

(b) Density evolution of turbo code over decoding iterations

Figure 3.7: Density evolution for the turbo code with $\mathbf{G} = [1, \frac{5}{7}]$.

3.3.2 Distance Properties

Consider a convolutional code with codebook $\mathcal{C}$. Let $\mathbf{c} \in \mathcal{C}$, $w(\mathbf{c})$ be the Hamming weight (the number of 1s) of $\mathbf{c}$. Given an information sequence $\mathbf{u}$ producing codeword $\mathbf{c}$, $w(\mathbf{u})$ is referred to as the information weight of $\mathbf{c}$.

The performance of a convolutional code can be described by its distance spectrum. Let d be the codeword weight, A_d be the multiplicity (the total number) of weight-d codewords, and $\bar{w}_d$ be the average information weight causing the weight-d codewords. The bit error rate (BER) performance of a convolutional code with maximum-likelihood decoding over an AWGN channel is upper-bounded by the union bound [83–85]

$$P_b \leq \sum_{d=d_{\text{free}}}^{N} \frac{A_d \bar{w}_d}{K} Q\left(\sqrt{\frac{2dE_s}{N_0}}\right), \tag{3.18}$$

where $d_{\text{free}} \triangleq \min\limits_{w(\mathbf{c}) \in \mathcal{C}, \mathbf{c} \neq \mathbf{0}} w(\mathbf{c})$ is the minimum weight of all non-zero codewords, called the free distance. The Q-function is defined as $Q(x) \triangleq \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{-z^2/2}\, dz$. For a high SNR, the free-distance term dominates the above union bound.

Therefore, the asymptotic performance of a convolutional code approaches

$$P_b \approx \frac{A_{\text{free}} \bar{w}_{\text{free}}}{K} Q\left(\sqrt{\frac{2d_{\text{free}} E_s}{N_0}}\right), \tag{3.19}$$

where A_{free} is the free distance multiplicity, i.e., the number of free-distance codewords whose weight equals d_{free}, and $\bar{w}_{\text{free}}$ is the average information weight causing the free-distance codewords.

Theoretically, the error performance of a turbo code (or a turbo-like code) with ML decoding can also be bounded using (3.18). In addition, [86] have derived upper bounds on the BLER and BER of turbo codes with ML decoding. Due to the interleaver between component encoders, an exhaustive search is required to obtain the exact distance spectrum. Thus, it is extremely difficult to compute the BER upper bound using (3.18) for a large information length due to the large search range. Various works [83,84,87,88] have studied the methods to search for the free distance term, and it was shown that the BER in the error floor regime can be lower bounded by using the free distance asymptote (3.19).

In practice, the BER in the error floor regime is largely determined by the low-weight codewords given by weight-2 information sequences, because the interleaving makes the number of low-weight codewords given by information sequences with a weight higher than two negligibly small [89]. Thus, the minimum codeword weight given by weight-2 information sequences is commonly referred to as the free distance (or effective minimum distance) of the turbo code.

Example 3.4. Consider the turbo code with $\mathbf{G} = [1, \frac{5}{7}]$, $K = 10^3$, and pseudo-random interleavers. Although the exact value of free distance terms may change with the interleaver, for most of the interleavers, it is observed that the free distance is 10, and the averaged multiplicity A_{10} is around 2. Fig. 3.8 shows the BER curve of this turbo code and its free distance asymptote with free distance

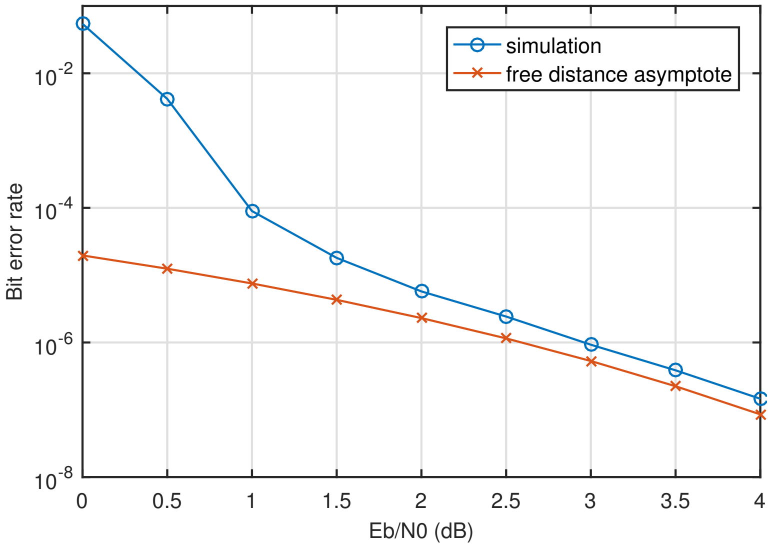


Figure 3.8: Free distance asymptote of the turbo code with $\mathbf{G} = [1, \frac{5}{7}]$ and $K = 10^3$.

terms $d_{\text{free}} = 10$, $A_{\text{free}} = 2$, and $\bar{w}_{\text{free}} = 2$. ■

3.4 Spatial Coupling Design for Turbo-Like Codes

There exist many works on applying spatial coupling technique to turbo-like codes. In [63], the authors proposed laminated turbo codes (LTCs), whose component code is a rate-k/n RSC code. The coupling is performed such that the RSC encoder takes the information bits at the current time instant and at the previous $k-1$ time instants, so that each information sequence is encoded by k times. The authors showed that LTCs outperform the turbo codes of the same rates, and meanwhile provide high rate flexibility and good convergence behavior for small codeword lengths. In [64], the authors constructed braided convolutional codes (BCCs), which can be seen as a sliding version of product codes [90] with an RSC code as the component code. BCCs outperform turbo codes in both the waterfall and the error floor regimes. It was also shown that the free distance of

BCCs grows linearly with constraint length. In [38], the authors extended BCCs to large coupling memories and applied spatial coupling on parallel concatenated RSC codes (SC-PCCs) and serially concatenated RSC codes (SC-SCC). In their paper, it was proved that threshold saturation occurs for their proposed codes.

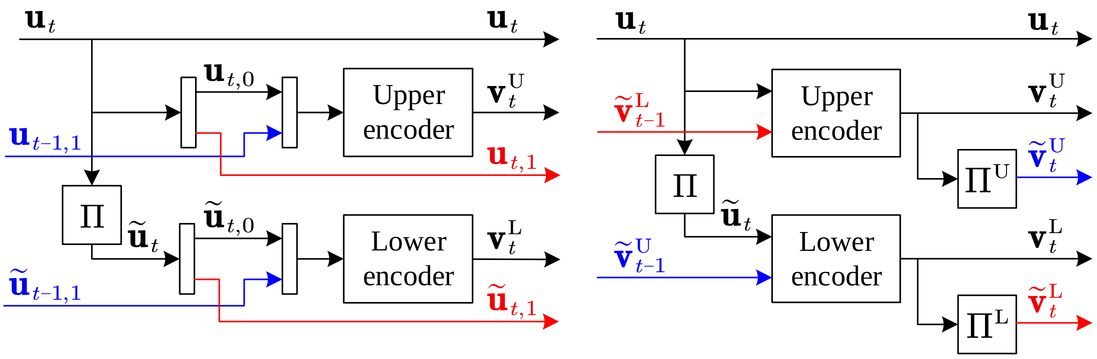
All the above works and their subsequent studies on spatially coupled turbo-like codes [65,79,81,91,92] have reported that spatial coupling brings significantly performance improvement on turbo-like codes in terms of decoding threshold and error floor.

In the following, we review some typical spatially coupled turbo-like codes, which will be used as the performance benchmark in Chapters 4 – 6.

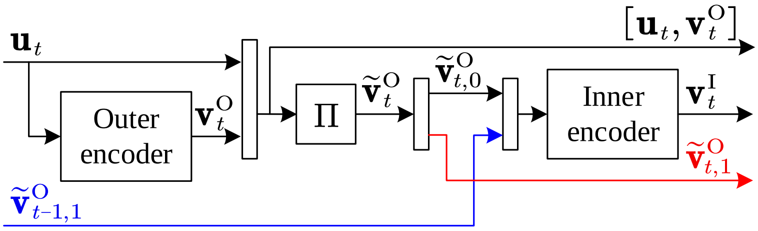
3.4.1 SC-PCCs and SC-SCCs

In [38], the authors introduced two types of spatially coupled turbo-like codes: spatially coupled parallel concatenated codes (SC-PCCs) and spatially coupled serially concatenated codes (SC-SCCs). For both codes, the coupling is done in a block-wise manner. That is, given a set of L turbo code/SCC encoders, each encoder inputs the subsequences of the outputs of its preceding encoders. In the following, we briefly review the encoding of SC-PCCs and SC-PCCs with coupling memory $m = 1$, i.e., the t-th encoder only takes a subsequence of the output of the $t - 1$-th encoder.

Fig. 3.9a plots the encoder block diagram of a SC-PCC with coupling memory $m = 1$. At time t, the t-th turbo encoder takes information sequence $\mathbf{u}_t$. At the upper encoder, $\mathbf{u}_t$ is de-multiplexed, giving two subsequences $\mathbf{u}_{t,0}$ and $\mathbf{u}_{t,1}$. The upper encoder takes $[\mathbf{u}_{t,0}, \mathbf{u}_{t-1,1}]$ as input and produces the upper parity sequence $\mathbf{v}_t^{\mathrm{U}}$. Meanwhile, at the lower encoder, $\widetilde{\mathbf{u}}_t$ is de-multiplexed, giving two subsequences $\widetilde{\mathbf{u}}_{t,0}$ and $\widetilde{\mathbf{u}}_{t,1}$. The lower encoder takes $[\widetilde{\mathbf{u}}_{t,0}, \widetilde{\mathbf{u}}_{t-1,1}]$ as input and


$\mathbf{u}_t$
$\mathbf{u}_t$
$\mathbf{u}_{t,0}$
Upper encoder
$\mathbf{v}_t^{\mathrm{U}}$
$\mathbf{u}_{t\text{-}1,1}$
$\mathbf{u}_{t,1}$
Π
$\widetilde{\mathbf{u}}_t$
$\widetilde{\mathbf{u}}_{t,0}$
Lower encoder
$\mathbf{v}_t^{\mathrm{L}}$
$\widetilde{\mathbf{u}}_{t\text{-}1,1}$
$\widetilde{\mathbf{u}}_{t,1}$
$\mathbf{u}_t$
$\mathbf{u}_t$
$\widetilde{\mathbf{v}}_{t\text{-}1}^{\mathrm{L}}$
Upper encoder
$\mathbf{v}_t^{\mathrm{U}}$
Π
Π^{U}
$\widetilde{\mathbf{v}}_t^{\mathrm{U}}$
$\widetilde{\mathbf{u}}_t$
$\widetilde{\mathbf{v}}_{t\text{-}1}^{\mathrm{U}}$
Lower encoder
$\mathbf{v}_t^{\mathrm{L}}$
Π^{L}
$\widetilde{\mathbf{v}}_t^{\mathrm{L}}$

(a) SC-PCC (b) BCC


$\mathbf{u}_t$
$[\mathbf{u}_t, \mathbf{v}_t^{\mathrm{O}}]$
Outer encoder
$\mathbf{v}_t^{\mathrm{O}}$
Π
$\widetilde{\mathbf{v}}_t^{\mathrm{O}}$
$\widetilde{\mathbf{v}}_{t,0}^{\mathrm{O}}$
Inner encoder
$\mathbf{v}_t^{\mathrm{I}}$
$\widetilde{\mathbf{v}}_{t,1}^{\mathrm{O}}$
$\widetilde{\mathbf{v}}_{t\text{-}1,1}^{\mathrm{O}}$

(c) SC-SCC

Figure 3.9: Encoder block diagram of spatially coupled turbo-like codes.

produces the lower parity sequence $\mathbf{v}_t^{\mathrm{L}}$. The output of the t-th turbo encoder is given by $[\mathbf{u}_t, \mathbf{v}_t^{\mathrm{U}}, \mathbf{v}_t^{\mathrm{L}}]$.

Fig. 3.9c plots the encoder block diagram of an SC-PCC with $m = 1$. At the t-th SCC encoder, the interleaved output of the outer encoder $\widetilde{\mathbf{v}}_t^{\mathrm{O}}$, consisting of the information sequence $\mathbf{u}_t$ and the outer parity sequence $\mathbf{v}_t^{\mathrm{O}}$, is de-multiplexed, giving the subsequences $\widetilde{\mathbf{v}}_{t,0}^{\mathrm{O}}$ and $\widetilde{\mathbf{v}}_{t,1}^{\mathrm{O}}$. The inner coder inputs $[\widetilde{\mathbf{v}}_{t,0}^{\mathrm{O}}, \widetilde{\mathbf{v}}_{t-1,1}^{\mathrm{O}}]$ and produce the inner parity sequence $\mathbf{v}_t^{\mathrm{I}}$. The output of the t-th SCC encoder is given by $[\mathbf{u}_t, \mathbf{v}_t^{\mathrm{O}}, \mathbf{v}_t^{\mathrm{I}}]$.

3.4.2 BCCs

There are two types of braided convolutional codes (BCCs). Unlike typical SC-PCCs and SC-SCCs which are built by concatenating rate-$\frac{1}{2}$ RSC codes, BCCs are built by concatenating duo-binary turbo codes, i.e., parallel concatenated rate-$\frac{2}{3}$ RSC codes. The original design of BCCs was firstly introduced in [64]. In this design, the authors only consider coupling memory $m = 1$. Later, in [38], the authors extend the structure of the original BCCs to $m \geq 1$, referred to as the type-I BCCs.

Fig. 3.9b shows the encoder block diagram of a type-I BCC with $m = 1$. In the t-th duo-binary turbo encoder, the upper encoder takes two input sequences. The first input sequence is the information sequence $\mathbf{u}_t$. The second input sequence is the interleaved parity sequence $\widetilde{\mathbf{v}}^{\mathrm{U}}_{t-1}$ from the lower encoder of the $t-1$-th duo-binary turbo encoder. Likewise, the lower encoder takes two input sequences: the interleaved information sequence $\widetilde{\mathbf{u}}_t$, and the interleaved parity sequence $\widetilde{\mathbf{v}}^{\mathrm{U}}_{t-1}$ from the upper encoder of the $t-1$-th duo-binary turbo encoder. The output of the t-th duo-binary turbo encoder is $[\mathbf{u}_t, \mathbf{v}^{\mathrm{U}}_t, \mathbf{v}^{\mathrm{L}}_t]$.

In addition, [38] introduced another type of BCCs, referred to as type-II BCCs. On the basis of the coupling structure of type-I BCCs, type-II BCCs also employ the coupling structure of SC-PCCs on the first input sequence of the duo-binary turbo encoder. That is, $\mathbf{u}_t$ is split into $\mathbf{u}_{t,0}$ and $\mathbf{u}_{t,1}$, while $\widetilde{\mathbf{u}}_t$ is split into $\widetilde{\mathbf{u}}_{t,0}$ and $\widetilde{\mathbf{u}}_{t,1}$. The first input sequence of the upper and the lower encoder are $[\mathbf{u}_{t,0}, \mathbf{u}_{t-1,1}]$ and $[\widetilde{\mathbf{u}}_{t,0}, \widetilde{\mathbf{u}}_{t-1,1}]$ respectively.

3.5 Summary

In this chapter, we have reviewed the background on turbo codes, including the encoding and decoding algorithm, performance analysis, and existing spatially coupled turbo-like codes.

In Chapters 4 – 6, we will present our spatially coupled turbo-like codes design. In Chapter 4, we will introduce partially information coupled turbo codes (PIC-TCs) and partially parity coupled turbo codes (PPC-TCs). In Chapter 5, we will introduce various partially coupled codes by applying coupling to duo-binary turbo codes. In particular, a generalization of type-I BCCs is also included. In Chapter 6, we will introduce generalized SC-PCCs (GSC-PCCs) and show that PIC-TCs and SC-PCCs are both special cases of GSC-PCCs.

Chapter 4

Partially Coupled Turbo Codes

In this chapter, we introduce the partial coupling technique to improve the transmission efficiency of the transport block based communication standards. We propose two types of partially coupled turbo codes: partially information coupled turbo codes and partially parity coupled turbo codes. The coupling is performed such that a fraction of the information/parity sequence of a turbo code block (CB) is used as a part of the information sequence of the succeeding CBs. The asymptotic decoding thresholds of the proposed codes over binary erasure channels (BECs) are analyzed via density evolution. Simulation results show the significant error performance improvement of the proposed codes over the benchmark turbo code and existing designs of spatially coupled turbo codes.

4.1 Introduction

Spatial coupling is a code construction technique that connects a sequence of component codes to form a long codeword chain. It was firstly introduced in [35] for spatially coupled LDPC (SC-LDPC) codes. Spatially coupled codes can be decoded by a sliding window decoder [36] with lower decoding latency compared to decoding the whole codeword block. Thus, spatially coupled codes become very popular in a wide range of applications in communications.

In this chapter, we introduce a new spatial coupling technique called *partial coupling*, including partially information coupling (PIC) and partially parity coupling (PPC). By applying partial coupling to turbo codes, we construct two types of partially coupled turbo codes: partially information coupled turbo codes (PIC-TCs) and partially parity coupled turbo codes (PPC-TCs) [93, 94]. The main contributions[1] of this chapter are as follows:

- We employ a rate-$\frac{1}{3}$ turbo code as the component code to construct PIC-TCs and PPC-TCs. To construct PIC-TCs, a fraction of the information sequence of a turbo encoder at one time instance is used as a part of the information sequence of the turbo code encoder at the succeeding time instances. To construct PPC-TCs, a fraction of the parity sequence of a turbo encoder at one time instance is used as the information sequence of the turbo code encoder at the succeeding time instances.

  The decoding scheme of PIC-TCs and PPC-TCs are presented, respectively. Each turbo CB is decoded with the BCJR [47] algorithm as partial coupling preserves the code structure of the underlying turbo code. Then, the reliable decoding messages of each CB are passed to other CBs via the coupled sequences.

- The (belief propagation) BP decoding thresholds of the proposed codes on the binary erasure channel (BEC) are investigated. We first look into the graph models of the proposed code ensembles. Based on the

---

[1] I was partially involved in the works presented in this chapter. To be specific, I partially contributed to the derivation of density evolution functions and the numerical computation of BP decoding thresholds. In addition, I also contributed to the implementation of simulation programs.

graph representations, we respectively derive their exact DE equations and compute their BP decoding thresholds for different coupling ratios λ and coupling memories m. We also investigate the rate-compatible design of our codes by considering random puncturing. The coupling ratios of our codes are optimized to achieve the largest decoding thresholds for different code rates and coupling memories.

We show that both PIC-TCs and PPC-TCs achieve larger decoding thresholds than turbo codes and SC-PCCs [38] of the same rates. Most notably, PPC-TCs only show a gap to the BEC capacity within 0.0002 for a wide range of code rates for large enough coupling memory.

- We evaluate the error performance of our codes over the BEC via simulation. Simulation results confirm our theoretical analysis and show significant error performance improvement over existing spatially coupled turbo-like codes.

4.2 PIC-Turbo Codes

In this section, we introduce PIC-TCs. We first present the encoding process of PIC-TCs. Then, the decoding scheme is presented.

4.2.1 Encoding of PIC-TCs

Fig. 4.1 depicts the encoder block diagram of the PIC-polar code with $m = 1$. The underlying component code of each CB encoder is a rate-$\frac{1}{3}$ turbo code with information length K. The encoding processes are as follows.

---

1. **Information segmentation:** Let $\mathbf{u}$ be the information sequence of a TB, L be the number of CBs, and CB_t represent the t-th CB. The sequence $\mathbf{u}$

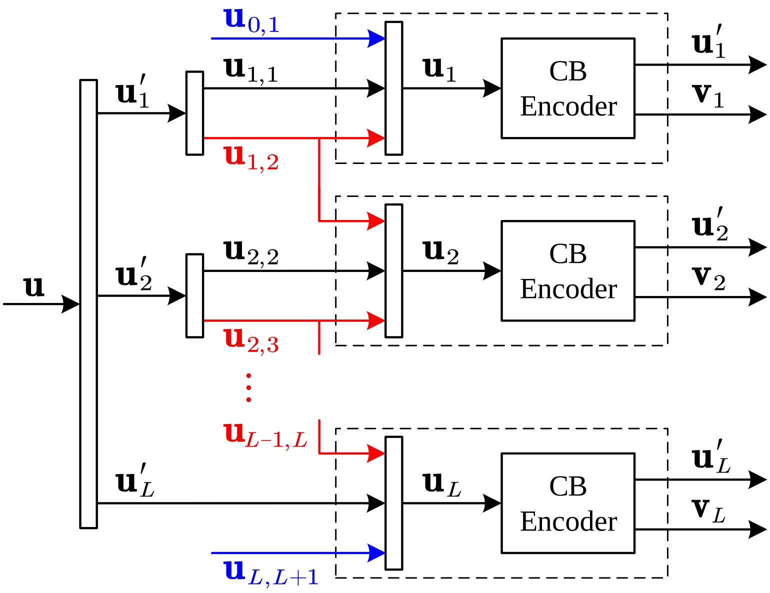


Figure 4.1: Encoding scheme of PIC-TCs.

is segmented into L subsequences $\mathbf{u}'_1, \dots, \mathbf{u}'_L$. For $t = 1, \dots, L-1$, $\mathbf{u}'_t$ is further segmented into two subsequences $\mathbf{u}_{t,t}$ and $\mathbf{u}_{t,t+1}$, where $\mathbf{u}_{t,t}$ is the uncoupled information sequence encoded by CB_t, and $\mathbf{u}_{t,t+1}$ is the coupled information sequence encoded by both CB_t and CB_{t+1}. At $t = L$, $\mathbf{u}'_t$ only consists of the uncoupled information sequence so that $\mathbf{u}_{L,L} = \mathbf{u}'_L$.

2. **Padding bits insertion:** To initialize and terminate the encoding, two all-zero blocks $\mathbf{u}_{0,1}$ and $\mathbf{u}_{L,L+1}$ are attached before $\mathbf{u}_{1,1}$ and after $\mathbf{u}_{L,L}$ respectively.

3. **Information coupling and CB encoding:** For $t = 1, \dots, L$, CB_t takes $\mathbf{u}_t = [\mathbf{u}_{t-1,t}, \mathbf{u}_{t,t}, \mathbf{u}_{t,t+1}]$ as its encoder input, and produce the parity sequence $\mathbf{v}_t = [\mathbf{v}_t^{\text{U}}, \mathbf{v}_t^{\text{L}}]$, where $\mathbf{v}_t^{\text{U}}$ and $\mathbf{v}_t^{\text{L}}$ are the parity subsequences produced by the upper and lower RSC encoder of the component turbo code, respectively. The final codeword of CB_t is obtained as $\mathbf{x}_t = [\mathbf{u}'_t, \mathbf{v}_t]$. That is, the coupled sequence $\mathbf{u}_{t-1,t}$ is not included in $\mathbf{x}_t$ since it has already been included as a part of $\mathbf{x}_{t-1}$. This also means that each coupled sequence is

encoded twice, but it is only transmitted once.

Remark 4.1. In practice, $\mathbf{u}_t$ needs to be processed with a CRC encoder before entering the CB encoder so that receiver can detect whether errors appear in CB_t. For simplicity, we omit the CRC bits because they do not affect the decoding performance of PIC-TCs and are only used for error detection. ◀

Recall that K is the information length of the CB encoder, that is $\|\mathbf{u}_t\| = K$. With the rate-$\frac{1}{3}$ component turbo code, the length of a parity sequence is $\|\mathbf{v}_t\| = 2K$. Let K_c be the length of a coupled sequence (or a padding sequence), then the length of an uncoupled sequence is $\|\mathbf{u}_{t,t}\| = K_u = K - 2K_c$. Note that $\|\mathbf{u}'_t\| = K - K_c$ for $t = 1, \ldots, L-1$ while $\|\mathbf{u}'_t\| = K - 2K_c$ for $t = L$ because $\mathbf{u}'_L$ only consists of an uncoupled sequence. The code rate of the PIC-TCs with $m = 1$ is

$$R = \frac{\sum_{t=1}^{L} \|\mathbf{u}'_t\|}{\sum_{t=1}^{L} \|\mathbf{u}'_t\| + \|\mathbf{v}_t\|} = \frac{LK - (L+1)K_c}{3LK - (L+1)K_c}, \tag{4.1}$$

We now consider the encoding of the PIC-TCs with coupling memory $m \geq 1$. The information sequence $\mathbf{u}$ is segmented into subsequences $\mathbf{u}'_1, \ldots, \mathbf{u}'_L$, where $\|\mathbf{u}_t\| = K - K_c$ for $t = 1, \ldots, L-m$, and $\|\mathbf{u}_t\| = K - \frac{L+2m-t}{m} K_c$ for $t = L - t + 1, \ldots, L$. At time t, $\mathbf{u}'_t$ is segmented into the uncoupled sequence $\mathbf{u}_{t,t}$ and the coupled subsequences $\mathbf{u}_{t,t+j}$, $1 \leq j \leq \min(m, L-t)$. The CB_t encoder takes $\mathbf{u}_t = [\mathbf{u}_{t-m,t}, \ldots, \mathbf{u}_{t-m,t}, \mathbf{u}_{t,t}, \mathbf{u}_{t,t+1}, \ldots, \mathbf{u}_{t,t+m}]$ as input and produces the parity sequence $\mathbf{v}_t$. The codeword of CB_t is obtained as $\mathbf{x}_t = [\mathbf{u}'_t, \mathbf{v}_t]$. Note that $\mathbf{u}_{t-j,t}$ such that $t - j < 1$ and $\mathbf{u}_{t,t+j}$ such that $t + j > L$ are the all-zero padding sequences, and are not included in the codewords $\mathbf{x}_1, \ldots, \mathbf{x}_L$. The length of a coupled (or padding) sequence is $\|\mathbf{u}_{t,t+j}\| = \frac{K_c}{m}$, $1 \leq j \leq m$. We define the coupling ratio as

$$\lambda \triangleq \frac{\sum_{j=1}^{m}\|\mathbf{u}_{t-j,t}\|}{\|\mathbf{u}_t\|} = \frac{K_c}{K}. \tag{4.2}$$

The code rate of the PIC-TCs with $m \geq 1$ is

$$R = \frac{\sum_{t=1}^{L}\|\mathbf{u}'_t\|}{\sum_{t=1}^{L}\|\mathbf{u}'_t\| + \|\mathbf{v}_t\|} = \frac{L - (L + \frac{m+1}{2})\lambda}{3L - (L + \frac{m+1}{2})\lambda} \overset{L\to\infty}{=} \frac{1-\lambda}{3-\lambda}. \tag{4.3}$$

Example 4.1. Consider the PIC-TC with $L = 5$, $K = 100$, $\lambda = 0.2$, and $m = 2$. An information sequence $\mathbf{u}$ is segmented into $\mathbf{u}'_1, \ldots, \mathbf{u}'_5$, where $\|\mathbf{u}'_1\| = \|\mathbf{u}'_2\| = \|\mathbf{u}'_3\| = 80$, $\|\mathbf{u}'_4\| = 70$, and $\|\mathbf{u}'_5\| = 60$. The padding sequences are: $\mathbf{u}_{-1,1}, \mathbf{u}_{0,1}, \mathbf{u}_{0,2}, \mathbf{u}_{4,6}, \mathbf{u}_{5,6}, \mathbf{u}_{5,7}$. The length of each padding sequence is 10. The input of each CB is as follows:

CB	$\mathbf{u}'_t$	$\mathbf{u}_t$
CB_1	$\mathbf{u}'_1 = [\mathbf{u}_{1,1}, \mathbf{u}_{1,2}, \mathbf{u}_{1,3}]$	$\mathbf{u}_1 = [\mathbf{u}_{-1,1}, \mathbf{u}_{0,1}, \mathbf{u}'_1]$
CB_2	$\mathbf{u}'_2 = [\mathbf{u}_{2,2}, \mathbf{u}_{2,3}, \mathbf{u}_{2,4}]$	$\mathbf{u}_2 = [\mathbf{u}_{0,2}, \mathbf{u}_{1,2}, \mathbf{u}'_2]$
CB_3	$\mathbf{u}'_3 = [\mathbf{u}_{3,3}, \mathbf{u}_{3,4}, \mathbf{u}_{3,5}]$	$\mathbf{u}_3 = [\mathbf{u}_{1,3}, \mathbf{u}_{2,3}, \mathbf{u}'_3]$
CB_4	$\mathbf{u}'_4 = [\mathbf{u}_{4,4}, \mathbf{u}_{4,5}]$	$\mathbf{u}_4 = [\mathbf{u}_{2,4}, \mathbf{u}_{3,4}, \mathbf{u}'_4, \mathbf{u}_{4,6}]$
CB_5	$\mathbf{u}'_5 = [\mathbf{u}_{5,5}]$	$\mathbf{u}_5 = [\mathbf{u}_{3,5}, \mathbf{u}_{4,5}, \mathbf{u}'_5, \mathbf{u}_{5,6}, \mathbf{u}_{5,7}]$

The code rate of this PIC-TC is

$$R = \frac{\sum_{i=1}^{5}\mathbf{u}'_t}{\sum_{i=1}^{5}(\mathbf{u}'_t + \mathbf{v}_t)} = \frac{80 * 3 + 70 + 60}{80 * 3 + 70 + 60 + 5 * 200} = 0.27.$$

■

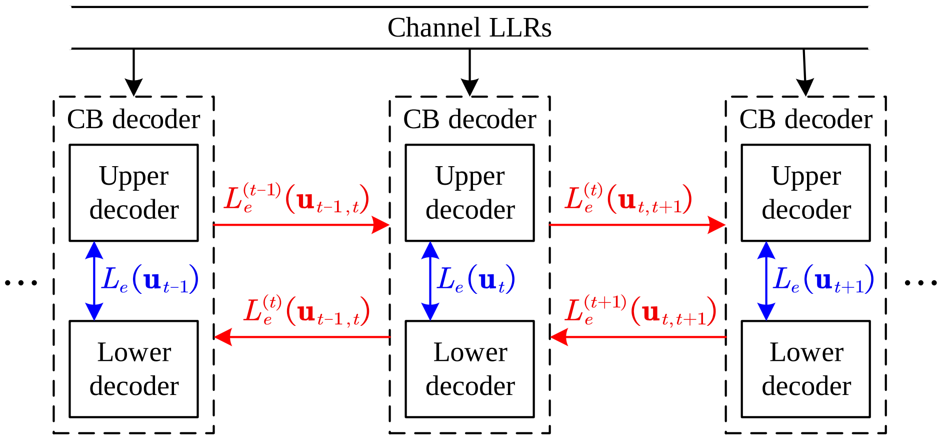

Channel LLRs
CB decoder
Upper decoder
Lower decoder
$L_e^{(t-1)}(\mathbf{u}_{t-1,t})$
$L_e^{(t)}(\mathbf{u}_{t-1,t})$
$L_e(\mathbf{u}_{t-1})$
$L_e(\mathbf{u}_t)$
$L_e(\mathbf{u}_{t+1})$
$L_e^{(t)}(\mathbf{u}_{t,t+1})$
$L_e^{(t+1)}(\mathbf{u}_{t,t+1})$
…

Figure 4.2: Decoding of PIC-TCs with $m = 1$.

4.2.2 Decoding of PIC-TCs

The decoding of PIC can be accomplished by a serial feed-forward and feed-back (FF-FB) scheme. As shown in Fig. 4.2, the FF-FB scheme includes two types of decoding iterations. One is the inner iteration (CB decoding) to decode a CB, where the upper and lower RSC decoders exchange extrinsic information between them, realized by the BCJR algorithm [47, 88]. The other is the outer iteration (global decoding), in which the extrinsic information associated with the coupled sequences is passed between consecutive CBs in forward-backward round trips.

We use $L_{ch}(\mathbf{u})$ to represent the LLRs of sequence $\mathbf{u}$. At CB$_t$, $t = 1, \ldots, L$, we use $L_a^{(t)}(\mathbf{u})$ and $L_e^{(t)}(\mathbf{u})$ to represent the a-priori LLR and extrinsic LLR associated with $\mathbf{u}$ at CB$_t$, respectively. The input of the CB decoder consists of both the channel LLRs $L_{ch}(\mathbf{u}_t)$ and $L_{ch}(\mathbf{v}_t)$, and the a-priori LLRs $L_a(\mathbf{u}_{t-j,t})$ and $L_a(\mathbf{u}_{t,t+j})$, $1 \leq j \leq m$. The a-priori LLRs $L_a^{(t)}(\mathbf{u}_{t-j,t})$ is from the extrinsic LLR $L_e^{(t-j)}(\mathbf{u}_{t-j,t})$ at CB$_{t-j}$, while $L_a^{(t)}(\mathbf{u}_{t,t+j})$ is from the extrinsic LLR $L_e^{(t-j)}(\mathbf{u}_{t,t+j})$

at CB_{t+j}. Thus, for a bit $u \in \{\mathbf{u}_t, \mathbf{v}_t\}$, its input LLR of the CB decoder is

$$L_{in}^{(t)}(u) = \begin{cases} L_{ch}(u) & \text{if } u \in \{\mathbf{u}_{t,t}, \mathbf{v}_t\}, \\ L_{ch}(u) + L_a^{(t)}(u) = L_{ch}(u) + L_e^{(t-j)}(u) & \text{if } u \in \mathbf{u}_{t-j,t}, \\ L_{ch}(u) + L_a^{(t)}(u) = L_{ch}(u) + L_e^{(t+j)}(u) & \text{if } u \in \mathbf{u}_{t,t+j}. \end{cases} \tag{4.4}$$

After CB decoder produces the a-posteriori LLR $L^{(t)}(\mathbf{u}_t)$. Then, the extrinsic LLR of a coupled bit $u \in \{\mathbf{u}_{t-j,t}, \mathbf{u}_{t,t+j}\}$ is updated following

$$L_e^{(t)}(u) = L^{(t)}(u) - L_{in}^{(t)}(u). \tag{4.5}$$

By putting (4.4) into (4.5), we can rewrite the updating of extrinsic LLRs at CB_t as follows:

$$L^{(t)}(\mathbf{u}_{t-j,t}) = L_e^{(t)}(\mathbf{u}_{t-j,t}) + L_e^{(t-j)}(\mathbf{u}_{t-j,t}) + L_{ch}(\mathbf{u}_{t-j,t}), \tag{4.6a}$$

$$L^{(t)}(\mathbf{u}_{t,t+j}) = L_e^{(t)}(\mathbf{u}_{t,t+j}) + L_e^{(t+j)}(\mathbf{u}_{t,t+j}) + L_{ch}(\mathbf{u}_{t,t+j}). \tag{4.6b}$$

4.3 PPC-Turbo Codes

In this section, we present the encoding and decoding of PPC-TCs.

4.3.1 Encoding of PPC-TCs

Fig. 4.3 depicts the encoder block diagram of a PPC-TC with $m = 1$. The encoding processes are as follows.

---

1. **Information segmentation:** The sequence $\mathbf{u}$ is segmented into L subsequences $\mathbf{u}'_1, \ldots, \mathbf{u}'_L$, where $\|\mathbf{u}'_t\| = K - K_c$ for $t = 1, \ldots, L-1$, and

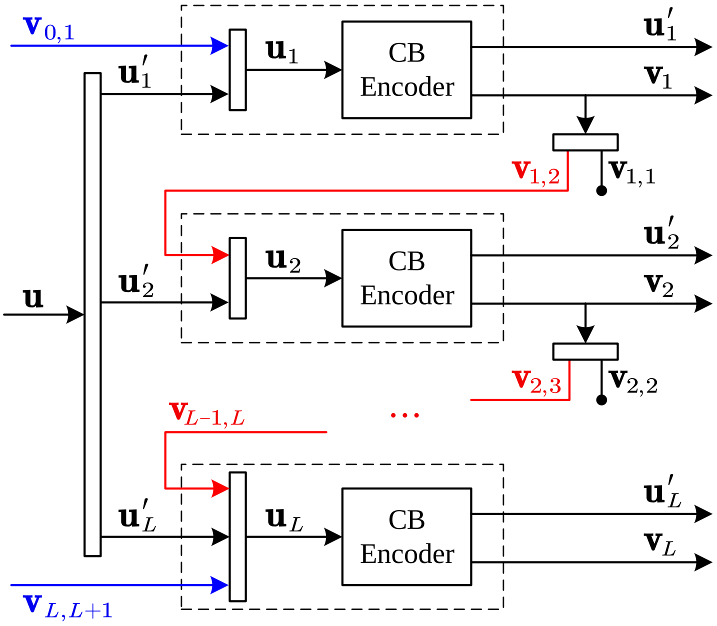


Figure 4.3: Encoding scheme of PPC-TCs.

$\|\mathbf{u}'_L\| = K - 2K_c$.

2. **Padding bits insertion:** To initialize and terminate the encoding, two all-zero blocks $\mathbf{v}_{0,1}$ and $\mathbf{v}_{L,L+1}$ of length K_c are attached before $\mathbf{u}'_1$ and after $\mathbf{u}'_L$ respectively.

3. **Information coupling and CB encoding:** For $t = 1, \ldots, L-1$, CB$_t$ takes $\mathbf{u}_t = [\mathbf{v}_{t-1,t}, \mathbf{u}'_t]$ as its encoder input, and produces the parity sequence $\mathbf{v}_t = [\mathbf{v}_t^{\mathrm{U}}, \mathbf{v}_t^{\mathrm{L}}]$. The codeword of CB$_t$ is obtained as $\mathbf{x}_t = [\mathbf{u}'_t, \mathbf{v}_t]$. Then, $\mathbf{v}_t$ is split into the uncoupled parity sequence $\mathbf{v}_{t,t}$, which only stays in $\mathbf{x}_t$, and the coupled parity sequence $\mathbf{v}_{t,t+1}$, which will become a part of the CB encoder input in time $t+1$. For $t = L$, CB$_L$ takes $\mathbf{u}_L = [\mathbf{v}_{L-1,L}, \mathbf{u}'_L, \mathbf{v}_{L,L+1}]$ as its encoder input, where $\mathbf{v}_{L,L+1}$ is a padding sequence, and produces the parity sequence $\mathbf{v}_L$. The codeword of CB$_L$ is $\mathbf{x}_L = [\mathbf{u}'_L, \mathbf{v}_L]$, i.e., the padding sequence $\mathbf{v}_{L,L+1}$ is not transmitted.

Now consider $m \geq 1$. The information sequence $\mathbf{u}$ is segmented into subsequences $\mathbf{u}'_1, \ldots, \mathbf{u}'_L$, where $\|\mathbf{u}_t\| = K - K_c$ for $t = 1, \ldots, L - m$, and $\|\mathbf{u}_t\| = K - \frac{L+2m-t}{m} K_c$ for $t = L - t + 1, \ldots, L$. For $t = 1, \ldots, L - m$, the CB encoder inputs $\mathbf{u}_t = [\mathbf{v}_{t-m,t}, \ldots, \mathbf{v}_{t-1,t}, \mathbf{u}'_t]$ and outputs the parity sequence $\mathbf{v}_t$. Then, $\mathbf{v}_t$ is segmented into the uncoupled sequence $\mathbf{v}_{t,t}$ and the coupled subsequences $\mathbf{v}_{t,t+j}$, $1 \leq j \leq m$. For $t = L - t + 1, \ldots, L$, the CB encoder inputs $\mathbf{u}_t = [\mathbf{v}_{t-m,t}, \ldots, \mathbf{v}_{t-1,t}, \mathbf{u}'_t, \mathbf{v}_{t,t+1}, \ldots, \mathbf{v}_{t,t+m-L}]$, where $\mathbf{v}_{t,t+1}, \ldots, \mathbf{v}_{t,t+m-L}$ are the padding sequences, and outputs the parity sequence $\mathbf{v}_t$. Then, $\mathbf{v}_t$ is segmented into the uncoupled sequence $\mathbf{v}_{t,t}$ and the coupled subsequences $\mathbf{v}_{t,t+j}$, $1 \leq j \leq L - t$. For $t = 1, \ldots, L$, the codeword of CB_t is obtained as $\mathbf{x}_t = [\mathbf{u}'_t, \mathbf{v}_t]$.

The length of a coupled (or padding) sequence is $\|\mathbf{v}_{t,t+j}\| = \frac{K_c}{m}$, $1 \leq j \leq m$. The coupling ratio of PPC-TCs is defined as

$$\lambda \triangleq \frac{\sum_{j=1}^{m} \|\mathbf{v}_{t-j,t}\|}{\|\mathbf{u}_t\|} = \frac{K_c}{K} \tag{4.7}$$

The code rate expression of PPC-TCs is the same as that of PIC-TCs in (4.3), so that PIC-TCs and PPC-TCs have the same code rate when they have the same coupling memory and coupling ratio.

Example 4.2. Consider the PPC-TC with $L = 5$, $K = 100$, $\lambda = 0.2$, and $m = 2$. An information sequence $\mathbf{u}$ is segmented into $\mathbf{u}'_1, \ldots, \mathbf{u}'_5$, where $\|\mathbf{u}'_1\| = \|\mathbf{u}'_2\| = \|\mathbf{u}'_3\| = 80$, $\|\mathbf{u}'_4\| = 70$, and $\|\mathbf{u}'_5\| = 60$. The padding sequences are: $\mathbf{v}_{-1,1}, \mathbf{v}_{0,1}, \mathbf{v}_{0,2}, \mathbf{v}_{4,6}, \mathbf{v}_{5,6}, \mathbf{v}_{5,7}$. The length of one padding sequence is 10. The information input and the parity sequence of each CB are as follows:

CB	Information input	Parity sequence
CB$_1$	$\mathbf{u}_1 = [\mathbf{v}_{-1,1}, \mathbf{v}_{0,1}, \mathbf{u}'_1]$	$\mathbf{v}_1 = [\mathbf{v}_{1,1}, \mathbf{v}_{1,2}, \mathbf{v}_{1,3}]$
CB$_2$	$\mathbf{u}_2 = [\mathbf{v}_{0,2}, \mathbf{v}_{1,2}, \mathbf{u}'_2]$	$\mathbf{v}_2 = [\mathbf{v}_{2,2}, \mathbf{v}_{2,3}, \mathbf{v}_{2,3}]$
CB$_3$	$\mathbf{u}_3 = [\mathbf{v}_{1,3}, \mathbf{v}_{2,3}, \mathbf{u}'_3]$	$\mathbf{v}_3 = [\mathbf{v}_{3,3}, \mathbf{v}_{3,4}, \mathbf{v}_{3,5}]$
CB$_4$	$\mathbf{u}_4 = [\mathbf{v}_{2,4}, \mathbf{v}_{3,4}, \mathbf{u}'_4, \mathbf{v}_{4,6}]$	$\mathbf{v}_4 = [\mathbf{v}_{4,4}, \mathbf{v}_{4,5}]$
CB$_5$	$\mathbf{u}_5 = [\mathbf{v}_{3,5}, \mathbf{v}_{4,5}, \mathbf{u}'_5, \mathbf{v}_{5,6}, \mathbf{v}_{5,7}]$	$\mathbf{v}_5 = [\mathbf{v}_{5,5}]$

The code rate of this PPC-TC is

$$R = \frac{\sum_{i=1}^{5} \mathbf{u}'_t}{\sum_{i=1}^{5} (\mathbf{u}'_t + \mathbf{v}_t)} = \frac{80 * 3 + 70 + 60}{80 * 3 + 70 + 60 + 5 * 200} = 0.27.$$

■

Remark 4.2. At time t, we consider that a coupled sequence $\mathbf{v}_{t,t+j}$ consists of the parity bits from both $\mathbf{v}_t^{\mathrm{U}}$ and $\mathbf{v}_t^{\mathrm{L}}$. In other words, $\mathbf{v}_{t,t+j} = [\mathbf{v}_{t,t+j}^{\mathrm{U}}, \mathbf{v}_{t,t+j}^{\mathrm{L}}]$. In addition, we consider $\|\mathbf{v}_{t,t+j}^{\mathrm{U}}\| = \|\mathbf{v}_{t,t+j}^{\mathrm{U}}\| = \frac{K_c}{2m}$. ◀

4.3.2 Decoding of PPC-TCs

Similar to PIC-TCs, the decoding of PPC can be accomplished by the FF-FB scheme. Consider CB$_t$, $1 \leq t \leq L$. Inside the CB decoder, the upper and lower RSC decoders exchange extrinsic LLR of $\mathbf{u}_t$ between them. For the global decoding, the extrinsic LLRs associated with the coupled sequences are updated in forward-backward round trips following

$$L^{(t)}(\mathbf{v}_{t-j,t}) = L_e^{(t)}(\mathbf{v}_{t-j,t}) + L_e^{(t-j)}(\mathbf{v}_{t-j,t}) + L_{ch}(\mathbf{v}_{t-j,t}), \tag{4.8a}$$

$$L^{(t)}(\mathbf{v}_{t,t+j}) = L_e^{(t)}(\mathbf{v}_{t,t+j}) + L_e^{(t+j)}(\mathbf{v}_{t,t+j}) + L_{ch}(\mathbf{v}_{t,t+j}). \tag{4.8b}$$

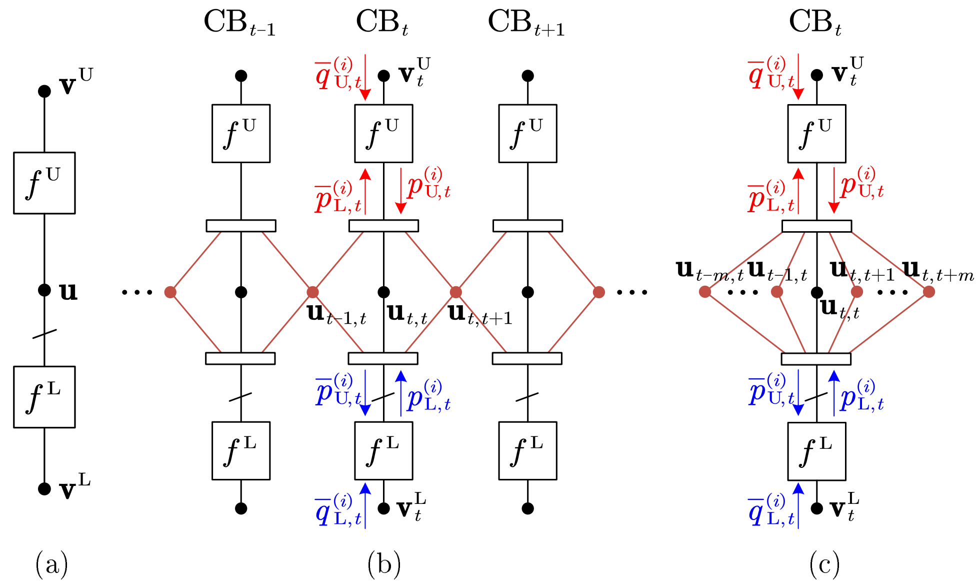

CB$_{t-1}$
CB$_t$
CB$_{t+1}$
CB$_t$
$\mathbf{v}^{\mathrm{U}}$
f^{U}
$\mathbf{u}$
f^{L}
$\mathbf{v}^{\mathrm{L}}$
$\overline{q}^{(i)}_{\mathrm{U},t}$
$\mathbf{v}^{\mathrm{U}}_t$
$\overline{p}^{(i)}_{\mathrm{L},t}$
$p^{(i)}_{\mathrm{U},t}$
$\mathbf{u}_{t-1,t}$
$\mathbf{u}_{t,t}$
$\mathbf{u}_{t,t+1}$
$\overline{p}^{(i)}_{\mathrm{U},t}$
$p^{(i)}_{\mathrm{L},t}$
$\overline{q}^{(i)}_{\mathrm{L},t}$
$\mathbf{v}^{\mathrm{L}}_t$
$\mathbf{u}_{t-m,t}$
$\mathbf{u}_{t,t+m}$
(a)
(b)
(c)

Figure 4.4: Compact graph of a) uncoupled turbo codes, b) PIC-TCs with $m = 1$, and c) PIC-TCs with $m \geq 1$.

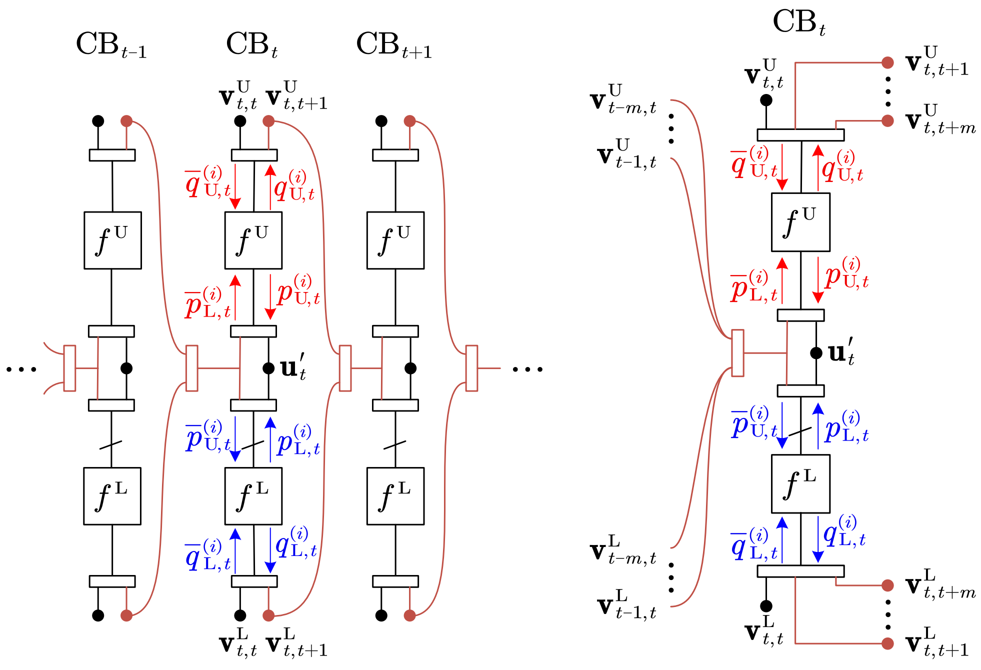
4.4 Density Evolution Analysis

In this section, we present the DE analysis of the PIC-TCs and PPC-TCs over the BEC. We first present the graph models and introduce the relevant notations. Then, we derive the DE equations.

4.4.1 Graph Model Representation

We first start with the compact graph [38] of uncoupled turbo codes. As shown in Fig. 4.4a, we use two factor nodes f^{U} and f^{L} to represent the upper and lower RSC decoders, respectively. The information sequence $\mathbf{u}$ is represented by a variable node (VN) connected to f^{U} and f^{L}. The two VNs, which only connect to one of the factor nodes, represent the parity sequences $\mathbf{v}^{\mathrm{U}}$ and $\mathbf{v}^{\mathrm{L}}$, respectively. The interleaver is represented by a cross on the edge between $\mathbf{u}$ and f^{L}.

Fig. 4.4b shows the compact graph of the PIC-TC ensemble with coupling memory $m = 1$. At time t, we use three VNs to represent $\mathbf{u}_t$ by treating the


CB$_{t\text{-}1}$
CB$_t$
CB$_{t+1}$
$\mathbf{v}^{\mathrm{U}}_{t,t}$ $\mathbf{v}^{\mathrm{U}}_{t,t+1}$
$\overline{q}^{(i)}_{\mathrm{U},t}$
$q^{(i)}_{\mathrm{U},t}$
f^{U}
$\overline{p}^{(i)}_{\mathrm{L},t}$
$p^{(i)}_{\mathrm{U},t}$
$\mathbf{u}'_t$
$\overline{p}^{(i)}_{\mathrm{U},t}$
$p^{(i)}_{\mathrm{L},t}$
f^{L}
$\overline{q}^{(i)}_{\mathrm{L},t}$
$q^{(i)}_{\mathrm{L},t}$
$\mathbf{v}^{\mathrm{L}}_{t,t}$ $\mathbf{v}^{\mathrm{L}}_{t,t+1}$
$\mathbf{v}^{\mathrm{U}}_{t\text{-}m,t}$
$\mathbf{v}^{\mathrm{U}}_{t\text{-}1,t}$
$\mathbf{v}^{\mathrm{U}}_{t,t+1}$
$\mathbf{v}^{\mathrm{U}}_{t,t+m}$
$\mathbf{v}^{\mathrm{L}}_{t\text{-}m,t}$
$\mathbf{v}^{\mathrm{L}}_{t\text{-}1,t}$
$\mathbf{v}^{\mathrm{L}}_{t,t+m}$
$\mathbf{v}^{\mathrm{L}}_{t,t+1}$

(a) (b)

Figure 4.5: Compact graph of a) PPC-TCs with $m = 1$ and b) PPC-TCs with $m \geq 1$.

uncoupled sequence $\mathbf{u}_{t,t}$ and the coupled sequences $\mathbf{u}_{t-1,t}$ and $\mathbf{u}_{t,t-1}$ separately. The VN $\mathbf{u}_{t,t-1}$ is connected to f^{U} and f^{L} at time $t-1$ and t, meaning that its extrinsic information will be propagated between the upper and lower decoders at time $t-1$ and t. In the same token, $\mathbf{u}_{t,t-1}$ is connected to f^{U} and f^{L} at time $t-1$ and t, meaning that its extrinsic information will be propagated between the upper and lower decoders at time t and $t+1$.

The generalized compact graph of the PIC-TC ensemble with $m \geq 1$ is depicted in Fig. 4.4c. The information sequence $\mathbf{u}_t$ is represented by $2m+1$ VNs, including an uncoupled VN $\mathbf{u}_{t,t}$ and $2m$ coupled VNs $\mathbf{u}_{t,t-m}, \dots, \mathbf{u}_{t,t-1}$ and $\mathbf{u}_{t,t+1}, \dots, \mathbf{u}_{t,t+m}$. The coupled VN $\mathbf{u}_{t,t-j}$ is connected to both time $t-j$ and time t, while $\mathbf{u}_{t,t+j}$ is connected to both time t and time $t+j$, $1 \leq j \leq m$.

Fig. 4.5a depicts the compact graph of the PPC-TC ensemble with $m = 1$. We use one VN to represent $\mathbf{u}'_t$ and four VNs to represent $\mathbf{v}_t$ by treating $\mathbf{v}^{\mathrm{U}}_{t,t}$,

$\mathbf{v}^{\mathrm{U}}_{t,t+1}$, $\mathbf{v}^{\mathrm{L}}_{t,t}$, and $\mathbf{v}^{\mathrm{L}}_{t,t+1}$] separately . The VNs $\mathbf{v}^{\mathrm{U}}_{t,t+1}$ and $\mathbf{v}^{\mathrm{L}}_{t,t+1}$ are connected to the information input of f^{U} and f^{L} at time $t+1$, meaning that the extrinsic information of $\mathbf{v}^{\mathrm{U}}_{t,t+1}$ will be propagated between the upper decoder at time t, and the upper and lower decoders at time $t+1$, while the extrinsic information of $\mathbf{v}^{\mathrm{L}}_{t,t+1}$ will be propagated between the lower decoder at time t, and the upper and lower decoders at time $t+1$. Likewise, the information input of f^{U} and f^{L} at time t are connected to the VNs $\mathbf{v}^{\mathrm{U}}_{t-1,t}$ and $\mathbf{v}^{\mathrm{L}}_{t-1,t}$ at time $t-1$.

The generalized compact graph of the PPC-TC ensemble with $m \geq 1$ is depicted in Fig. 4.5b. The parity sequence $\mathbf{v}_t$ is represented by $2m+2$ VNs, including two uncoupled VNs $\mathbf{v}^{\mathrm{U}}_{t,t}$ and $\mathbf{v}^{\mathrm{L}}_{t,t}$, and $2m$ coupled VNs $\mathbf{v}^{\mathrm{U}}_{t,t+1}, \ldots, \mathbf{v}^{\mathrm{U}}_{t,t+m}$ and $\mathbf{v}^{\mathrm{L}}_{t,t+1}, \ldots, \mathbf{v}^{\mathrm{L}}_{t,t+m}$. The coupled VNs $\mathbf{v}^{\mathrm{U}}_{t,t+j}$ and $\mathbf{v}^{\mathrm{L}}_{t,t+j}$ are connected to the information input of f^{U} and f^{L} at time $t+j$, while the information input of f^{U} and f^{L} at time t are connected to VNs $\mathbf{v}^{\mathrm{U}}_{t-j,t}$ and $\mathbf{v}^{\mathrm{L}}_{t-j,t}$ at time $t-j$, $1 \leq j \leq m$.

4.4.2 Notations and Definitions

Now we introduce some notations before we analyze our codes in the following subsections. Let ε be the BEC erasure probability. The input-output relations of f^{U} are described by the transfer functions: F^{U}_1 and F^{U}_2, corresponding to the information sequence and the parity sequence, respectively. Likewise, the input-output relations of f^{L} are described by the transfer functions F^{L}_1 and F^{L}_2, respectively. The transfer functions are derived following [82]. At time t and the

Table 4.1: Notations for the DE analysis of PIC-TCs and PPC-TCs.

Notation	Definition
$\bar{p}_{\mathrm{L},t}^{(i)}$	Information input erasure probability of f^{U}
$\bar{q}_{\mathrm{U},t}^{(i)}$	Parity input erasure probability of f^{U}
$p_{\mathrm{U},t}^{(i)}$	Information output erasure probability of f^{U}
$q_{\mathrm{U},t}^{(i)}$	Parity output erasure probability of f^{U}
$\bar{p}_{\mathrm{U},t}^{(i)}$	Information input erasure probability of f^{L}
$\bar{q}_{\mathrm{L},t}^{(i)}$	Parity input erasure probability of f^{L}
$p_{\mathrm{L},t}^{(i)}$	Information output erasure probability of f^{L}
$q_{\mathrm{L},t}^{(i)}$	Parity output erasure probability of f^{L}

i-th decoding iteration, the DE equations of f^{U} and f^{L} are

$$f^{\mathrm{U}}:\begin{cases} p_{\mathrm{U},t}^{(i)}=F_1^{\mathrm{U}}\left(\bar{p}_{\mathrm{L},t}^{(i)},\bar{q}_{\mathrm{U},t}^{(i)}\right)\\ q_{\mathrm{U},t}^{(i)}=F_2^{\mathrm{U}}\left(\bar{p}_{\mathrm{L},t}^{(i)},\bar{q}_{\mathrm{U},t}^{(i)}\right)\end{cases}, \tag{4.9a}$$

$$f^{\mathrm{L}}:\begin{cases} p_{\mathrm{L},t}^{(i)}=F_1^{\mathrm{L}}\left(\bar{p}_{\mathrm{U},t}^{(i)},\bar{q}_{\mathrm{L},t}^{(i)}\right)\\ q_{\mathrm{L},t}^{(i)}=F_2^{\mathrm{L}}\left(\bar{p}_{\mathrm{U},t}^{(i)},\bar{q}_{\mathrm{L},t}^{(i)}\right)\end{cases}. \tag{4.9b}$$

The notations for the inputs and outputs of the above transfer functions are in Table 4.1.

4.4.3 Density Evolution of PIC-TCs

We first consider the PIC-TC ensemble with coupling memory $m = 1$. For f^{U} at time t and the i-th iteration, $\bar{p}_{\mathrm{L},t}^{(i)}$ depends on the erasure probabilities of VNs $\mathbf{u}_{t-1,t}$, $\mathbf{u}_{t,t}$, and $\mathbf{u}_{t,t+1}$. These three VNs can obtain extrinsic information from f^{L} at time t. In addition, $\mathbf{u}_{t-1,t}$ can further obtain extrinsic information from f^{L} and f^{U} at time $t-1$, while $\mathbf{u}_{t,t+1}$ can further obtain extrinsic information from f^{L} and f^{U} at time $t+1$. Meanwhile, $\bar{q}_{\mathrm{U},t}^{(i)}$ only depends on the channel erasure probability. Thus, the input erasure probabilities of f^{U} are computed as

$$\bar{p}_{\mathrm{L},t}^{(i)} = \varepsilon \cdot p_{\mathrm{L},t}^{(i)} \cdot \left(1 - 2\lambda + \lambda \cdot p_{\mathrm{U},t-1}^{(i-1)} \cdot p_{\mathrm{L},t-1}^{(i)} + \lambda \cdot p_{\mathrm{U},t+1}^{(i-1)} \cdot p_{\mathrm{L},t+1}^{(i)}\right), \tag{4.10a}$$

$$\bar{q}_{\mathrm{U},t}^{(i)} = \varepsilon. \tag{4.10b}$$

Consider coupling memory $m \geq 1$. For f^{U} at time t and the i-th iteration, $\bar{p}_{\mathrm{L},t}^{(i)}$ depends on the erasure probabilities of VNs $\mathbf{u}_{t-m,t}, \ldots, \mathbf{u}_{t-1,t}$, $\mathbf{u}_{t,t}$, and $\mathbf{u}_{t,t+1}, \ldots, \mathbf{u}_{t,t+m}$. All VNs can obtain extrinsic information from f^{L} at time t. In addition, $\mathbf{u}_{t-j,t}$ can further obtain extrinsic information from f^{L} and f^{U} at time $t-i$, while $\mathbf{u}_{t,t+j}$ can further obtain extrinsic information from f^{L} and f^{U} at time $t+j$, $1 \leq j \leq m$. Meanwhile, $\bar{q}_{\mathrm{U},t}^{(i)}$ only depends on the channel erasure probability. Thus, the input erasure probabilities of f^{U} are computed as

$$\bar{p}_{\mathrm{L},t}^{(i)} = \varepsilon \cdot p_{\mathrm{L},t}^{(i)} \cdot \left(1 - 2\lambda + \frac{\lambda}{m} \sum_{j=1}^{m} p_{\mathrm{U},t-j}^{(i-1)} \cdot p_{\mathrm{L},t-j}^{(i)} + \frac{\lambda}{m} \sum_{j=1}^{m} p_{\mathrm{U},t+j}^{(i-1)} \cdot p_{\mathrm{L},t+j}^{(i)}\right), \tag{4.11a}$$

$$\bar{q}_{\mathrm{U},t}^{(i)} = \varepsilon. \tag{4.11b}$$

We omit the density evolution functions for f^{L} here and also for the rest analysis as it can be derived in the same manner to that for f^{U}. With the knowledge of $p_{\mathrm{U},t}^{(i)}$ and $p_{\mathrm{L},t}^{(i)}$, the a-posteriori erasure probability of $\mathbf{u}_t$ after i iterations is

$$p_{\mathbf{u}_t}^{(i)} = \varepsilon \cdot p_{\mathrm{U},t}^{(i)} \cdot p_{\mathrm{L},t}^{(i)} \cdot \left(1 - 2\lambda + \frac{\lambda}{m} \sum_{j=1}^{m} p_{\mathrm{U},t-j}^{(i)} \cdot p_{\mathrm{L},t-j}^{(i)} + \frac{\lambda}{m} \sum_{j=1}^{m} p_{\mathrm{U},t+j}^{(i)} \cdot p_{\mathrm{L},t+j}^{(i)}\right). \tag{4.12}$$

4.4.4 Density Evolution of PPC-TCs

Consider the PPC-TC ensemble with $m = 1$. For f^{U} at time t and the i-th iteration, $\bar{p}^{(i)}_{\mathrm{L},t}$ depends on the erasure probabilities of VNs $\mathbf{u}'_t$, $\mathbf{v}^{\mathrm{U}}_{t-1,t}$, and $\mathbf{v}^{\mathrm{L}}_{t-1,t}$, where $\mathbf{u}'_t$ can obtain extrinsic information from f^{L} at time t, $\mathbf{v}^{\mathrm{U}}_{t-1,t}$ can obtain extrinsic information from f^{U} at time $t-1$, and $\mathbf{v}^{\mathrm{L}}_{t-1,t}$ can obtain extrinsic information from f^{L} at time $t-1$. Meanwhile, $\bar{q}^{(i)}_{\mathrm{U},t}$ depends on the erasure probabilities of VNs $\mathbf{v}^{\mathrm{U}}_{t,t}$ and $\mathbf{v}^{\mathrm{U}}_{t,t+1}$, where $\mathbf{v}^{\mathrm{U}}_{t,t+1}$ can obtain extrinsic information from f^{U} and f^{L} at time $t+1$. Thus, the input erasure probabilities of f^{U} are computed as

$$\bar{p}^{(i)}_{\mathrm{L},t} = \varepsilon \cdot p^{(i)}_{\mathrm{L},t} \cdot \left(1 - \lambda + \frac{\lambda}{2} \cdot q^{(i-1)}_{\mathrm{U},t-1} + \frac{\lambda}{2} \cdot q^{(i)}_{\mathrm{L},t-1}\right), \tag{4.13a}$$

$$\bar{q}^{(i)}_{\mathrm{U},t} = \varepsilon \cdot \left(1 - \frac{\lambda}{2} + \frac{\lambda}{2} \cdot p^{(i-1)}_{\mathrm{U},t+1} \cdot p^{(i)}_{\mathrm{L},t+1}\right). \tag{4.13b}$$

Consider coupling memory $m \geq 1$. For f^{U} at time t and the i-th iteration, $\bar{p}^{(i)}_{\mathrm{L},t}$ depends on the erasure probabilities of VNs $\mathbf{u}'_t$, $\mathbf{v}^{\mathrm{U}}_{t-m,t}, \ldots, \mathbf{v}^{\mathrm{U}}_{t-1,t}$, and $\mathbf{v}^{\mathrm{L}}_{t-m,t}, \ldots, \mathbf{v}^{\mathrm{L}}_{t-1,t}$. To be specific, $\mathbf{u}'_t$ can obtain extrinsic information from f^{L} at time t, while $\mathbf{v}^{\mathrm{U}}_{t-j,t}/\mathbf{v}^{\mathrm{U}}_{t-j,t}$ can obtain extrinsic information from $f^{\mathrm{U}}/f^{\mathrm{L}}$ at time $t-j$, $1 \leq j \leq m$. Meanwhile, $\bar{q}^{(i)}_{\mathrm{U},t}$ depends on the erasure probabilities of VNs $\mathbf{v}^{\mathrm{U}}_{t,t}$ and $\mathbf{v}^{\mathrm{U}}_{t,t+1}, \ldots, \mathbf{v}^{\mathrm{U}}_{t,t+m}$, where $\mathbf{v}^{\mathrm{U}}_{t,t+j}$ can obtain extrinsic information from f^{U} and f^{L} at time $t+j$, $1 \leq j \leq m$. The input erasure probabilities of f^{U} are computed as

$$\bar{p}^{(i)}_{\mathrm{L},t} = \varepsilon \cdot p^{(i)}_{\mathrm{L},t} \cdot \left(1 - \lambda + \frac{\lambda}{2m} \sum_{j=1}^{m} q^{(i-1)}_{\mathrm{U},t-j} + \frac{\lambda}{2m} \sum_{j=1}^{m} q^{(i)}_{\mathrm{L},t-j}\right), \tag{4.14a}$$

$$\bar{q}^{(i)}_{\mathrm{U},t} = \varepsilon \cdot \left(1 - \frac{\lambda}{2} + \frac{\lambda}{2m} \sum_{j=1}^{m} p^{(i-1)}_{\mathrm{U},t+j} \cdot p^{(i)}_{\mathrm{L},t+j}\right). \tag{4.14b}$$

With the knowledge of $p_{\mathrm{U},t}^{(i)}$, $p_{\mathrm{L},t}^{(i)}$, $\mathbf{v}_{t-m,t}^{\mathrm{U}},\dots,\mathbf{v}_{t-1,t}^{\mathrm{U}}$, and $\mathbf{v}_{t-m,t}^{\mathrm{L}},\dots,\mathbf{v}_{t-1,t}^{\mathrm{L}}$, the a-posteriori erasure probability of $\mathbf{u}_t$ after i iterations is

$$p_{\mathbf{u}_t}^{(i)} = \varepsilon \cdot p_{\mathrm{U},t}^{(i)} \cdot p_{\mathrm{L},t}^{(i)} \cdot \Big(1 - \lambda + \frac{\lambda}{2m}\sum_{j=1}^{m} p_{\mathrm{U},t-j}^{(i)} + \frac{\lambda}{2m}\sum_{j=1}^{m} p_{\mathrm{L},t-j}^{(i)}\Big). \tag{4.15}$$

4.4.5 Density Evolution with Random Puncturing

We have noted that partial coupling brings rate loss. When using a rate-$\frac{1}{3}$ turbo code to as the building block, the PIC-TCs and PPC-TCs have a rate below $\frac{1}{3}$. Thus, external rate-matching techniques are required to reach $R \geq \frac{1}{3}$. We consider the rate-matching design of our codes by randomly puncturing the parity bits.

Let δ represent the puncturing ratio, which is the fraction of punctured bits in the parity sequence of a CB. The code rate of a punctured partially coupled turbo code with coupling ratio λ, coupling memory m, and puncturing ratio δ is

$$R = \frac{L - (L + \frac{m+1}{2})\lambda}{L(3-2\delta) - (L + \frac{m+1}{2})\lambda} \overset{L\to\infty}{=} \frac{1-\lambda}{3-2\delta-\lambda}. \tag{4.16}$$

On the other hand, we can obtain δ when given a target code rate R as

$$\delta \overset{L\to\infty}{=} \frac{3-\lambda}{2} - \frac{1-\lambda}{2R}. \tag{4.17}$$

Given BEC erasure probability ε, the parity erasure probability from channel becomes $\varepsilon' = \varepsilon \cdot (1-\delta) + \delta$. For the DE analysis of PIC-TCs, we simply substitute $\bar{q}_{\mathrm{U},t}^{(i)} = \bar{q}_{\mathrm{L},t}^{(i)} = \varepsilon'$. For the DE analysis of PPC-TCs, we substitute ε with ε' in (4.13b) and (4.14b).

Table 4.2: BP decoding thresholds for PIC-TCs and PPC-TCs

λ	Rate	Ensemble	$\varepsilon_{\mathrm{BP}}^{(m=1)}$	$\varepsilon_{\mathrm{BP}}^{(m=5)}$
0.1	0.3103	PIC-TC	0.6743	0.6749
		PPC-TC	0.6732	0.6741
0.2	0.2857	PIC-TC	0.7038	0.7053
		PPC-TC	0.7015	0.7036
0.3	0.2593	PIC-TC	0.7327	0.7351
		PPC-TC	0.7298	0.7332
0.4	0.2308	PIC-TC	0.7621	0.7655
		PPC-TC	0.7590	0.7637
0.5	0.2000	PIC-TC	0.7926	0.7974
		PPC-TC	0.7899	0.7958

4.4.6 BP Decoding Thresholds

We now evaluate the BP decoding thresholds of our codes over the BEC by using the derived DE equations. We consider that the underlying component code is a rate-$\frac{1}{3}$ turbo code with generator $\mathbf{G} = [1, \frac{5}{7}]$.

For a given coupling memory m, the BP decoding thresholds $\varepsilon_{\mathrm{BP}}^{(m)}$ of PIC-TCs and PPC-TCs over coupling ratio λ are shown in Table 4.2. In addition, we plot the gap from the decoding thresholds to the BEC capacity $(1 - R - \varepsilon_{\mathrm{BP}}^{(m)})$ for PIC-TCs and PPC-TCs in Fig. 4.6. From these numerical results, it can be seen that the decoding thresholds of both codes are close to the BEC capacity. It is also observed that PIC-turbos perform better than PPC-TCs when they have the same coupling ratio and coupling memory.

In Fig. 4.7, we plot the decoding thresholds of PIC-TCs and PPC-TCs at $R = \frac{1}{3}$ for various coupling ratio λ when the parity bits are randomly punctured. It can be observed that simply maximizing λ may not promise the optimal decoding threshold. Thus, we optimize the λ to maximize the decoding thresholds.

The maximum decoding thresholds and the corresponding λ (which uniquely

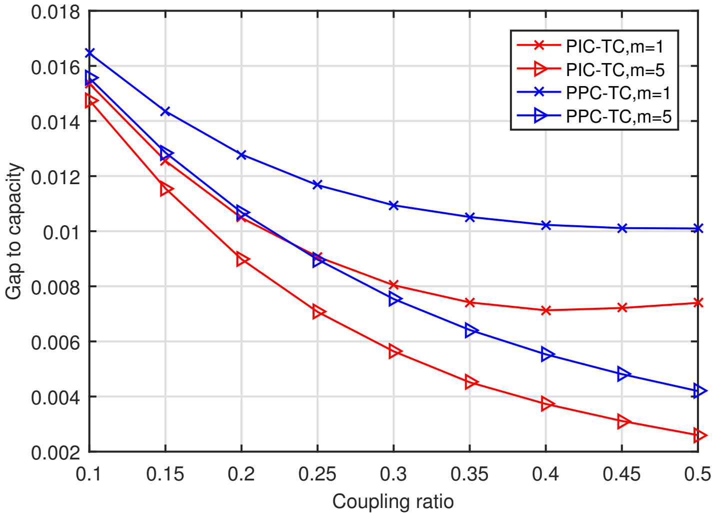


Figure 4.6: Gap to capacity for PIC-TCs and PPC-TCs.

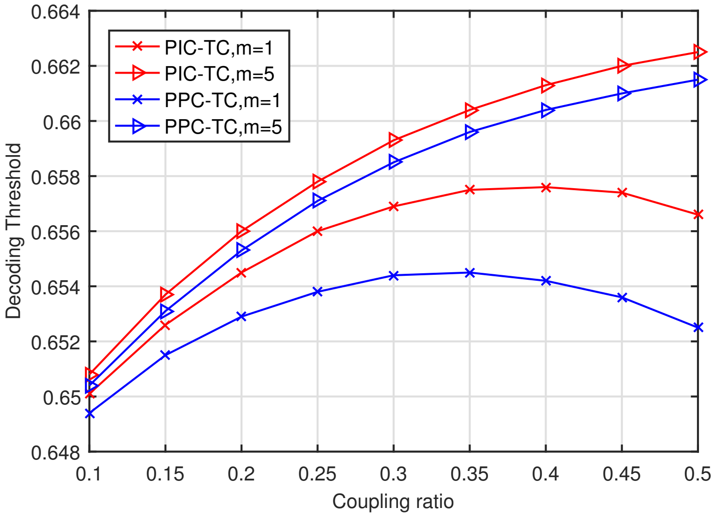


Figure 4.7: BP decoding thresholds for rate-$\frac{1}{3}$ PIC-TCs and PPC-TCs.

determines puncturing ratio δ according to (4.17)) of PIC-TCs and PPC-TCs are reported in Table 4.3 for various code rates. For both codes with coupling memory $m = 1$, the gap to the BEC capacity is less than 0.02 for all presented rates. Furthermore, the decoding thresholds of both codes improve with increasing coupling memory. We also note that PPC-TCs outperform PIC-TCs for $R \geq \frac{2}{3}$. For comparison purposes, we also include the decoding thresholds of SC-PCCs [38] in the same table. For the thresholds, it can be seen that both PIC-TCs and

Table 4.3: BP decoding thresholds for PIC-TCs and PPC-TCs for $R \geq \frac{1}{3}$.

Rate	Ensemble	$\varepsilon_{\mathrm{BP}}^{(m=1)}$, λ	$\varepsilon_{\mathrm{BP}}^{(m=15)}$, λ	$\varepsilon_{\mathrm{BP}}^{(m=50)}$, λ
1/3	PIC-TC	0.6576, $[0.37, 0.42]$	0.6635, 0.5	0.6639, 0.5
	PPC-TC	0.6545, $[0.32, 0.35]$	0.6645, $[0.43, 0.47]$	0.6658, $[0.50, 0.64]$
	SC-PCC	0.6553	0.6553	0.6553
1/2	PIC-TC	0.4865, $[0.44, 0.48]$	0.4930, 0.5	0.4933, 0.5
	PPC-TC	0.4865, $[0.32, 0.33]$	0.4977, $[0.51, 0.58]$	0.4991, $[0.59, 0.73]$
	SC-PCC	0.4689	0.4689	0.4689
2/3	PIC-TC	0.3151, 0.5	0.3204, 0.5	0.3206, 0.5
	PPC-TC	0.3206, $[0.30, 0.31]$	0.3313, $[0.54, 0.61]$	0.3325, $[0.61, 0.76]$
	SC-PCC	0.2772	0.2772	0.2772
3/4	PIC-TC	0.2307, 0.5	0.2349, 0.5	0.2351, 0.5
	PPC-TC	0.2385, $[0.28, 0.30]$	0.2483, $[0.56, 0.67]$	0.2493, $[0.65, 0.80]$
	SC-PCC	0.1876	0.1876	0.1876
4/5	PIC-TC	0.1811, 0.5	0.1846, 0.5	0.1847, 0.5
	PPC-TC	0.1896, $[0.25, 0.28]$	0.1985, $[0.64, 0.72]$	0.1994, $[0.76, 0.83]$
	SC-PCC	0.1391	0.1391	0.1391
9/10	PIC-TC	0.0863, 0.5	0.0881, 0.5	0.0881, 0.5
	PPC-TC	0.0931, $[0.19, 0.20]$	0.0990, $[0.69, 0.78]$	0.0996, $[0.78, 0.88]$
	SC-PCC	0.0582	0.0582	0.0582

PPC-TCs can achieve significant coding gains over SC-PCCs.

4.5 Simulation Results

To verify our DE analysis, we simulate the error performance of our codes over the BEC. The error performance is measured in terms of bit erasure rate (BER) versus the channel erasure probability. FF-FB decoding is considered. For all the simulated results, we set $K = 10^5$ and $m = 1$. In addition, we use $L = 100$ to minimize the rate loss due to padding sequences. Both interleavers and the positions of coupled bits are randomly chosen.

In Fig. 4.8, we show the performance of both codes with $\lambda \in \{\frac{1}{8}, \frac{1}{3}, \frac{1}{2}\}$. The corresponding code rates are $R \in \{0.3043, 0.2500, 0.2000\}$. We also plot the

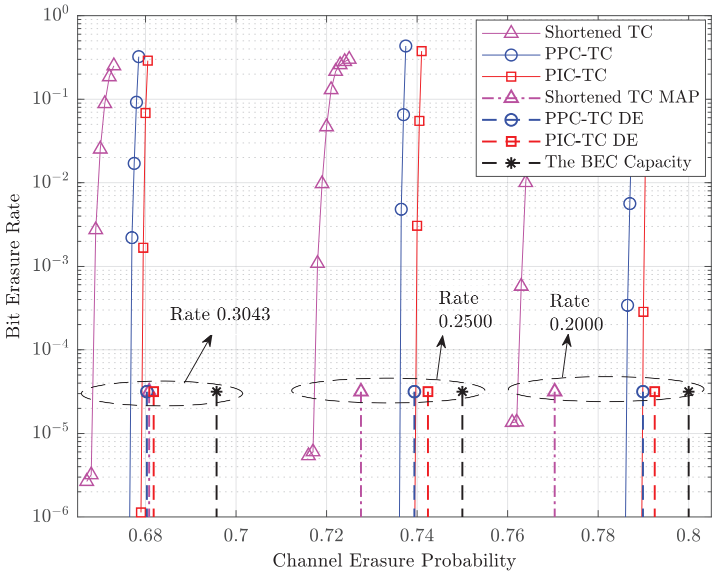


Figure 4.8: Error performance of PIC-TCs and PPC-TCs with $\lambda \in \{\frac{1}{8}, \frac{1}{3}, \frac{1}{2}\}$.

performance curves of the uncoupled turbo codes, which have $K = 10^5$ and are shortened to have the same code rates. Additionally, we include the BP decoding thresholds of our codes and the MAP decoding thresholds [95] of shortened turbo codes. From Fig. 4.8, it can be observed that our codes significantly outperform the shortened turbo codes for all code rates we have simulated. Moreover, the simulated performance of our codes is in agreement with their corresponding DE thresholds.

We also evaluate the performance of our codes with random puncturing. In particular, we plot the error performance of both codes for $R \in \{\frac{2}{3}, \frac{3}{4}\}$ in Fig. 4.9. We choose $\lambda = 0.5$ for the PIC-TC with $R \in \{\frac{2}{3}, \frac{3}{4}\}$, $\lambda = 0.2870$ for the PPC-TC with $R = \frac{2}{3}$, and $\lambda = 0.3093$ for the PPC-TC with $R = \frac{2}{3}$. The BP decoding thresholds for both codes and the MAP decoding thresholds for the uncoupled turbo codes are also shown in the figure. It can be observed that both PIC-TCs and PPC-TCs perform close to their respective DE thresholds. In addition, PPC-TCs outperform PIC-TCs.

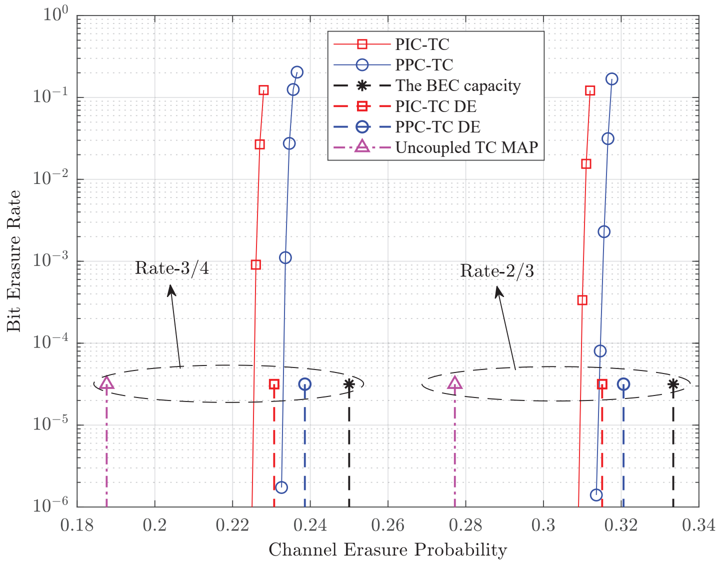


Figure 4.9: Error performance of PIC-TCs and PPC-TCs at $R \in \{\frac{2}{3}, \frac{3}{4}\}$.

4.6 Conclusion

In this chapter, we introduced two coding schemes with partial coupling and constructed PIC-TCs and PPC-TCs. We derived the exact DE equations for our codes with general coupling ratio and coupling memory over the BEC. Rate-compatible designs of our codes for high code rates are also studied. The BP decoding thresholds of our codes with optimized coupling ratio were presented for various code rates. Both theoretical analysis and simulation results showed that the proposed coding scheme can significantly improve the decoding performance over turbo codes.

Chapter 5

Partially Coupled Duo-binary Turbo Codes

In this chapter, we propose the partially coupled duo-binary turbo codes. Given a rate-$\frac{1}{3}$ turbo code as the benchmark, we construct a rate-$\frac{1}{2}$ duo-binary turbo code by introducing one extra input to the benchmark code. The coupling is performed such that a fraction of the information/parity sequence of the component code at the current time instance is used as the input of the component encoder at the next time instance. The proposed codes retain the same rates as the benchmark turbo codes, unlike the previous partially information coupled turbo codes whose code rates decrease with increasing the coupling ratio. The asymptotic decoding thresholds of the proposed codes over binary erasure channels are analyzed via density evolution. The impacts of coupling ratios on waterfall and error floor performance are studied. Simulation results show the significant error performance improvement of the proposed codes over the benchmark turbo code, our PIC-TCs and PPC-TCs introduced in Chapter 4, and other existing designs of spatially coupled turbo codes.

5.1 Introduction

In this chapter, we still focus on the partial information coupling (PIC) and partial parity coupling (PPC) techniques we have introduced in Chapter 4. Our studies have shown that both PIC-turbo codes and (PIC-TCs) and PPC-turbo codes (PPC-TCs) have comparable performance to the spatially coupled parallel concatenated codes (SC-PCCs) in [38]. However, PIC-TCs and PPC-TCs have lower code rates than their component turbo codes. Hence, some of the coded bits need to be punctured to achieve the same code rates as their component turbo codes. For example, consider a PIC-TC with a rate-$\frac{1}{3}$ turbo code with input length K as its component code. When half of the information sequences are coupled, the PIC-TC has a code rate of $\frac{1}{5}$, and each CB only carries $\frac{K}{2}$ bits information. To achieve a target code rate of $\frac{1}{3}$, half of the parity bits need to be punctured. This could lead to potential performance loss when the CB length is small or medium since the effective codeword length is reduced due to puncturing.

To overcome the aforementioned rate loss issue, and to further investigate the impacts of partial coupling on turbo codes and convolutional codes, we propose a new class of partially coupled codes, namely the partially coupled duo-binary turbo codes (dTCs). Specifically, depending on whether to couple information bits or parity bits, and whether the coupling is performed on the turbo code level or the convolutional code level, we construct partially information coupled dTCs (PIC-dTCs) [96], partially information coupled duo-binary parallel concatenated convolutional codes (PIC-dPCCs), partially parity coupled dTCs (PPC-dTCs), and partially parity coupled duo-binary parallel concatenated convolutional codes (PPC-dPCCs). The main contributions of this chapter are as follows:

- We employ a rate-$\frac{1}{2}$ dTC [74, 76] as the component code to construct PIC-dTCs, PIC-dPCCs, PPC-dTCs, and PPC-dPCCs. The rate-$\frac{1}{2}$ dTCs have one extra input compared to the benchmark rate-$\frac{1}{3}$ turbo codes. To construct our codes, one input of the component code carries the information bits at the current time instance, while the extra input carries the coupled information/parity bits from the previous time instance. Specifically, PIC-dTCs and PPC-dTCs are constructed by applying partial coupling on the turbo code level, while PIC-dPCCs and PPC-dPCCs are constructed by applying partial coupling on the convolutional code level. Our codes can retain the same rate as the benchmark turbo codes regardless of the coupling ratio. In addition, we stress that the class of partially coupled dTCs has a larger design space and more flexibility than the original PIC-TCs and PPC-TCs. Each construction within the proposed class has its unique characteristics that can be exploited to satisfy different requirements.

- The belief propagation (BP) decoding thresholds of the proposed codes over the binary erasure channel (BEC) are investigated. We first look into the graph models of the proposed code ensembles. Based on the graph representations, we respectively derive their exact DE equations and compute their decoding thresholds for different coupling ratios λ and coupling memories m. We also investigate the rate-compatible design of our codes by considering random puncturing. The coupling ratios of our codes are optimized to achieve the largest decoding thresholds for different code rates and coupling memories.

  The DE analysis shows that at code rate $R = \frac{1}{3}$, PIC-dPCCs have the best decoding threshold when coupling memory $m \leq 3$, while PIC-dTCs have the best decoding threshold with a gap to the BEC capacity of around 0.003 when $m > 3$. At $R \geq \frac{1}{2}$, PPC-dTCs outperform the rest of our codes and have a gap less than 0.005 to the BEC capacity when $m = 5$.

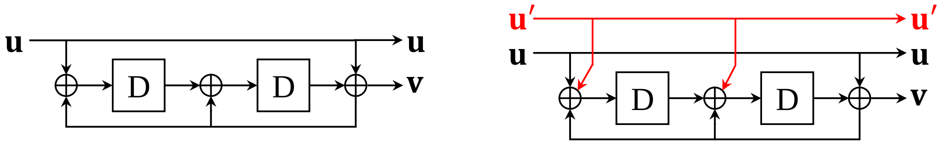
- The error floor performance of our codes is investigated. From both distance property analysis and simulation, it is observed that the error floor of our codes decreases with the increasing coupling ratio.
- The trade-off between decoding threshold and error floor performance is also studied. Specifically, PPC codes with the optimized coupling ratio have better decoding thresholds but worse error floor performance than PIC codes. Hence, we use the interpolation between PIC and PPC to achieve a better trade-off. By allowing both information bits and parity bits to be coupled simultaneously, the resultant partially hybrid coupled dTCs (PHC-dTCs) achieve a gap within 0.005 to the BEC capacity for a wide range of code rates when $m = 5$, while having a better error floor than the PPC-dTCs.
- We evaluate the error performance of our codes over the BEC and the AWGN channel via simulation. Simulation results confirm our theoretical analysis and show significant error performance improvement over existing spatially coupled turbo-like codes.

5.2 Partially Coupled Duo-binary Turbo Codes

In this section, we introduce the architecture of our partially coupled dTCs. We first review dTCs, which are the building blocks of our codes. Then, we present the encoding and decoding of our codes.

5.2.1 Construction of Duo-binary Turbo Codes

Duo-binary turbo codes refer to the turbo codes which encode two information sequences [74–76]. A rate-$\frac{1}{2}$ dTC is a parallel concatenation of two identical rate-$\frac{2}{3}$



(a) (b)

Figure 5.1: Encoder block diagram of (a) a rate-$\frac{1}{2}$ RSC code and (b) a rate-$\frac{2}{3}$ RSC code.

RSC codes, whose generator matrix is

$$\mathbf{G}_{\frac{2}{3}} = \begin{bmatrix} 1 & 0 & \frac{\mathbf{g}_f}{\mathbf{g}_b} \\ 0 & 1 & \frac{\mathbf{g}'_f}{\mathbf{g}_b} \end{bmatrix}, \tag{5.1}$$

where $\mathbf{g}_f$ and $\mathbf{g}'_f$ are the forward generator polynomials for two information sequences $\mathbf{u}$ and $\mathbf{u}'$, respectively, and $\mathbf{g}_b$ is the feedback generator polynomial.

Shortening one of the input sequences of the rate-$\frac{2}{3}$ RSC code gives a rate-$\frac{1}{2}$ RSC code. For instance, when $\mathbf{u}' = \mathbf{0}$, the rate-$\frac{2}{3}$ RSC code becomes a rate-$\frac{1}{2}$ RSC code with generator matrix $\mathbf{G}_{\frac{1}{2}} = \begin{bmatrix} 1 & \frac{\mathbf{g}_f}{\mathbf{g}_b} \end{bmatrix}$. An example is shown in Fig. 5.1, where the block diagrams of a rate-$\frac{1}{2}$ RSC code with $\mathbf{G}_{\frac{1}{2}} = [1, \frac{5}{7}]$ and a rate-$\frac{2}{3}$ RSC code with $\mathbf{G}_{\frac{2}{3}} = [1\ 0\ \frac{5}{7};\ 0\ 1\ \frac{3}{7}]$ are plotted. Consequently, a rate-$\frac{1}{3}$ turbo code can be obtained by shortening one of the input sequences of a rate-$\frac{1}{2}$ dTC.

From now on, we will use rate-$\frac{1}{2}$ dTCs to construct the partially coupled dTCs, which have a consistent code rate of $\frac{1}{3}$.

5.2.2 Notations and Definitions

Before proceeding, we introduce some frequently used notations in this Chapter. We use $\mathbf{0}_k$ to represent a k-bit all-zero sequence. Let L denote the number of CBs in a transport block (TB). For $1 \leq t \leq L$, let CB_t represent the t-th CB in the TB. The information sequence $\mathbf{u}$ of a TB is divided into L sub-sequences

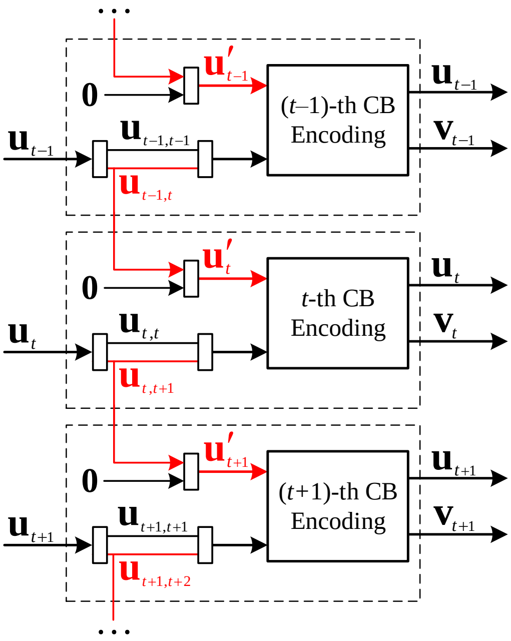


Figure 5.2: Encoder block diagram of PIC-dTCs with coupling memory $m = 1$.

$\mathbf{u}_1, \ldots, \mathbf{u}_L$ of equal length, where $\mathbf{u}_t$ is the information sequence of CB$_t$. The parity sequence of CB$_t$ is denoted by $\mathbf{v}_t = [\mathbf{v}_t^{\mathrm{U}}, \mathbf{v}_t^{\mathrm{L}}]$, where $\mathbf{v}_t^{\mathrm{U}}$ and $\mathbf{v}_t^{\mathrm{L}}$ are the parity sequences from the upper and lower RSC encoders of CB$_t$, respectively. The information length of the component code is $\|\mathbf{u}_t\| = K$, and parity length is $\|\mathbf{v}_t\| = 2K$.

5.2.3 Construction of PIC-dTCs

In this section, we introduce the architecture of PIC-dTCs. We first describe the encoding of PIC-dTCs with coupling memory $m = 1$, i.e., a CB is only coupled with two consecutive CBs. After that, we will describe the encoding of PIC-dTCs with coupling memory $m \geq 1$.

The block diagram of the PIC-dTC encoder with coupling memory $m = 1$ is depicted in Fig. 5.2, where the CB encoder refers to a rate-$\frac{1}{2}$ dTC encoder. At time t, information sequence $\mathbf{u}_t$ is decomposed into $\mathbf{u}_{t,t}$ of length $K - K_c$ and

$\mathbf{u}_{t,t+1}$ of length K_c, where $\mathbf{u}_{t,t}$ is the uncoupled information sequence which only stays in CB$_t$, and $\mathbf{u}_{t,t+1}$ is the coupled information sequence shared between CB$_t$ and CB$_{t+1}$. Then, the CB encoder takes two input sequences $\mathbf{u}_t$ and $\mathbf{u}'_t = [\mathbf{u}_{t-1,t}, \mathbf{0}_{K-K_c}]$, where $\mathbf{u}_{t-1,t}$ is the coupled information sequence of length K_c shared between CB$_{t-1}$ and CB$_t$, and $\mathbf{0}_{K-K_c}$ is the shortened bits to make sure that $\mathbf{u}_t$ and $\mathbf{u}'_t$ have equal length. Note that we set $\mathbf{u}_{0,1} = \mathbf{0}_{K_c}$ to initialize the coupling chain, and set $\mathbf{u}_{L,L+1} = \mathbf{0}_{K_c}$ to terminate the coupling chain. The codeword of CB$_t$ is obtained as $[\mathbf{u}_t, \mathbf{v}_t]$. Since $\mathbf{u}_{t-1,t}$ is a part of $\mathbf{u}_{t-1}$, it is natural to exclude $\mathbf{u}_{t-1,t}$ from the codeword of CB$_t$.

Now consider a general case of coupling memory $m \geq 1$. To initialize and terminate the coupling chain, we set $\mathbf{u}_{t-i,t} = \mathbf{0}_{\frac{K_c}{m}}$ for $t \leq i \leq m$ when $1 \leq t \leq m$, and set $\mathbf{u}_{t,t+i} = \mathbf{0}_{\frac{K_c}{m}}$ for $L-t+1 \leq i \leq m$ when $L-m+1 \leq t \leq L$. At time t, the CB encoder takes $\mathbf{u}_t$ and $\mathbf{u}'_t = [\mathbf{u}_{t-m,t}, \ldots, \mathbf{u}_{t-1,t}, \mathbf{0}_{K-K_c}]$ as inputs, where $\mathbf{u}_{t-i,t}$ is the coupled information sequence from CB$_{t-i}$, $1 \leq i \leq m$, and $\|\mathbf{u}_{t-i,t}\| = \frac{K_c}{m}$. The codeword of CB$_t$ is obtained as $[\mathbf{u}_t, \mathbf{v}_t]$, excluding $\mathbf{u}_{t-m,t}, \ldots, \mathbf{u}_{t-1,t}$.

The coupling ratio of PIC-dTCs is defined as $\lambda \triangleq \frac{K_c}{K} \in [0,1]$. The component code rate of PIC-dTCs is

$$R_0 = \frac{\|\mathbf{u}_t\| + \sum_{i=1}^{m} \|\mathbf{u}_{t-i,t}\|}{\|\mathbf{u}_t\| + \|\mathbf{v}_t\| + \sum_{i=1}^{m} \|\mathbf{u}_{t-i,t}\|} = \frac{1+\lambda}{3+\lambda}. \tag{5.2}$$

The code rate of PIC-dTCs is

$$R = \frac{\sum_{t=1}^{L} \|\mathbf{u}_t\| - K_{\text{Term}}}{\sum_{t=1}^{L} \left(\|\mathbf{u}_t\| + \|\mathbf{v}_t\|\right) - K_{\text{Term}}} = \frac{L - \frac{m+1}{2}\lambda}{3L - \frac{m+1}{2}\lambda} \overset{L\to\infty}{=} \frac{1}{3}, \tag{5.3}$$

where K_{Term} is the total number of termination bits.

Remark 5.1. We highlight the differences between PIC-TCs in [93, 97] and the proposed PIC-dTCs. PIC-TCs use a rate-$\frac{1}{3}$ turbo code as the component code

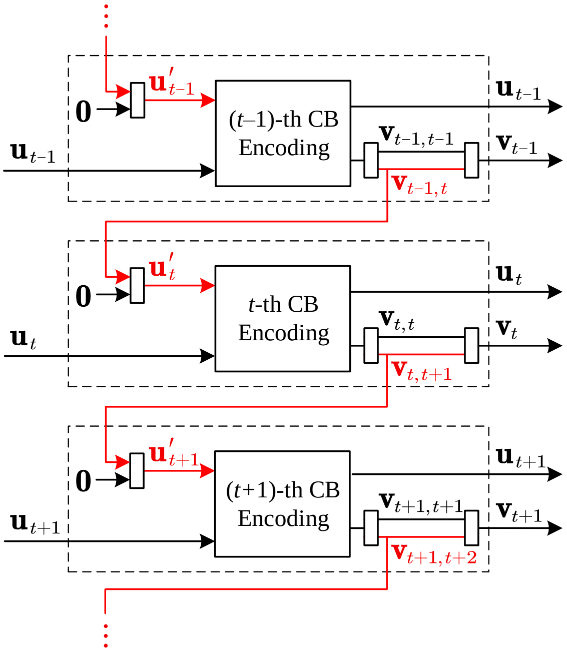


Figure 5.3: Encoder block diagram of PPC-dTCs with coupling memory $m = 1$.

so that the CB encoder only has one input sequence. For input length K and coupling ratio $\lambda \in [0, \frac{1}{2}]$, each CB effectively carries $(1-\lambda)K$ bits information, and the code rate is given by $\frac{1-\lambda}{3-\lambda}$. Therefore, there is a rate loss compared to the uncoupled turbo code, and puncturing is required to increase the code rate of PIC-TCs to $\frac{1}{3}$. In contrast, PIC-dTCs introduce an additional information sequence on top of the benchmark rate-$\frac{1}{2}$ turbo code to carry the coupled information sequences from the CBs in the previous time instances. Thus, PIC-dTCs have a constant code rate of $\frac{1}{3}$. ◀

5.2.4 Construction of PPC-dTCs

In this section, we present the construction of PPC-dTCs. As an example, the encoder block diagram of the PPC-dTCs with $m = 1$ is depicted in Fig. 5.3. For initialization, we set $\mathbf{v}_{t-i,t} = \mathbf{0}_{\frac{K_c}{m}}$ for $t \leq i \leq m$ and $1 \leq t \leq m$. For termination, we insert termination bits into $\mathbf{u}_t$ for $L - m + 1 \leq t \leq L$ so that $\mathbf{u}_t$ consists of $K - \frac{(t+m-L)K_c}{m}$ information bits and $\frac{(t+m-L)K_c}{m}$ termination bits. At time t, the CB encoder takes $\mathbf{u}_t$ and $\mathbf{u}'_t = [\mathbf{v}_{t-m,t}, \ldots, \mathbf{v}_{t-1,t},\ \mathbf{0}_{K-K_c}]$ as inputs, where $\mathbf{v}_{t-m,t}, \ldots, \mathbf{v}_{t-1,t}$ are the coupled parity sequences of length $\frac{K_c}{m}$ decomposed from $\mathbf{v}_{t-m}, \ldots, \mathbf{v}_{t-1}$, respectively. The codeword of CB_t is obtained as $[\mathbf{u}_t, \mathbf{v}_t]$. Then, $\mathbf{v}_t$ is decomposed as $\mathbf{v}_t = [\mathbf{v}_{t,t}, \mathbf{v}_{t,t+1}, \ldots, \mathbf{v}_{t,t+m}]$, where the uncoupled parity sequence $\mathbf{v}_{t,t}$ stays in CB_t, and $\mathbf{v}_{t,t+1}, \ldots, \mathbf{v}_{t,t+m}$ are passed to CB_{t+1}, …, CB_{t+m}, respectively. The coupling ratio of PPC-dTCs is defined as $\lambda \triangleq \frac{K_c}{K}$. The code rate, as well as the component code rate, of the PPC-dTCs is the same as that of the PIC-dTCs when both codes have the same coupling memory and coupling ratio.

Remark 5.2. PPC-dTCs are similar to the type-I BCCs [38] in the sense that the parity sequence of a component code at a time instance becomes a part of the input of the component encoders at the succeeding time instances. The component encoder of both schemes consists of two parallel RSC encoders. Contrary to the type-I BCCs, where the whole parity sequence is coupled and the two RSC encoders take different subsequences of it, for PPC-dTCs, only a part of the parity sequence is coupled and the two RSC encoders take the same parity subsequences. ◀

5.2.5 Construction of PIC-dPCCs and PPC-dPCCs

Apart from applying partial coupling on turbo code level to construct PIC-dTCs and PPC-dTCs, we also apply partial coupling on convolutional code level and construct PIC-dPCCs and PPC-dPCCs. To show the differences between these two coupling levels, we plot their CB encoder block diagrams in Fig. 5.4. As shown in Fig. 5.4a, coupling on turbo code level refers to that the upper and lower RSC encoders of CB_t take the same input sequences. In specific, both RSC encoders take $\mathbf{u}_t$ and $\mathbf{u}'_t = [\mathbf{u}_{t-m,t}, \ldots, \mathbf{u}_{t-1,t}, \mathbf{0}_{K-K_c}]$ for encoding PIC-dTCs, and take $\mathbf{u}_t$ and $\mathbf{u}'_t = [\mathbf{v}_{t-m,t}, \ldots, \mathbf{v}_{t-1,t}, \mathbf{0}_{K-K_c}]$ for encoding PPC-dTCs. On contrary, as shown in Fig. 5.4b, coupling on convolutional code level refers to that the coupled information or parity sequences are de-multiplexed to the upper and lower RSC encoders. Therefore, the upper and lower RSC encoders of CB_t take different input sequences. In the following, we present the encoding of PIC-dPCCs and PPC-dPCCs respectively.

We first introduce the construction of PIC-dPCCs. At time t, the CB encoder obtains $\mathbf{u}_t$ and $\mathbf{u}_{t-m,t}, \ldots, \mathbf{u}_{t-1,t}$. For $1 \leq i \leq m$, $\mathbf{u}_{t-i,t}$ is decomposed into two sub-sequences $\mathbf{u}^{\text{U}}_{t-i,t}$ and $\mathbf{u}^{\text{L}}_{t-i,t}$. The upper RSC encoder of CB_t takes $\mathbf{u}_t$ and $\mathbf{u}^{\text{U}}_t = [\mathbf{u}^{\text{U}}_{t-m,t}, \ldots, \mathbf{u}^{\text{U}}_{t-1,t}, \mathbf{0}_{K-\frac{K_c}{2}}]$, while the lower RSC encoder of CB_t takes $\mathbf{u}_t$ and $\mathbf{u}^{\text{L}}_t = [\mathbf{u}^{\text{L}}_{t-m,t}, \ldots, \mathbf{u}^{\text{L}}_{t-1,t}, \mathbf{0}_{K-\frac{K_c}{2}}]$, respectively. The codeword of CB_t is $[\mathbf{u}_t, \mathbf{v}_t]$. The length of the upper and lower coupled sequences is $K^{\text{U}}_c = \sum_{i=1}^{m} \|\mathbf{u}^{\text{U}}_{t-i,t}\|$ and $K^{\text{L}}_c = \sum_{i=1}^{m} \|\mathbf{u}^{\text{L}}_{t-i,t}\|$, respectively. We define the coupling ratio of PIC-dPCCs as $\lambda \triangleq \frac{K_c}{K} = \lambda_{\text{U}} + \lambda_{\text{L}}$, where $\lambda_{\text{U}} = \frac{K^{\text{U}}_c}{K}$ and $\lambda_{\text{L}} = \frac{K^{\text{L}}_c}{K}$ are the upper and lower coupling ratios, respectively. For simplicity, we consider that $\lambda_{\text{U}} = \lambda_{\text{L}} = \frac{\lambda}{2}$. The component code rate is

$$R_0 = \frac{1 + \frac{\lambda}{2}}{3 + \frac{\lambda}{2}} = \frac{2 + \lambda}{6 + \lambda}. \tag{5.4}$$

The termination procedure for PIC-dPCCs is the same as for PIC-dTCs. Thus, the code rate expression of the PIC-dPCCs is the same as in (5.3).

Now we introduce the construction of PPC-dPCCs. At time t, the CB encoder obtains $\mathbf{u}_t$ and $\mathbf{v}_{t-m,t}, \ldots, \mathbf{v}_{t-1,t}$. For $1 \leq i \leq m$, $\mathbf{v}_{t-i,t}$ is interleaved and then decomposed into $\widetilde{\mathbf{v}}^{\text{U}}_{t-i,t}$ and $\widetilde{\mathbf{v}}^{\text{L}}_{t-i,t}$. The upper RSC encoder of CB_t takes $\mathbf{u}_t$ and $\mathbf{u}^{\text{U}}_t = [\widetilde{\mathbf{v}}^{\text{U}}_{t-m,t}, \ldots, \widetilde{\mathbf{v}}^{\text{U}}_{t-1,t}, \mathbf{0}_{K-\frac{K_c}{2}}]$ as inputs, while the lower RSC encoder CB_t takes $\mathbf{u}_t$ and $\mathbf{u}^{\text{L}}_t = [\widetilde{\mathbf{v}}^{\text{L}}_{t-m,t}, \ldots, \widetilde{\mathbf{v}}^{\text{L}}_{t-1,t}, \mathbf{0}_{K-\frac{K_c}{2}}]$ as inputs, respectively.

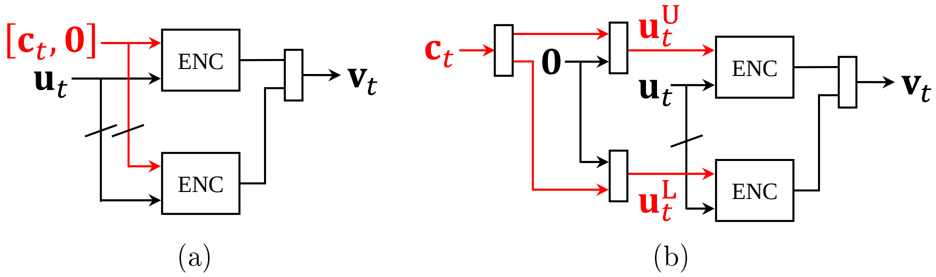


Figure 5.4: CB encoder block diagram of (a) PIC-dTCs and PPC-dTCs, and (b) PIC-dPCCs and PPC-dPCCs, where $\mathbf{c}_t$ represents coupled information/parity sequences. "ENC" represents the RSC encoder, and the slash on the edge represents the interleaver.

The codeword of CB$_t$ is $[\mathbf{u}_t, \mathbf{v}_t]$. The length of the coupled sequences are $K_c^{\mathrm{U}} = \sum_{i=1}^{m} \|\tilde{\mathbf{v}}_{t-i,t}^{\mathrm{U}}\|$ and $K_c^{\mathrm{L}} = \sum_{i=1}^{m} \|\tilde{\mathbf{v}}_{t-i,t}^{\mathrm{L}}\|$, respectively. We define the coupling ratio[1] as $\lambda \triangleq \frac{K_c}{2K} = \frac{\lambda_{\mathrm{U}}+\lambda_{\mathrm{L}}}{2}$, where $\lambda_{\mathrm{U}} \triangleq \frac{K_c^{\mathrm{U}}}{K}$, and $\lambda_{\mathrm{L}} \triangleq \frac{K_c^{\mathrm{L}}}{K}$. For simplicity, we consider that $\lambda_{\mathrm{U}} = \lambda_{\mathrm{L}} = \lambda$ such that $R_0 = \frac{1+\lambda}{3+\lambda}$. The termination procedure for PPC-dPCCs is the same as for PPC-dTCs. Thus, the code rate expression of PPC-dPCCs is the same as (5.3).

Remark 5.3. For constructing PPC-dPCCs, the selection of $\tilde{\mathbf{v}}_{t-i,t}^{\mathrm{U}}$ and $\tilde{\mathbf{v}}_{t-i,t}^{\mathrm{L}}$ can be implemented in two ways. In addition to the one described above, where $\mathbf{v}_{t-i,t}$ is randomly decomposed, another way is to choose $\tilde{\mathbf{v}}_{t-i,t}^{\mathrm{U}} = \mathbf{v}_{t-i,t}^{\mathrm{L}}$ and $\tilde{\mathbf{v}}_{t-i,t}^{\mathrm{L}} = \mathbf{v}_{t-i,t}^{\mathrm{U}}$, such that the upper parity sequence of CB$_{t-i}$ becomes a part of the input of the lower RSC encoder of CB$_t$, while the lower parity sequence of CB$_{t-i}$ becomes a part of the input of the upper RSC encoder of CB$_t$. As we use identical generator for the upper and lower RSC encoders, these two methods have the same decoding performance. In this work, we consider the second implementation. Note that type-I BCCs can be seen as a special case of PIC-dPCCs with $\lambda = 1$ and using the second implementation. ◀

---

[1] For PIC-dTCs, PIC-dPCCs, and PPC-dTCs, the coupling ratio is $\frac{K_c}{K}$ as they can couple at most K information/parity bits. In contrast, the PPC-dPCCs can couple at most $2K$ parity bits. Hence, the coupling ratio is $\frac{K_c}{2K}$.

Table 5.1: Structural characteristics of various spatially coupled turbo-like codes.

Ensemble	Component code	Coupling level	Coupled bits	Node degree regularity
PIC-TC	Turbo code	Turbo code	Info.	Regular/irregular
PPC-TC	Turbo code	Turbo code	Parity	Irregular
PIC-dTC	dTC	Turbo code	Info.	Regular/irregular
PPC-dTC	dTC	Turbo code	Parity	Irregular
PIC-dPCC	dTC	RSC code	Info.	Regular/irregular
PPC-dPCC	dTC	RSC code	Parity	Regular/irregular
SC-SCC	SCC	RSC code	Info. & parity	Regular
SC-PCC	Turbo code	RSC code	Info.	Regular
Type-I BCC	dTC	RSC code	Parity	Regular
Type-II BCC	dTC	RSC code	Info. & parity	Regular

5.2.6 Comparison with Existing Codes

In Table. 5.1, we compare the structural characteristics of our codes with other benchmark coding schemes, including PIC-TCs and PPC-TCs, SC-PCCs, SC-SCCs, and BCCs. The most important property of the partially coupled codes is the irregularity of variable node degrees. When PIC is employed, the uncoupled information bits are protected by two RSC codes. For PIC-TCs and PIC-dTCs, the coupled information bits are protected by four RSC codes. For PIC-dPCCs, the coupled information bits are protected by three RSC codes. For PPC codes, the uncoupled parity bits are only protected by one RSC code, while the coupled parity bits are protected by one (for PPC-dPCCs) or two (for PPC-TCs and PPC-dTCs) RSC codes.

In contrast, other SC-TCs have regular node degree. For SC-PCCs, although a part of the information bits is coupled, all the information bits are protected by two RSC codes so that they have the same node degree. For SC-SCCs, where a part of the information bits and a part of the outer parity bits are coupled, all information bits have the same node degree, and all outer parity bits have the same node degree. For type-I and type-II BCCs, all parity bits have the same node degree, while all information bits have the same node degree.

Note that for PIC codes, their node degree becomes regular when $\lambda = 1$. This is because all information bits are coupled and consequently have regular node degree. Similarly, PPC-dPCCs with $\lambda = 1$ have regular node degree as all parity bits are coupled. Meanwhile, PPC-TCs and PPC-dTCs are always irregular even with the largest coupling ratio, because at most half of the parity bits can be coupled.

5.2.7 Decoding Scheme

The decoding of PIC codes and PPC codes can be accomplished by a feed-forward and feed-back (FF-FB) scheme [93, 97]. In summary, the FF-FB scheme includes two types of decoding iterations. One is the inner iteration of the CB decoder, where the upper and lower RSC decoders exchange extrinsic information between them, realized by the BCJR algorithm [47, 88]. The other is the outer iteration, in which the extrinsic information of the coupled sequences is passed between consecutive CBs in forward-backward round trips. Let I_{CB} and I_{TB} represent the maximum numbers of the inner and outer iterations, respectively. To decode CB_t, the decoder takes the received signal associated with the component codeword $[\mathbf{u}_t, \mathbf{v}_t]$ as well as the extrinsic information associated with the coupled sequences. Then, it outputs the estimation of $\mathbf{u}_t$ and $\mathbf{v}_t$, and updates the extrinsic information of the coupled sequences. Specifically, for PIC codes, the extrinsic information of $\mathbf{u}_{t-i,t}$ and $\mathbf{u}_{t,t+i}$ for $1 \leq i \leq m$ is updated during CB decoding. For PPC codes, the extrinsic information of $\mathbf{v}_{t-i,t}$ and $\mathbf{v}_{t,t+i}$ for $1 \leq i \leq m$ is updated during CB decoding. The update rules for coupled information sequences and coupled parity sequences are the same with that of PIC-TCs and PPC-TCs presented in Chapter 4, so we omit the details here.

5.3 Density Evolution Analysis

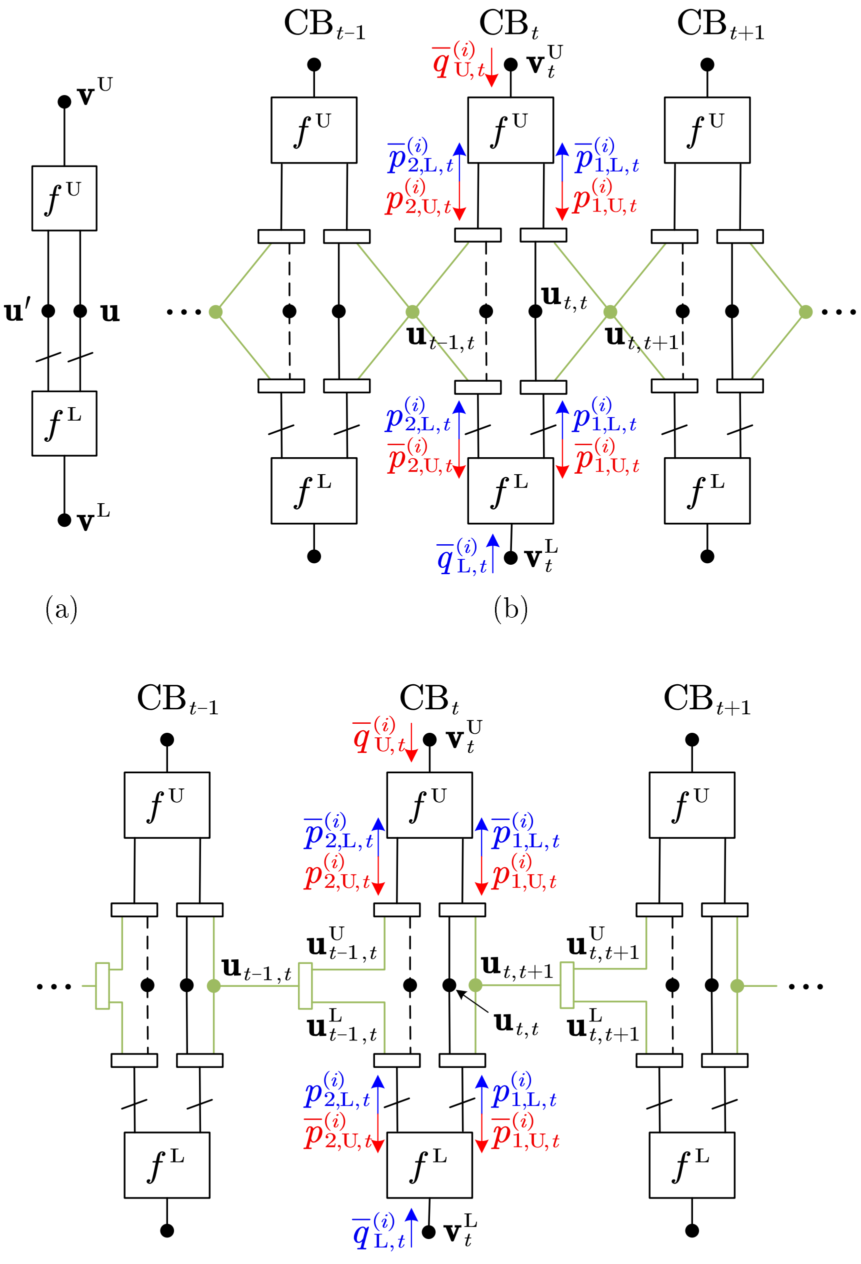
In this section, we present the DE analysis of the partially coupled dTCs over the BEC. We first present the graph models of all proposed code ensembles and then introduce the relevant notations. Next, we derive the DE equations. Coupling memory $m = 1$ is considered in this section. The derivation of DE equations for $m \geq 1$ is deferred to Appendix 5.A.

5.3.1 Graph Model Representation

We start with the graph model of the uncoupled dTCs. Then, we look into the graph models of our partially coupled dTCs, respectively.

Fig. 5.5a shows the compact graph representation of the uncoupled dTCs. We use two factor nodes f^{U} and f^{L} to represent the upper and lower RSC decoders, respectively. The two information sequences $\mathbf{u}$ and $\mathbf{u}'$ are represented by two different variable nodes connected to the upper and lower factor nodes, meaning that the extrinsic information is passed between the upper and lower decoders via the corresponding variable nodes. The two variable nodes, which only connect to one of the factor nodes, represent the parity sequences $\mathbf{v}^{\mathrm{U}}$ and $\mathbf{v}^{\mathrm{L}}$, respectively. An interleaver is represented by a cross on the edge between the variable nodes and the lower factor node.

Fig. 5.5b shows the compact graph of PIC-dTCs. We use a node connecting to dashed lines to represent the shortened bits. At time t, we decompose the information node $\mathbf{u}_t$ into the uncoupled node $\mathbf{u}_{t,t}$, the coupled node $\mathbf{u}_{t,t+1}$. The coupled node $\mathbf{u}_{t-1,t}$ is connected to both $\mathbf{u}_{t-1}$ and $\mathbf{u}'_t$, meaning that $\mathbf{u}_{t-1,t}$ is a


$\mathbf{v}^{\mathrm{U}}$
f^{U}
$\mathbf{u}'$
$\mathbf{u}$
f^{L}
$\mathbf{v}^{\mathrm{L}}$
$\mathrm{CB}_{t\text{-}1}$
CB_{t}
CB_{t+1}
$\overline{q}^{(i)}_{\mathrm{U},t}$
$\mathbf{v}^{\mathrm{U}}_{t}$
$\overline{p}^{(i)}_{2,\mathrm{L},t}$
$p^{(i)}_{2,\mathrm{U},t}$
$\overline{p}^{(i)}_{1,\mathrm{L},t}$
$p^{(i)}_{1,\mathrm{U},t}$
$\mathbf{u}_{t,t}$
$\mathbf{u}_{t\text{-}1,t}$
$\mathbf{u}_{t,t+1}$
$p^{(i)}_{2,\mathrm{L},t}$
$\overline{p}^{(i)}_{2,\mathrm{U},t}$
$p^{(i)}_{1,\mathrm{L},t}$
$\overline{p}^{(i)}_{1,\mathrm{U},t}$
$\overline{q}^{(i)}_{\mathrm{L},t}$
$\mathbf{v}^{\mathrm{L}}_{t}$
$\mathbf{u}^{\mathrm{U}}_{t\text{-}1,t}$
$\mathbf{u}^{\mathrm{L}}_{t\text{-}1,t}$
$\mathbf{u}^{\mathrm{U}}_{t,t+1}$
$\mathbf{u}^{\mathrm{L}}_{t,t+1}$
$\cdots$

(a) (b)

(c)

Figure 5.5: Compact graph of (a) uncoupled dTCs, (b) PIC-dTCs, and (c) PIC-dPCCs. Coupling memory $m = 1$.

part of the decoder inputs at both $t-1$ and time t. Due to the similar reason, the coupled node $\mathbf{u}_{t,t+1}$ is connected to both $\mathbf{u}_t$ and $\mathbf{u}'_{t+1}$.

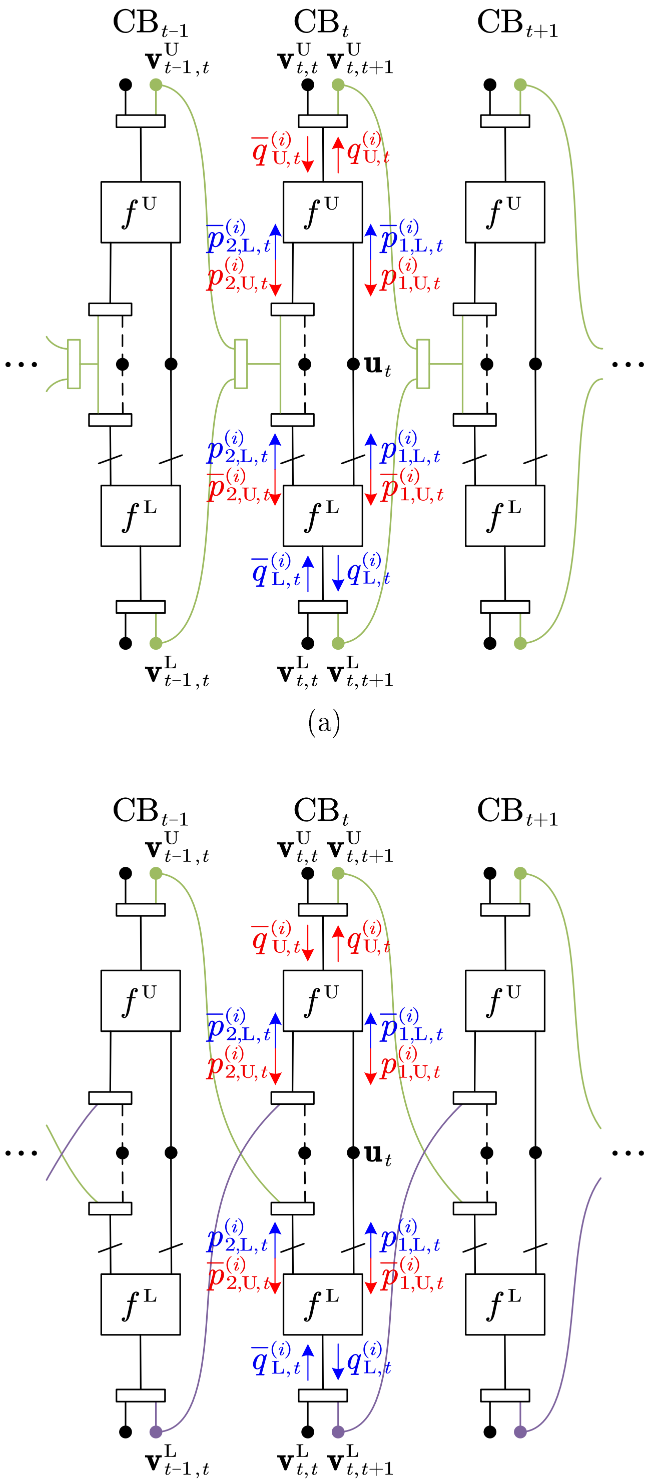
Fig. 5.5c shows the compact graph of PIC-dPCCs. At time t, we use $\mathbf{u}_t^{\mathrm{U}}$ and $\mathbf{u}_t^{\mathrm{L}}$ to represent the second information sequence entering the upper and lower decoders, respectively. Variable node $\mathbf{u}_{t,t+1}$ is decomposed into $\mathbf{u}_{t,t+1}^{\mathrm{U}}$ and $\mathbf{u}_{t,t+1}^{\mathrm{L}}$, connecting to $\mathbf{u}_{t+1}^{\mathrm{U}}$ and $\mathbf{u}_{t+1}^{\mathrm{L}}$, respectively.

Fig. 5.6a shows the compact graph of PPC-dTCs. At time t, we decompose the parity nodes $\mathbf{v}_t^{\mathrm{U}}$ and $\mathbf{v}_t^{\mathrm{L}}$ into the coupled nodes $\mathbf{v}_{t,t+1}^{\mathrm{U}}$ and $\mathbf{v}_{t,t+1}^{\mathrm{L}}$, and the uncoupled nodes $\mathbf{v}_{t,t}^{\mathrm{U}}$ and $\mathbf{v}_{t,t}^{\mathrm{L}}$, respectively. Then, the coupled nodes are connected to $\mathbf{u}'_{t+1}$, meaning that their extrinsic information is a part of the decoder input at time $t+1$.

Fig. 5.6b shows the compact graph of PPC-dPCCs. At time t, the coupled node $\mathbf{v}_{t,t+1}^{\mathrm{U}}$ is connected to $\mathbf{u}_{t+1}^{\mathrm{L}}$, meaning that its extrinsic information is a part of the lower decoder input at time $t+1$. The coupled node $\mathbf{v}_{t,t+1}^{\mathrm{L}}$ is connected to $\mathbf{u}_{t+1}^{\mathrm{U}}$, meaning that its extrinsic information is a part of the upper decoder input at time $t+1$.

5.3.2 Notations and Definitions

Now we introduce some notations before we analyze our codes in the following subsections. Let ε be the BEC erasure probability. The input-output relations of f^{U} are described by the transfer functions: F_1^{U}, F_2^{U}, and F_3^{U}, corresponding to the two information sequences and the parity sequence, respectively. Likewise, the input-output relations of f^{L} are described by the transfer functions F_1^{L}, F_2^{L}, and F_3^{L}, respectively. The transfer functions are derived following [82]. At time


CB_{t-1}
CB_{t}
CB_{t+1}
$\mathbf{v}^{\mathrm{U}}_{t-1,t}$
$\mathbf{v}^{\mathrm{U}}_{t,t}$
$\mathbf{v}^{\mathrm{U}}_{t,t+1}$
$\bar{q}^{(i)}_{\mathrm{U},t}$
$q^{(i)}_{\mathrm{U},t}$
f^{U}
$\bar{p}^{(i)}_{2,\mathrm{L},t}$
$p^{(i)}_{2,\mathrm{U},t}$
$\bar{p}^{(i)}_{1,\mathrm{L},t}$
$p^{(i)}_{1,\mathrm{U},t}$
$\cdots$
$\mathbf{u}_t$
$\cdots$
$p^{(i)}_{2,\mathrm{L},t}$
$\bar{p}^{(i)}_{2,\mathrm{U},t}$
$p^{(i)}_{1,\mathrm{L},t}$
$\bar{p}^{(i)}_{1,\mathrm{U},t}$
f^{L}
$\bar{q}^{(i)}_{\mathrm{L},t}$
$q^{(i)}_{\mathrm{L},t}$
$\mathbf{v}^{\mathrm{L}}_{t-1,t}$
$\mathbf{v}^{\mathrm{L}}_{t,t}$
$\mathbf{v}^{\mathrm{L}}_{t,t+1}$

(a)

(b)

Figure 5.6: Compact graph of (a) PPC-dTCs, and (b) PPC-dPCCs. Coupling memory $m = 1$.

Table 5.2: Notations for the DE analysis of PIC-dTCs and PPC-dTCs.

Notation	Definition
$\bar{p}_{1,\mathrm{L},t}^{(i)}$	Information input ($\mathbf{u}_t \to f^{\mathrm{U}}$) erasure probability of f^{U}
$\bar{p}_{2,\mathrm{L},t}^{(i)}$	Information input ($\mathbf{u}'_t \to f^{\mathrm{U}}$) erasure probability of f^{U}
$\bar{q}_{\mathrm{U},t}^{(i)}$	Parity input erasure probability of f^{U}
$p_{1,\mathrm{U},t}^{(i)}$	Information output ($f^{\mathrm{U}} \to \mathbf{u}_t$) erasure probability of f^{U}
$p_{2,\mathrm{U},t}^{(i)}$	Information output ($f^{\mathrm{U}} \to \mathbf{u}'_t$) erasure probability of f^{U}
$q_{\mathrm{U},t}^{(i)}$	Parity output erasure probability of f^{U}
$\bar{p}_{1,\mathrm{U},t}^{(i)}$	Information input ($\mathbf{u}_t \to f^{\mathrm{L}}$) erasure probability of f^{L}
$\bar{p}_{2,\mathrm{U},t}^{(i)}$	Information input ($\mathbf{u}'_t \to f^{\mathrm{L}}$) erasure probability of f^{L}
$\bar{q}_{\mathrm{L},t}^{(i)}$	Parity input erasure probability of f^{L}
$p_{1,\mathrm{L},t}^{(i)}$	Information output ($f^{\mathrm{L}} \to \mathbf{u}_t$) erasure probability of f^{L}
$p_{2,\mathrm{L},t}^{(i)}$	Information output ($f^{\mathrm{L}} \to \mathbf{u}'_t$) erasure probability of f^{L}
$q_{\mathrm{L},t}^{(i)}$	Parity output erasure probability of f^{L}

t and the i-th iteration, the DE equations of f^{U} and f^{L} are

$$f^{\mathrm{U}}:\begin{cases} p_{1,\mathrm{U},t}^{(i)} = F_1^{\mathrm{U}}\left(\bar{p}_{1,\mathrm{L},t}^{(i)}, \bar{p}_{2,\mathrm{L},t}^{(i)}, \bar{q}_{\mathrm{U},t}^{(i)}\right) \\ p_{2,\mathrm{U},t}^{(i)} = F_2^{\mathrm{U}}\left(\bar{p}_{1,\mathrm{L},t}^{(i)}, \bar{p}_{2,\mathrm{L},t}^{(i)}, \bar{q}_{\mathrm{U},t}^{(i)}\right) \\ q_{\mathrm{U},t}^{(i)} = F_3^{\mathrm{U}}\left(\bar{p}_{1,\mathrm{L},t}^{(i)}, \bar{p}_{2,\mathrm{L},t}^{(i)}, \bar{q}_{\mathrm{U},t}^{(i)}\right) \end{cases}, \tag{5.5a}$$

$$f^{\mathrm{L}}:\begin{cases} p_{1,\mathrm{L},t}^{(i)} = F_1^{\mathrm{L}}\left(\bar{p}_{1,\mathrm{U},t}^{(i)}, \bar{p}_{2,\mathrm{U},t}^{(i)}, \bar{q}_{\mathrm{L},t}^{(i)}\right) \\ p_{2,\mathrm{L},t}^{(i)} = F_2^{\mathrm{L}}\left(\bar{p}_{1,\mathrm{U},t}^{(i)}, \bar{p}_{2,\mathrm{U},t}^{(i)}, \bar{q}_{\mathrm{L},t}^{(i)}\right) \\ q_{\mathrm{L},t}^{(i)} = F_3^{\mathrm{L}}\left(\bar{p}_{1,\mathrm{U},t}^{(i)}, \bar{p}_{2,\mathrm{U},t}^{(i)}, \bar{q}_{\mathrm{L},t}^{(i)}\right) \end{cases}. \tag{5.5b}$$

The notations for the inputs and outputs of the above transfer functions can be found in Table 5.2.

With (5), it then remains to derive the input erasure probabilities for each proposed code at f^{U} and f^{L}. As the DE analysis for f^{L} is derived similarly as that for f^{U}, we only present the DE update at f^{U} in the following subsection.

5.3.3 Density Evolution Analysis

PIC-dTCs

We first consider PIC-dTCs with coupling memory $m = 1$. For f^{U} at time t and the i-th iteration, $\bar{p}^{(i)}_{1,\mathrm{L},t}$ depends on the erasure probabilities of $\mathbf{u}_{t,t}$ and $\mathbf{u}_{t,t+1}$, while $\bar{p}^{(i)}_{2,\mathrm{L},t}$ only depends on the erasure probability of $\mathbf{u}_{t-1,t}$ because the erasure probability of shortened bits is 0. Both $\mathbf{u}_{t-1,t}$, $\mathbf{u}_{t,t}$, and $\mathbf{u}_{t,t+1}$ can obtain extrinsic information from f^{L} at time t. In addition, $\mathbf{u}_{t-1,t}$ can further obtain extrinsic information from f^{L} and f^{U} at time $t-1$, while $\mathbf{u}_{t,t+1}$ can further obtain extrinsic information from f^{L} and f^{U} at time $t+1$. Thus, the average erasure probabilities from $\mathbf{u}_t$, $\mathbf{u}'_t$, and $\mathbf{v}^{\mathrm{U}}_t$ to f^{U} are computed as

$$\bar{p}^{(i)}_{1,\mathrm{L},t} = \varepsilon \cdot p^{(i)}_{1,\mathrm{L},t} \cdot \left(1 - \lambda + \lambda \cdot p^{(i-1)}_{2,\mathrm{U},t+1} \cdot p^{(i)}_{2,\mathrm{L},t+1}\right), \tag{5.6a}$$

$$\bar{p}^{(i)}_{2,\mathrm{L},t} = \varepsilon \cdot \lambda \cdot p^{(i)}_{2,\mathrm{L},t} \cdot p^{(i-1)}_{1,\mathrm{U},t-1} \cdot p^{(i)}_{1,\mathrm{L},t-1}, \tag{5.6b}$$

$$\bar{q}^{(i)}_{\mathrm{U},t} = \varepsilon. \tag{5.6c}$$

As no extrinsic information is passed to the parity nodes, the erasure probability of $\mathbf{v}^{\mathrm{U}}_t$ is always the same as in (5.6c) for PIC codes. So we do not show $\mathbf{v}^{\mathrm{U}}_t$ repeatedly in the rest of the DE analysis for PIC codes.

With the knowledge of the input erasure probabilities, the evolution of the erasure probability for $\mathbf{u}_t$ and $\mathbf{u}'_t$ inside f^{U} can be obtained with transfer functions (5.5). After computing the evolution of the erasure probability for $\mathbf{u}_t$ and $\mathbf{u}'_t$, the a-posteriori erasure probability of $\mathbf{u}_t$ after i time of iterations is

$$p^{(i)}_{\mathbf{u}_t} = \varepsilon \cdot p^{(i)}_{1,\mathrm{U},t} \cdot p^{(i)}_{1,\mathrm{L},t} \cdot \left(1 - \lambda + \lambda \cdot p^{(i)}_{2,\mathrm{U},t+1} \cdot p^{(i)}_{2,\mathrm{L},t+1}\right). \tag{5.7}$$

The DE analysis for PIC-dTCs with $m \geq 1$ is given in Appendix 5.A.1.

PIC-dPCCs

We now consider PIC-dPCCs with coupling memory $m = 1$. For f^{U} at time t and the i-th iteration, $\bar{p}_{1,\mathrm{L},t}^{(i)}$ depends on the erasure probabilities of $\mathbf{u}_{t,t}$, $\mathbf{u}_{t,t+1}^{\mathrm{U}}$ and $\mathbf{u}_{t,t+1}^{\mathrm{L}}$, while $\bar{p}_{2,\mathrm{L},t}^{(i)}$ depends on the erasure probability of $\mathbf{u}_{t-1,t}^{\mathrm{U}}$. Specifically, $\mathbf{u}_{t,t+1}^{\mathrm{U}}$ can obtain extrinsic information from f^{L} at time t and f^{U} at time $t+1$, $\mathbf{u}_{t,t+1}^{\mathrm{L}}$ can obtain extrinsic information from f^{L} at time t and f^{L} at time $t+1$, and $\mathbf{u}_{t-1,t}^{\mathrm{U}}$ can obtain extrinsic information from f^{U} and f^{L} at time $t-1$. Thus, the average erasure probabilities from $\mathbf{u}_t$ and $\mathbf{u}_t'$ to f^{U} are computed as

$$\bar{p}_{1,\mathrm{L},t}^{(i)} = \varepsilon \cdot p_{1,\mathrm{L},t}^{(i)} \cdot \left(1 - \lambda + \frac{\lambda}{2} \cdot \left(p_{2,\mathrm{U},t+1}^{(i-1)} + p_{2,\mathrm{L},t+1}^{(i)}\right)\right), \tag{5.8a}$$

$$\bar{p}_{2,\mathrm{L},t}^{(i)} = \varepsilon \cdot \frac{\lambda}{2} \cdot p_{1,\mathrm{U},t-1}^{(i-1)} \cdot p_{1,\mathrm{L},t-1}^{(i)}. \tag{5.8b}$$

The a-posteriori erasure probability of $\mathbf{u}_t$ after i iterations is

$$p_{\mathbf{u}_t}^{(i)} = \varepsilon \cdot p_{1,\mathrm{U},t}^{(i)} \cdot p_{1,\mathrm{L},t}^{(i)} \cdot \left(1 - \lambda + \frac{\lambda}{2} \cdot \left(p_{2,\mathrm{U},t+1}^{(i)} + p_{2,\mathrm{L},t+1}^{(i)}\right)\right). \tag{5.9}$$

The DE analysis for PIC-dPCCs with $m \geq 1$ is given in Appendix 5.A.2.

PPC-dTCs

Consider PPC-dTCs with coupling memory $m = 1$. For f^{U} at time t and the i-th iteration, $\bar{p}_{1,\mathrm{L},t}^{(i)}$ only depends on the erasure probability of $\mathbf{u}_t$, and $\bar{p}_{2,\mathrm{L},t}^{(i)}$ depends on the erasure probabilities of $\mathbf{v}_{t-1,t}^{\mathrm{U}}$ and $\mathbf{v}_{t-1,t}^{\mathrm{L}}$, where $\mathbf{v}_{t-1,t}^{\mathrm{U}}$ can obtain extrinsic information from f^{U} at time $t-1$ and f^{L} at time t, and $\mathbf{v}_{t-1,t}^{\mathrm{L}}$ can obtain extrinsic information from f^{L} at time $t-1$ and f^{L} at time t. For the parity node $\mathbf{v}_t^{\mathrm{U}}$, $\bar{q}_{\mathrm{U},t}^{(i)}$ depends on the erasure probabilities of $\mathbf{v}_{t,t}^{\mathrm{U}}$ and $\mathbf{v}_{t,t+1}^{\mathrm{U}}$, where $\mathbf{v}_{t,t+1}^{\mathrm{U}}$ can obtain extrinsic information from both f^{U} and f^{L} at time $t+1$. The average erasure probabilities from $\mathbf{u}_t$, $\mathbf{u}'_t$, and $\mathbf{v}_t^{\mathrm{U}}$ to f^{U} are computed as

$$\bar{p}_{1,\mathrm{L},t}^{(i)} = \varepsilon \cdot p_{1,\mathrm{L},t}^{(i)}, \tag{5.10a}$$

$$\bar{p}_{2,\mathrm{L},t}^{(i)} = \varepsilon \cdot \frac{\lambda}{2} \cdot p_{2,\mathrm{L},t}^{(i)} \cdot \left(q_{\mathrm{U},t-1}^{(i-1)} + q_{\mathrm{L},t-1}^{(i)}\right), \tag{5.10b}$$

$$\bar{q}_{\mathrm{U},t}^{(i)} = \varepsilon \cdot \left(1 - \frac{\lambda}{2} + \frac{\lambda}{2} \cdot p_{1,\mathrm{U},t+1}^{(i-1)} \cdot p_{1,\mathrm{L},t+1}^{(i)}\right). \tag{5.10c}$$

The a-posteriori erasure probability of $\mathbf{u}_t$ is computed as

$$p_{\mathbf{u}_t}^{(i)} = \varepsilon \cdot p_{1,\mathrm{U},t}^{(i)} \cdot p_{1,\mathrm{L},t}^{(i)}. \tag{5.11}$$

The DE analysis for PPC-dTCs with $m \geq 1$ is given in Appendix 5.A.3. Note that for both PPC-dTCs and PPC-dPCCs, as $\mathbf{u}_t$ does not share information with other nodes, $\bar{p}_{1,\mathrm{L},t}^{(i)}$ is always the same as in (5.10a) regardless the coupling memory. So we do not show $\bar{p}_{1,\mathrm{L},t}^{(i)}$ repeatedly in the rest of the DE analysis for PPC codes.

PPC-dPCCs

Lastly, we consider PPC-dPCCs with coupling memory $m = 1$. For f^{U} at time t and the i-th iteration, $\bar{p}_{2,\mathrm{L},t}^{(i)}$ depends on the erasure probability of $\mathbf{v}_{t-1,t}^{\mathrm{L}}$, while $\bar{q}_{\mathrm{U},t}^{(i)}$ depends on the erasure probabilities of $\mathbf{v}_{t,t}^{\mathrm{U}}$ and $\mathbf{v}_{t,t+1}^{\mathrm{U}}$, where $\mathbf{v}_{t,t+1}^{\mathrm{U}}$ can obtain extrinsic information from f^{L} at time $t+1$. The average erasure probabilities from $\mathbf{u}'_t$ and $\mathbf{v}_t^{\mathrm{U}}$ to f^{U} are computed as

$$\bar{p}_{2,\mathrm{L},t}^{(i)} = \varepsilon \cdot \lambda \cdot q_{\mathrm{L},t-1}^{(i)}, \tag{5.12a}$$

$$\bar{q}_{\mathrm{U},t}^{(i)} = \varepsilon \cdot \left(1 - \lambda + \lambda \cdot p_{1,\mathrm{L},t+1}^{(i)}\right). \tag{5.12b}$$

The a-posteriori erasure probability of $\mathbf{u}_t$ is computed with the same expression in (5.11). The DE analysis for PPC-dPCCs with $m \geq 1$ is given in Appendix

5.A.4.

5.3.4 Density Evolution with Random Puncturing

In this subsection, we consider rate-compatible designs for the proposed codes by randomly puncturing the parity bits. Let δ represent the puncturing ratio, which is the fraction of punctured bits in the parity sequence of a CB. The code rate of a punctured partially coupled dTC with coupling ratio λ, coupling memory m, and puncturing ratio δ is

$$R = \frac{L - \frac{m+1}{2}\lambda}{L(3-2\delta) - \frac{m+1}{2}\lambda} \overset{L\to\infty}{=} \frac{1}{3-2\delta}. \tag{5.13}$$

On the other hand, we can obtain the puncturing ratio when given a target code rate R as

$$\delta \overset{L\to\infty}{=} 1 - \frac{1-R}{2R}. \tag{5.14}$$

Given BEC erasure probability ε, the parity erasure probability becomes $\varepsilon' = \varepsilon \cdot (1-\delta)+\delta$. For the DE analysis of PIC codes, we simply substitute $\bar{q}^{(i)}_{\mathrm{U},t} = \bar{q}^{(i)}_{\mathrm{L},t} = \varepsilon'$. For the DE analysis of PPC codes, we substitute ε with ε' in (5.10b), (5.10c), and (5.12).

5.4 BP Decoding Thresholds

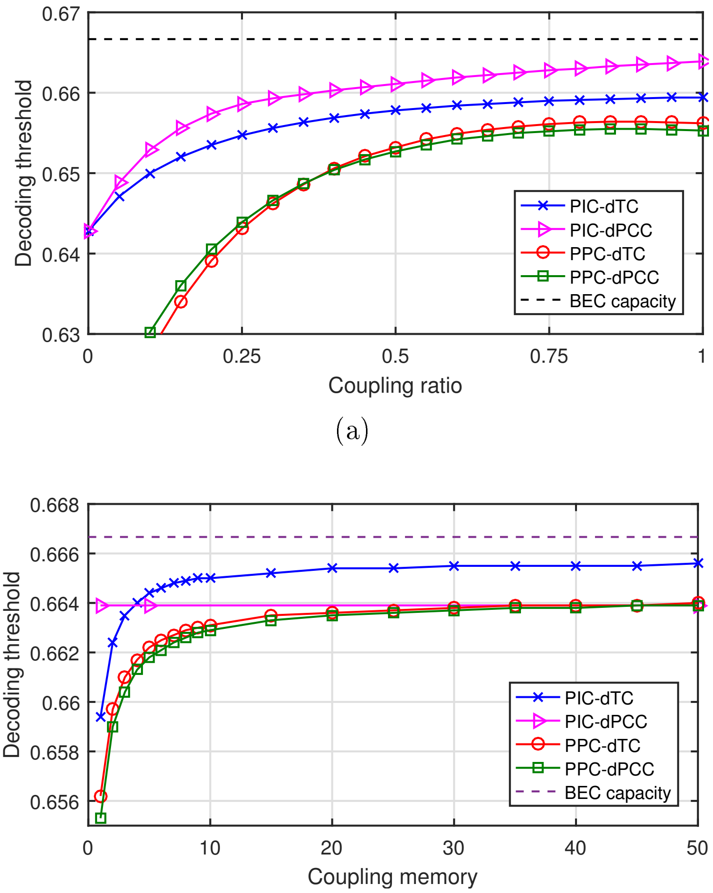
In this section, we present the BP decoding thresholds $\varepsilon_{\mathrm{BP}}$ of our codes over the BEC by using the DE equations derived in Section 5.3. The error floor performance is also discussed.

We consider the rate-$\frac{1}{3}$ turbo code with generator $\mathbf{G}_{\frac{1}{2}} = [1, \frac{5}{7}]$ as the performance benchmark. For constructing PIC codes, we consider $\mathbf{G}_{\frac{2}{3}} = \left[1\ 0\ \frac{5}{7};\ 0\ 1\ \frac{3}{7}\right]$. For constructing PPC codes, we consider $\mathbf{G}_{\frac{2}{3}} = \left[1\ 0\ \frac{4}{7};\ 0\ 1\ \frac{5}{7}\right]$. We choose different generators for PIC codes and PPC codes due to the different coupling design. PIC codes require a good information distance spectrum [83] as only information sequences are coupled. PPC codes require a good codeword distance spectrum [64] because the distance properties of parity sequences can also affect the decoding performance.

5.4.1 Decoding Thresholds at $R = \frac{1}{3}$

In Fig. 5.7a, we compare the BP decoding thresholds of our codes with coupling memory $m = 1$ and various coupling ratio λ. The decoding thresholds of PIC codes improve with λ increasing, and the improvement becomes smaller when λ approaches 1. When $\lambda = 1$, PIC-dPCCs have the best decoding threshold which is within 0.0028 to the BEC capacity. PIC-dTCs have a gap to the BEC capacity of around 0.0072. The decoding thresholds of PPC-dTCs and PPC-dPCCs are close. Both of them have a gap to the BEC capacity of around 0.01.

To see how decoding threshold improves with the coupling memory m, we plot the decoding thresholds of our codes with $\lambda = 1$ as functions of m in Fig. 5.7b. It is observed that the decoding thresholds of all codes except PIC-dPCC improve with m. In particular, when $m = 50$, they can approach the BEC capacity with a gap around 0.001 to 0.002. Although it is unclear how the decoding performance would be when m approach infinite, the thresholds still keep improving even when $m > 50$. For example, the decoding threshold of the PIC-dTC with $\lambda = 1$ and $m = 200$ is 0.6657, which is 0.0001 better than that with $m = 50$. In contrast, although PIC-dPCC outperforms others when $m = 1$, its decoding threshold does not improve with m.



(a)

(b)

Figure 5.7: BP decoding thresholds of rate-$\frac{1}{3}$ partially coupled dTCs with a) $m = 1$ over coupling ratio, and b) $\lambda = 1$ over coupling memory.

In Table 5.3, we compare our codes with the PIC-TCs and PPC-TCs in [94], and the SC-TCs (SC-PCCs, SC-SCCs, and BCCs[2]) in [38]. Code rate $R = \frac{1}{3}$ and $m \in \{1, 5, 50\}$ are considered. All codes are constructed with state-4 RSC codes. When $m = 1$, PIC-dTCs and PIC-dPCCs outperform PIC-TCs, while PPC-dTCs and PPC-dPCCs outperform PPC-TCs. PIC-dPCCs are slightly worse than the type-II BCCs but outperform all other codes. When $m = 50$, PIC-dTCs are

---

[2] Note that the decoding thresholds of BCCs shown in Table 5.3 are different from the decoding thresholds given in [38] due to different termination methods. In [38], the authors consider using parity bits for termination, i.e., the coupled parity bits at the end of the coupling chain are perfectly known by the decoder. In contrast, we consider a practical termination method, where the coupling is terminated by appending all-zero bits at the end of information sequence.

Table 5.3: Decoding thresholds for rate-$\frac{1}{3}$ spatially coupled codes.

Ensemble	$\varepsilon_{\mathrm{BP}}^{(m=1)}, \lambda$	$\varepsilon_{\mathrm{BP}}^{(m=5)}, \lambda$	$\varepsilon_{\mathrm{BP}}^{(m=50)}, \lambda$
PIC-TC	0.6576, [0.37, 0.42]	0.6625, 0.5	0.6639, 0.5
PPC-TC	0.6545, [0.32, 0.35]	0.6618, [0.58, 0.61]	0.6658, [0.78, 0.88]
PIC-dTC	0.6594, 1	0.6644, 1	0.6656, 1
PIC-dPCC	0.6639, 1	0.6639, 1	0.6639, 1
PPC-dTC	0.6564, [0.83, 0.93]	0.6622, 1	0.6640, 1
PPC-dPCC	0.6555, [0.83, 0.94]	0.6619, 1	0.6639, 1
SC-PCC	0.6553	0.6553	0.6553
SC-SCC	0.6437	0.6654	0.6654
Type-I BCC	0.6553	0.6619	0.6639
Type-II BCC	0.6641	0.6653	0.6653

slightly worse than PPC-TCs but outperform all other codes. On the other hand, PPC-dTCs and PPC-dPCCs slightly outperform type-I BCCs.

5.4.2 Decoding Threshold with Random Puncturing

Table 5.4 shows the optimal decoding thresholds and the corresponding λ of the partially coupled dTCs for various code rates[3]. At a high code rate, PPC codes outperform PIC codes. Note that choosing $\lambda = 1$ always maximizes the decoding thresholds of PIC codes, which may not be the case for PPC codes. To be specific, for a fixed m, the optimized λ which gives the best decoding threshold for PPC codes decreases with an increasing code rate. However, as we will show later in Sec. 5.5, the error floor of partially coupled dTCs rises with a decreasing λ. This means that the PPC-dTCs with optimized decoding thresholds may suffer from high error floor.

To achieve a better trade-off between waterfall and error floor at high code rates, we allow both information sequences and parity sequences to be coupled

---

[3] As the decoding thresholds of PIC-dPCCs do not improve with coupling memory, we only put the decoding thresholds for $m = 1$.

Table 5.4: Decoding thresholds for punctured partially coupled dTCs.

R	Ensemble	$\varepsilon_{\mathrm{BP}}^{(m=1)}, \lambda$	$\varepsilon_{\mathrm{BP}}^{(m=5)}, \lambda$	$\varepsilon_{\mathrm{BP}}^{(m=50)}, \lambda$
1/2	PIC-dTC	0.4905, 1	0.4955, 1	0.4966, 1
	PIC-dPCC	0.4910, 1	-	-
	PPC-dTC	0.4887, [0.75, 0.78]	0.4960, 1	0.4983, 1
	PPC-dPCC	0.4872, [0.75, 0.77]	0.4953, 1	0.4982, 1
2/3	PIC-dTC	0.3197, 1	0.3237, 1	0.3246, 1
	PIC-dPCC	0.3146, 1	-	-
	PPC-dTC	0.3231, [0.60, 0.63]	0.3301, [0.86, 0.94]	0.3325, 1
	PPC-dPCC	0.3214, [0.57, 0.65]	0.3293, [0.80, 0.96]	0.3324, 1
3/4	PIC-dTC	0.2343, 1	0.2377, 1	0.2383, 1
	PIC-dPCC	0.2275, 1	-	-
	PPC-dTC	0.2407, [0.49, 0.57]	0.2472, [0.72, 0.84]	0.2495, 1
	PPC-dPCC	0.2393, [0.49, 0.56]	0.2466, [0.75, 0.78]	0.2494, 1
4/5	PIC-dTC	0.1839, 1	0.1867, 1	0.1872, 1
	PIC-dPCC	0.1768, 1	-	-
	PPC-dTC	0.1916, [0.45,0.47]	0.1975, [0.65, 0.80]	0.1996, 1
	PPC-dPCC	0.1903, [0.43, 0.49]	0.1969, [0.63, 0.77]	0.1995, 1
9/10	PIC-dTC	0.0872, 1	0.0887, 1	0.0890, 1
	PIC-dPCC	0.0820, 1	-	-
	PPC-dTC	0.0938, [0.26, 0.32]	0.0982, [0.48, 0.54]	0.0997, 1
	PPC-dPCC	0.0932, [0.28, 0.30]	0.0978, [0.46, 0.56]	0.0996, 1

simultaneously, which we call partially hybrid coupled duo-binary codes (PHC-dTCs). The DE equations of PHC-dTCs are derived similarly to PIC-dTCs and PPC-dTCs shown in Sec. 5.3, so we omit the details here. Let λ_{i} and λ_{p} represent the coupling ratio corresponding to the coupled information sequences and the coupled parity sequences, respectively. As the error floor of partially coupled dTCs rises with a decreasing λ, we let $\lambda_{\mathrm{i}} + \lambda_{\mathrm{p}} = 1$ to ensure that the PHC-dTCs have better error floor than the PPC-dTCs shown in Table. 5.4. Then, we search for the range of λ_i which gives the best decoding threshold. The optimized decoding thresholds and corresponding λ_i are shown in Table 5.5. When $m = 1$, the decoding thresholds of PHC-dTCs are between those of PIC-

Table 5.5: Decoding thresholds for punctured PHC-dTCs.

R	$\varepsilon_{\mathrm{BP}}^{(m=1)}, \lambda_{\mathrm{i}}$	$\varepsilon_{\mathrm{BP}}^{(m=5)}, \lambda_{\mathrm{i}}$
1/2	0.4897, [0.63,0.84]	0.4960, [0.00,0.40]
2/3	0.3227, [0.54,0.67]	0.3301, [0.06,0.34]
3/4	0.2399, [0.60,0.67]	0.2472, [0.18,0.39]
4/5	0.1905, 0.70	0.1975, [0.29,0.41]
9/10	0.0927, [0.75,0.82]	0.0981, [0.48,0.56]

*$\lambda = 1$ and $\lambda_{\mathrm{p}} = 1 - \lambda_{\mathrm{i}}$.

dTCs and PPC-dTCs. When $m = 5$, the decoding thresholds of PHC-dTCs are close to PPC-dTCs. This means that PHC-dTCs can achieve a close-to-optimal performance in the waterfall regime while maintaining a low error floor, and we will show that in Fig. 5.10, Sec. 5.6.

5.5 Error Floor Performance

Here, we investigate the error floor performance of partially coupled dTCs via distance property analysis. As discussed in various literature [83–85,87], the error floor performance of a turbo code depends on its distance spectrum, especially the free distance d_{free} and the free distance multiplicity $A(d_{\mathrm{free}})$. In particular, the free distance asymptote of a turbo code under ML decoding over the AWGN channel is

$$P_b \approx \frac{A(d_{\mathrm{free}})\bar{w}_{\mathrm{free}}}{K} Q\left(\sqrt{\frac{2d_{\mathrm{free}}RE_b}{N_0}}\right), \tag{5.15}$$

where $\bar{w}_{\mathrm{free}}$ is the average weight of the information sequences causing free-distance codewords. In the following, we will discuss the impacts of coupling level and coupling ratio on the BER in the error floor regime by using the free distance asymptote. Strictly speaking, the distance-based analysis does not take the sub-optimally of iterative decoding into account, so it may not give a very tight bound on the error floor BER. With this in mind, several works [78, 81, 98] have demonstrated that such an analysis can roughly reflect the error floor performance of different classes of spatially coupled turbo-like codes. Thus, we choose to use free distance asymptote to give a rough description of how our partially coupled codes behave in the error floor regime.

First, we introduce a notation θ, which is the proportion of uncoupled sequence in a CB. To be specific, for PIC-dTCs and PPC-dTCs, θ is the ratio between the length of the uncoupled information sequence and the information length, i.e.,

$$\theta \triangleq \frac{\|\mathbf{u}_{t,t}\|}{\|\mathbf{u}_t\|} = 1 - \frac{K_c}{K} = 1 - \lambda. \tag{5.16}$$

For PPC-dTCs and PPC-dPCCs, θ is the ratio between the length of the uncoupled parity sequence and the parity length, i.e.,

$$\theta \triangleq \frac{\|\mathbf{v}_{t,t}\|}{\|\mathbf{v}_t\|} = 1 - \frac{K_c}{2K} = \begin{cases} 1 - \dfrac{\lambda}{2} \text{ for PPC-dTCs} \\ 1 - \lambda \text{ for PPC-dPCCs} \end{cases}. \tag{5.17}$$

For PIC-dTCs and PIC-dPCCs, at most K information bits can be coupled so that $0 \leq \theta \leq 1$. Likewise, $0 \leq \theta \leq 1$ for PPC-dPCCs because at most $2K$ parity bits can be coupled. For PPC-dTCs, $0.5 \leq \theta \leq 1$ because at most half of the parity sequence in a CB can be coupled.

We first consider the cases that $\theta > 0$. For PIC-dTCs and PIC-dPCCs with $\mathbf{G}_{\frac{2}{3}} = \left[1\ 0\ \frac{\mathbf{g}_f}{\mathbf{g}_b};\ 0\ 1\ \frac{\mathbf{g}'_f}{\mathbf{g}_b}\right]$, their free distances are the same as the free distance of the turbo code with $\mathbf{G}_{\frac{1}{2}} = \left[1\ \frac{\mathbf{g}_f}{\mathbf{g}_b}\right]$ when $\lambda < 1$. This is because when the bits leading to a CB decoding error event are not coupled, the extrinsic information from other CBs is unable to correct this error event. We call this case the uncoupled error event. On the contrary, when the bits leading to a CB decoding error event are coupled, it is more likely that it can be corrected in the later decoding iterations with the extrinsic information from other CBs. In another aspect, although the free distance stays constant when $\theta > 0$, its multiplicity decreases with θ because the probability that uncoupled error events occur reduces with θ. Thus, the error floor drops. When PIC-dTCs and PIC-dPCCs have the same θ, they have similar free distance multiplicity and thus similar free distance asymptote. Due to similar reasons, we can say that the error floor of PPC-dTCs and PPC-dPCCs reduces with decreasing θ. When PPC-dTCs and PPC-dPCCs have the same θ, they have similar free distance asymptote.

In Table 5.6, we show d_{free} and $\bar{A}(d_{\text{free}})$ for the partially coupled dTCs with $K = 1000$. The component codes we considered here are the same as in Sec. 5.4. Note that the exact free distance[4] $\widetilde{d}_{\text{free}}$ and its multiplicity of a partially coupled dTC are affected by information length, interleaver, and coupling indices (position of coupled bits), we consider the averaged free distance multiplicity $\bar{A}(d_{\text{free}})$ with randomly generated interleaver and coupling indices. The method to obtain $\bar{A}(d_{\text{free}})$ is given in Appendix 5.B.

Now we consider that $\theta = 0$. Recall that PPC-dTCs have $0.5 \le \theta \le 1$, so they are not considered in this case. For PIC-dTC, PIC-dPCC, and PPC-dPCC with $K = 1000$ and $\lambda = 1$ the minimum $\widetilde{d}_{\text{free}}$ we found by searching over randomly generated interleaver and coupling indices are 22, 12, and 28, respectively. This suggests that the PIC-dTCs and PPC-dPCCs might have better error floor performance than the PIC-dPCCs.

---

[4] For a partially coupled dTC, the exact free distance $\widetilde{d}_{\text{free}}$ is the minimum codeword weight for a group of fixed information length, interleaver, and coupling indices. This is different from the free distance d_{free} of the code ensemble, which is the minimum codeword weight for infinite information length, all possible interleaver, and all possible coupling indices. In particular, free distance is the lower bound of the exact free distance.

Table 5.6: Free distance multiplicity for partially coupled dTCs.

	$\bar{A}(10)$		$\bar{A}(6)$	
θ	PIC-dTC	PIC-dPCC	PPC-dTC	PPC-dPCC
0.7	0.98	0.97	0.47	0.47
0.6	0.69	0.68	0.26	0.26
0.5	0.49	0.49	0.13	0.13
0.4	0.29	0.30	-	0.05
0.3	0.17	0.17	-	0.02
0.2	0.09	0.08	-	<0.01
0.1	0.01	0.01	-	<0.01

*$K = 1000$, $m = 1$.

Table 5.7: Error floor BER for partially coupled dTCs at $E_b/N_0 = 3$ dB.

	Error floor BER			
θ	PIC-dTC	PIC-dPCC	PPC-dTC	PPC-dPCC
0.7	4.8×10^{-7}	4.9×10^{-7}	5.4×10^{-6}	4.7×10^{-6}
0.6	3.5×10^{-7}	3.4×10^{-7}	2.0×10^{-6}	2.1×10^{-6}
0.5	2.6×10^{-7}	2.3×10^{-7}	7.3×10^{-7}	7.8×10^{-7}
0.4	1.8×10^{-7}	1.7×10^{-7}	-	3.9×10^{-7}
0.3	1.3×10^{-7}	1.4×10^{-7}	-	1.5×10^{-7}
0.2	8.0×10^{-8}	9.9×10^{-7}	-	6.1×10^{-8}
0.1	4.9×10^{-8}	7.0×10^{-8}	-	1.6×10^{-8}
0	1.8×10^{-9}	3.4×10^{-8}	-	$< 10^{-9}$

*$K = 1000$, $L = 100$, $m = 1$.

In 5.7, we show the error floor BER for our partially coupled dTCs over the AWGN channel. For each class of codes, the error floor BER decreases with θ. When $\theta > 0$, PIC-dTCs and PIC-dPCCs have similar error floor BER, while PPC-dTCs and PPC-dPCCs have similar error floor BER. For $0 < \theta \leq 0.2$, although PPC-dPCCs have a smaller free distance than PIC codes, PPC-dPCCs have a lower error floor than PIC codes due to low free distance multiplicity. When $\theta = 0$, the PIC-dTC and the PPC-dPCC have significantly lower error floor BER than the PIC-dPCC as they have much higher free distance.

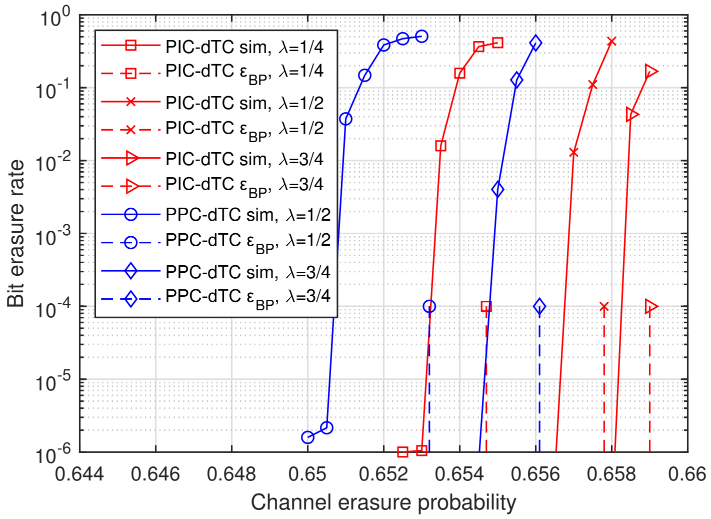


Figure 5.8: Error performance of PIC-dTCs and PPC-dTCs with $K = 10^5$ over the BEC.

5.6 Simulation Results

In this section, we present the simulation results for the partially coupled dTCs. We first present the results over the BEC to verify the DE analysis. Later, we investigate the performance of the partially coupled dTCs over the AWGN channel. For all the results in this section, $m = 1$ is employed. Also, $R = \frac{1}{3}$ is employed if not specified. We set $L = 100$ to minimize the rate loss due to coupling termination unless otherwise specified.

5.6.1 Error Performance over the BEC

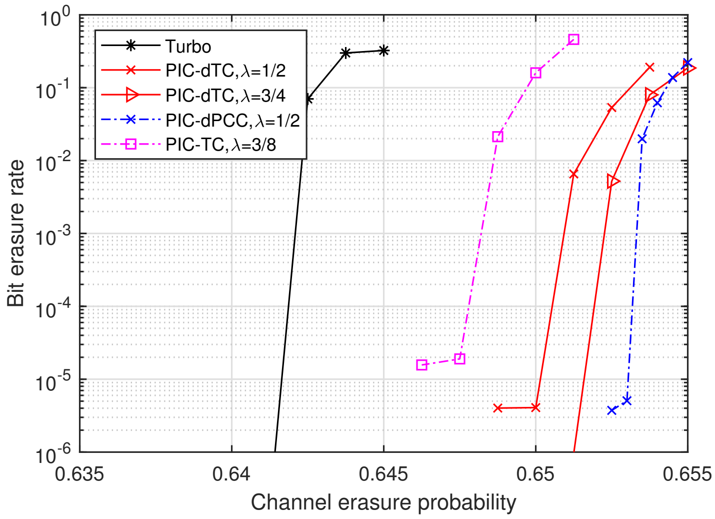
In this subsection, we demonstrate the error performance of our codes over the BEC. The component codes we used here are the same as in Sec. 5.4. Turbo code interleavers and coupling indices are randomly chosen. FF-FB decoding is adopted. The error performance is measured in terms of bit erasure rate (BER) versus the BEC erasure probability.

To verify the DE analysis, we plot the decoding thresholds and the simulated

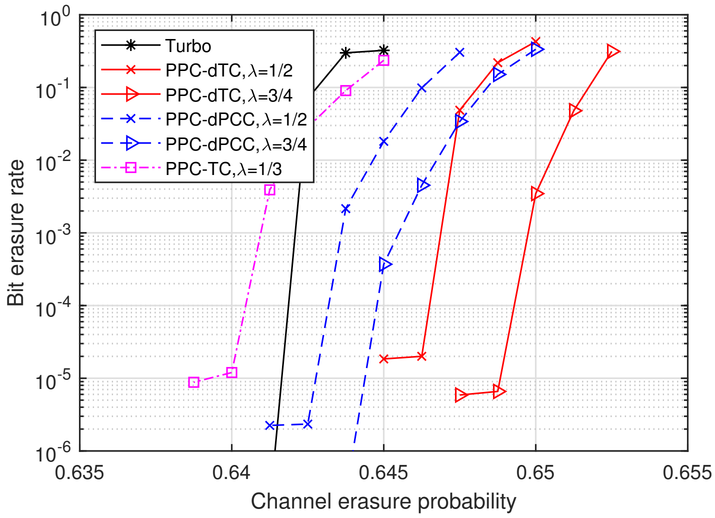
BER curves (denoted as “sim” in the legend) of PIC-dTCs and PPC-dTCs in Fig. 5.8. For simulation, $K = 10^5$ is employed. At BER $= 10^{-5}$, all PIC-dTCs are within 0.002 to their decoding thresholds, and all PPC-dTCs are within 0.004 to their decoding thresholds. This is in agreement with our DE analysis. In addition, it can be seen that the PPC-dTCs have a larger gap to the decoding threshold than the PIC-dTCs with the same λ. This suggests that PPC codes may have worse scaling parameter [99, 100] than the PIC codes, i.e., the PPC codes may suffer more performance degradation than the PIC codes when K decreases.

To evaluate the performance of our codes with moderate blocklength, we plot the BER of the our codes with $K = 10^4$ in Fig. 5.9. The PIC-TC and the PPC-TC in [97] with $K = 10^4$ and optimized λ, and the turbo code with $K = 10^6$ are also shown in Fig. 5.9. All of our codes outperform the turbo code even though they have almost the same total information length. Our codes also outperform the optimized PIC-TC and PPC-TC. Particularly, the PIC-dPCC with $\lambda = \frac{1}{2}$ has the best error performance among all simulated codes. The performance of PPC-dTCs is better than PPC-dPCCs but worse than PIC-dTCs when they have the same λ.

Next, we investigate the performance of our codes with random puncturing over the BEC. In Fig. 5.10, we plot the error performance of our codes with $K = 10^4$ and the turbo code with $K = 10^6$ at code rate $R = \frac{4}{5}$. Although the PPC-dTC has the best performance in the waterfall regime among all simulated codes, its error floor is at a BER above 10^{-5}. The PPC-dPCC has slightly worse performance in the waterfall regime but a lower error floor compared to PPC-dTC. Both PIC codes show an error floor below 10^{-6}, but their performance in the waterfall regime is worse than the PPC codes. In contrast, PHC-dTC has performance close to PPC-dTC in the waterfall regime and an error floor lower



(a) PIC codes.



(b) PPC codes.

Figure 5.9: Error performance of partially coupled codes with $K = 10^4$ over the BEC.

than PPC-dTC.

Finite Length Scaling

Here, we demonstrate how finite-length scaling [100, 101] affects the performance of our codes. Finite-length scaling refers to that the performance of a code degrades with its length. The speed of performance degradation depends on

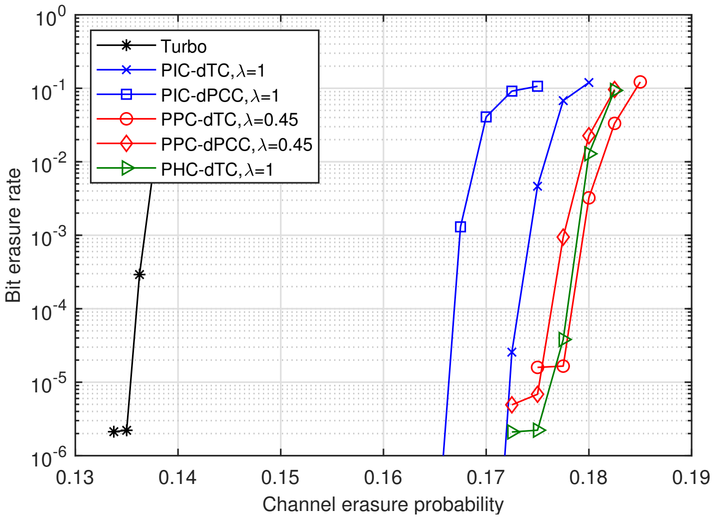


Figure 5.10: Error performance of rate-$\frac{4}{5}$ partially coupled dTCs with $K = 10^4$ over the BEC. $(\lambda_\mathrm{i}, \lambda_\mathrm{p}) = (0.7, 0.3)$ for PHC-dTC.

the code structure. Although the BP decoding threshold of a code indicates the performance of this code under infinite information length and random interleaving, the coupling ratio that maximizes iterative decoding threshold may not promise optimal performance for a short information length due to scaling.

For our partially coupled dTCs, we observe that the performance degradation becomes worse when coupling ratio λ increases. To demonstrate that, Fig. 5.11 shows the performance of PIC-dTCs with $K \in \{10^3, 10^4\}$ at $R = \frac{4}{5}$. Recall that the decoding threshold of PIC-dTC is maximized when $\lambda = 1$. When $K = 10^3$, the PIC-dTC with $\lambda = 1$ does not show significant performance improvement over the PIC-dTC with $\lambda = 0.5$ when channel erasure probability is above 0.16. When $K = 10^4$, the PIC-dTC with $\lambda = 1$ outperforms the PIC-dTC with $\lambda = 0.5$, which complies with their decoding threshold.

We also observe that PPC codes suffer from more performance degradation than PIC codes. To show that, we plot the performance of our codes with $K = 10^3$ at $R = \frac{4}{5}$ in Fig. 5.12. Although we have shown that PPC codes outperforms PIC codes when $K = 10^4$ in Fig. 5.10, we observe that PIC codes outperform

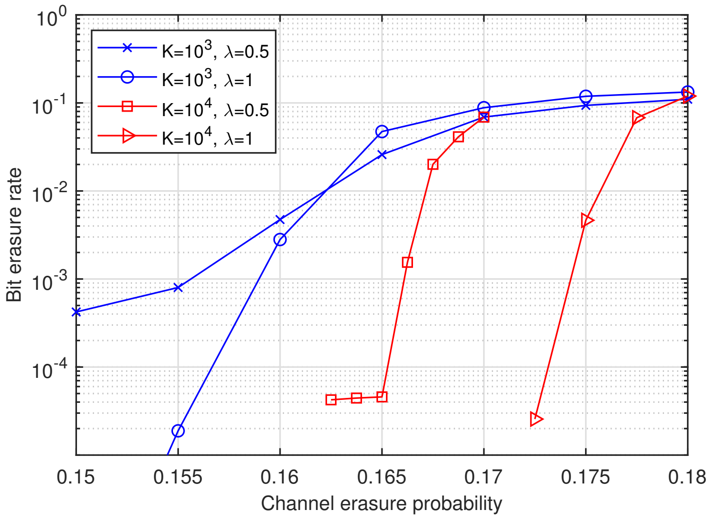


Figure 5.11: Error performance of rate-$\frac{4}{5}$ PIC-dTCs over the BEC.

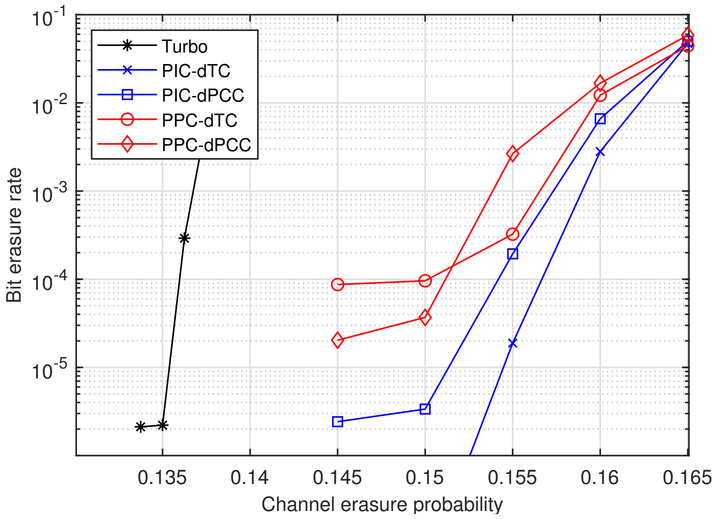


Figure 5.12: Error performance of rate-$\frac{4}{5}$ partially coupled dTCs with $K = 10^3$ over the BEC.

PPC codes when $K = 10^3$.

5.6.2 Error Performance over the AWGN Channel

In this subsection, we investigate the performance of our codes over the AWGN channel. We evaluate the error performance in terms of TB error rate (TBER). We use the LTE turbo codes [12] with generator $\mathbf{G}_{\frac{1}{2}} = [1 \ \frac{15}{13}]$ as a performance benchmark. For constructing PIC codes, we use generator $\mathbf{G}_{\frac{2}{3}} = \left[1 \; 0 \; \frac{15}{13}; \; 0 \; 1 \; \frac{17}{13}\right]$. For constructing PPC codes, we use generator $\mathbf{G}_{\frac{2}{3}} = \left[1 \; 0 \; \frac{17}{13}; \; 0 \; 1 \; \frac{15}{13}\right]$. The LTE turbo interleaver [12] is also applied to our codes, while the coupling indices are randomly chosen. The FF-FB decoding scheme[5] is adopted if otherwise mentioned.

We first show the error performance of PIC-dTCs for various λ and K. We consider that the length of a TB is 102400, and $(K, L) \in \{(1024, 100), (5120, 20)\}$ for both LTE turbo codes and PIC-dTCs. For $\lambda \in \{\frac{1}{2}, 1\}$, the effective code rates with $K = 1024$ are $R \in \{0.3309, 0.3298\}$, and the effective code rates with $K = 5120$ are $R \in \{0.3275, 0.3218\}$. The simulation results are shown in Fig. 5.13. Although Fig. 5.7a shows that the decoding threshold of PIC-dTCs improve with λ over BEC, it is observed that PIC-dTCs with $\lambda = \frac{1}{2}$ outperform the PIC-dTCs with $\lambda = 1$ over the AWGN channel. This is due to two reasons: one reason is that the rate loss caused by termination increases with λ, and another reason is that the finite-length scaling. Similar observations can be made for PIC-dPCCs and PPC codes. At TBER $= 10^{-2}$ and $\lambda = \frac{1}{2}$, the PIC-dTC with $K = 1024$ can achieve about 0.6 dB gain over the LTE turbo code with $K = 1024$, and the PIC-dTC with $K = 5120$ can achieve about 0.3 dB gain over the LTE turbo code with $K = 5120$.

Next, we show the error performance of our codes under sliding window decoding in Fig. 5.14. We use $K = 1024$ and $\lambda = \frac{1}{2}$ for code construction. Window size 10, $I_{\mathrm{TB}} = 10$, and $I_{\mathrm{CB}} = 5$ are used for sliding window decoding. For comparison, we also plot the error performance of LTE turbo code and the type-I BCC. The BCC has the same component RSC generator as the PPC-

---

[5] For FF-FB scheme, we set $I_{\mathrm{CB}} = 20$ for the CB decoder, i.e., the inner decoding iterations. The CB decoding terminates if the soft-decision outputs of the upper and lower decoder converge. For the outer decoding iteration, the decoder runs for at least 10 iterations, and terminates if the number of CB decoding errors does not decrease over 5 outer iterations.

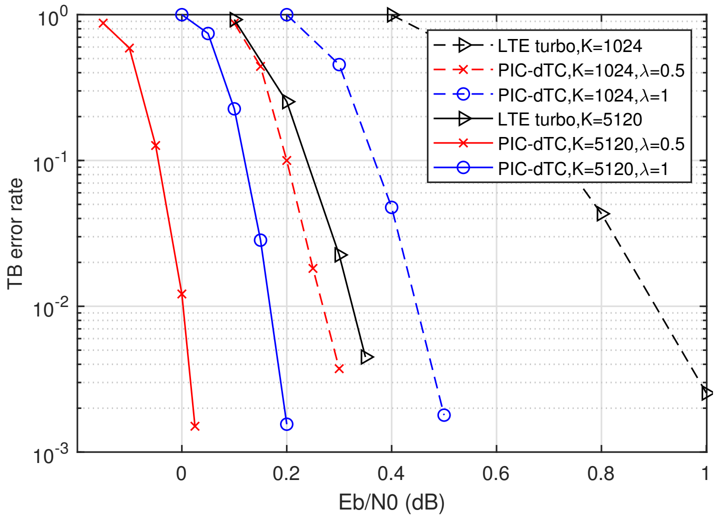
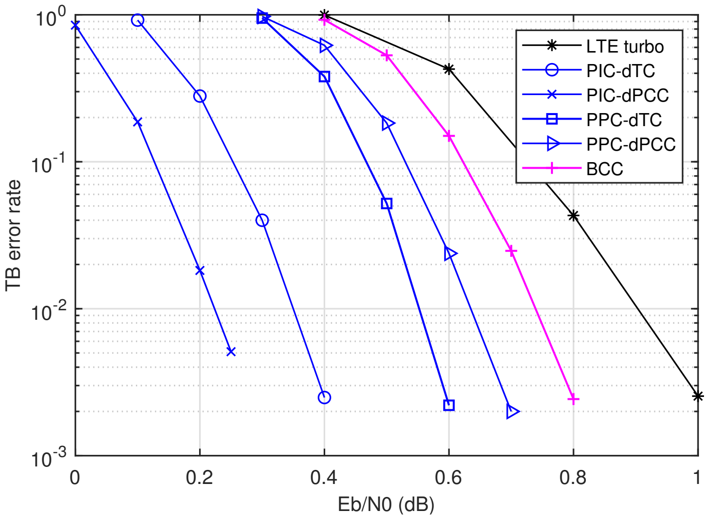


Figure 5.13: Error performance of rate-$\frac{1}{3}$ PIC-dTCs with $KL = 102400$ over the AWGN channel.

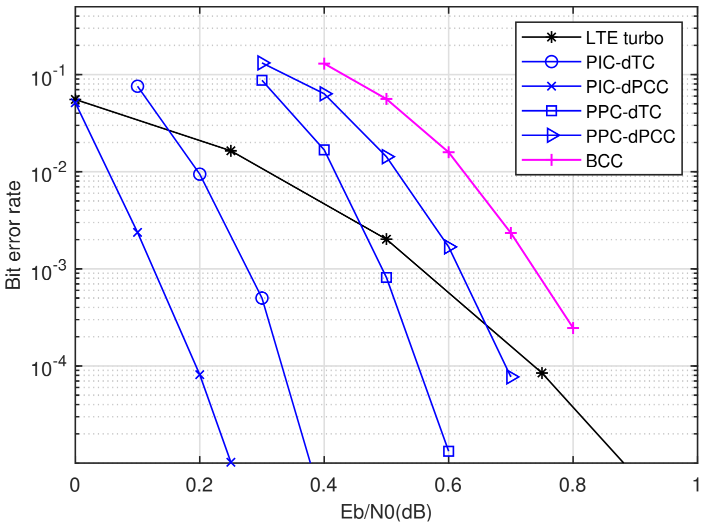
dTC and PPC-dPCC, and has the same sliding window decoder parameter. In terms of TBER, all of our codes outperform the BCC and LTE turbo codes. In terms of BER, although all of our codes have lower BER than the BBC, the PPC codes have higher BER than the LTE turbo code when E_b/N_0 is relatively low. In specific, the PPC-dTC only outperforms the LTE turbo code when E_b/N_0 is above 0.45 dB, while the PPC-dPCC only outperforms the LTE turbo when E_b/N_0 is above 0.65 dB.

5.6.3 Discussion

The simulation results for both the BEC and the AWGN channel show that the existing turbo codes can significantly benefit from partial coupling schemes. The partially coupled dTCs have different characteristics. At $R = \frac{1}{3}$, PIC codes have a better decoding threshold than PPC codes. Specifically, PIC-dPCCs have the best decoding threshold for small coupling memory, while PIC-dTCs have the best decoding threshold for large coupling memory. It is also observed that PIC-dTCs have better error floor performance than PIC-dPCC. For $R \geq \frac{1}{2}$, PPC



(a) TB error rate.



(b) Bit error rate.

Figure 5.14: Error performance of rate-$\frac{1}{3}$ partially coupled dTCs under sliding window decoding over the AWGN channel. $K = 1024, \lambda = \frac{1}{2}$.

codes have a better decoding threshold than PIC codes. PPC-dTCs have the best decoding threshold but the worst error floor performance. By allowing coupling information and parity simultaneously, PHC-dTCs can achieve a better trade-off between decoding threshold and error floor performance.

5.7 Conclusion

In this chapter, we studied spatially coupled turbo-like codes by investigating the impact of partial coupling on the turbo code level and convolutional code level. We introduced four different coding schemes with partial coupling, and constructed PIC-dTCs, PIC-dPCCs, PPC-dTCs, and PPC-dPCCs. We derived the exact DE equations for our codes with general coupling ratio and coupling memory over the BEC. Rate-compatible designs of our codes for high code rates are also studied. The BP decoding thresholds of our codes with optimized coupling ratio were presented for various code rates. The error floor performance was also investigated via distance property analysis. Both theoretical analysis and simulation results showed that the proposed coding scheme can significantly improve the decoding performance over turbo codes.

Appendices

5.A Density Evolution Analysis for $m \geq 1$

5.A.1 PIC-dTCs with $m \geq 1$

For coupling memory $m \geq 1$, at the upper decoder, $\bar{p}^{(i)}_{1,\mathrm{L},t}$ depends on the erasure probabilities of $\mathbf{u}_{t,t}$ and $\mathbf{u}_{t,t+1}, \ldots, \mathbf{u}_{t,t+m}$, while $\bar{p}^{(i)}_{2,\mathrm{L},t}$ depends on the erasure probabilities of $\mathbf{u}_{t-m,t}, \ldots, \mathbf{u}_{t-1,t}$. The average extrinsic erasure probabilities from

$\mathbf{u}_t$ and $\mathbf{u}'_t$ to f^{U} are computed as

$$\bar{p}^{(i)}_{1,\mathrm{L},t} = \varepsilon \cdot p^{(i)}_{1,\mathrm{L},t} \cdot \left(1 - \lambda + \frac{\lambda}{m} \sum_{j=1}^{m} p^{(i-1)}_{2,\mathrm{U},t+j} \cdot p^{(i)}_{2,\mathrm{L},t+j}\right). \tag{5.18a}$$

$$\bar{p}^{(i)}_{2,\mathrm{L},t} = \varepsilon \cdot p^{(i)}_{2,\mathrm{L},t} \cdot \frac{\lambda}{m} \sum_{j=1}^{m} p^{(i-1)}_{1,\mathrm{U},t-j} \cdot p^{(i)}_{1,\mathrm{L},t-j}. \tag{5.18b}$$

The a-posteriori erasure probability of $\mathbf{u}_t$ after i iterations is

$$p^{(i)}_{\mathbf{u}_t} = \varepsilon \cdot p^{(i)}_{1,\mathrm{U},t} \cdot p^{(i)}_{1,\mathrm{L},t} \cdot \left(1 - \lambda + \frac{\lambda}{m} \sum_{j=1}^{m} p^{(i)}_{2,\mathrm{U},t+j} \cdot p^{(i)}_{2,\mathrm{L},t+j}\right). \tag{5.19}$$

5.A.2 PIC-dPCCs with $m \geq 1$

For coupling memory $m \geq 1$, at the upper decoder, $\bar{p}^{(i)}_{1,\mathrm{L},t}$ depends on the erasure probabilities of $\mathbf{u}_{t,t}$, $\mathbf{u}^{\mathrm{U}}_{t,t+1}, \ldots, \mathbf{u}^{\mathrm{U}}_{t,t+m}$, and $\mathbf{u}^{\mathrm{L}}_{t,t+1}, \ldots, \mathbf{u}^{\mathrm{L}}_{t,t+m}$, while $\bar{p}^{(i)}_{2,\mathrm{L},t}$ depends on the erasure probabilities of $\mathbf{u}^{\mathrm{U}}_{t,t-m}, \ldots, \mathbf{u}^{\mathrm{U}}_{t,t-1}$. The average erasure probabilities from $\mathbf{u}_t$ and $\mathbf{u}'_t$ to f^{U} is computed as

$$\bar{p}^{(i)}_{1,\mathrm{L},t} = \varepsilon \cdot p^{(i)}_{1,\mathrm{L},t} \cdot \left(1 - \lambda + \frac{\lambda}{2m} \cdot \sum_{j=1}^{m} \left(p^{(i-1)}_{2,\mathrm{U},t+j} + p^{(i)}_{2,\mathrm{L},t+j}\right)\right), \tag{5.20a}$$

$$\bar{p}^{(i)}_{2,\mathrm{L},t} = \varepsilon \cdot \frac{\lambda}{2m} \sum_{j=1}^{m} p^{(i-1)}_{1,\mathrm{U},t-j} \cdot p^{(i)}_{1,\mathrm{L},t-j}. \tag{5.20b}$$

The a-posteriori erasure probability of $\mathbf{u}_t$ after i iterations is

$$p^{(i)}_{\mathbf{u}_t} = \varepsilon \cdot p^{(i)}_{1,\mathrm{U},t} \cdot p^{(i)}_{1,\mathrm{L},t} \cdot \left(1 - \lambda + \frac{\lambda}{2m} \cdot \sum_{j=1}^{m} \left(p^{(i)}_{2,\mathrm{U},t+j} + p^{(i)}_{2,\mathrm{L},t+j}\right)\right). \tag{5.21}$$

5.A.3 PPC-dTCs with $m \geq 1$

For coupling memory $m \geq 1$, at the upper decoder, $\bar{p}^{(i)}_{2,\mathrm{L},t}$ depends on the erasure probabilities of $\mathbf{v}^{\mathrm{U}}_{t,t-m}, \ldots, \mathbf{v}^{\mathrm{U}}_{t,t-1}$ and $\mathbf{v}^{\mathrm{L}}_{t,t-m}, \ldots, \mathbf{v}^{\mathrm{L}}_{t,t-1}$, while $\bar{q}^{(i)}_{\mathrm{L},t}$ depends on the erasure probabilities of $\mathbf{v}_{t,t}^{\mathrm{U}}$ and $\mathbf{v}_{t,t+1}^{\mathrm{U}}, \ldots, \mathbf{v}_{t,t+m}^{\mathrm{U}}$. The averaged extrinsic probabilities from $\mathbf{u}_t'$ and $\mathbf{v}_t^{\mathrm{U}}$ to f^{U} are computed as

$$\bar{p}_{2,\mathrm{L},t}^{(i)} = \varepsilon \cdot p_{2,\mathrm{L},t}^{(i)} \cdot \frac{\lambda}{2m} \sum_{j=1}^{m} \left(q_{\mathrm{U},t-j}^{(i-1)} + q_{\mathrm{L},t-j}^{(i)} \right), \tag{5.22a}$$

$$\bar{q}_{\mathrm{L},t}^{(i)} = \varepsilon \cdot \left(1 - \frac{\lambda}{2} + \frac{\lambda}{2m} \sum_{j=1}^{m} p_{1,\mathrm{U},t+j}^{(i-1)} \cdot p_{1,\mathrm{L},t+j}^{(i)} \right). \tag{5.22b}$$

The a-posteriori erasure probability of $\mathbf{u}_t$ is computed with the same expression in (5.11).

5.A.4 PPC-dPCCs with $m \geq 1$

For coupling memory $m \geq 1$, at the upper decoder, $\bar{p}_{2,\mathrm{L},t}^{(i)}$ depends on the erasure probabilities of $\mathbf{v}_{t-m,t}^{\mathrm{L}}, \ldots, \mathbf{v}_{t-1,t}^{\mathrm{L}}$, while $\bar{q}_{\mathrm{U},t}^{(i)}$ depends on the erasure probabilities of $\mathbf{u}_{t,t+1}^{\mathrm{L}}, \ldots, \mathbf{u}_{t,t+m}^{\mathrm{L}}$. Thus, the average erasure probabilities from $\mathbf{u}_t'$ and $\mathbf{v}_t$ to f^{U} are computed as

$$\bar{p}_{2,\mathrm{L},t}^{(i)} = \varepsilon \cdot \frac{\lambda}{m} \sum_{j=1}^{m} q_{\mathrm{L},t-j}^{(i)}, \tag{5.23a}$$

$$\bar{q}_{\mathrm{U},t}^{(i)} = \varepsilon \cdot \left(1 - \lambda + \frac{\lambda}{m} \sum_{j=1}^{m} p_{1,\mathrm{L},t+j}^{(i)} \right). \tag{5.23b}$$

The a-posteriori erasure probability of $\mathbf{u}_t$ is computed with the same expression in (5.11).

5.B Process to Obtain Free Distance Multiplicity

Let $T_{\max}$ be sampling number of the code ensembles with the given information length and coupling ratio. To simplify the process of searching the exact free distance $\widetilde{d}_{\text{free}}$, we adopt two commonly used assumptions for turbo codes [83, 85, 87, 89, 102, 103].

- The free distance of a turbo code is given by weight-2 information sequences, i.e., $\bar{w}_{\text{free}} = 2$.
- Let $\mathbf{u}_{(i,D)}$ be a weight-2 information sequence $[u_1, \ldots, u_K]$ with $u_i = u_{i+D} = 1$. The codeword weight of $\mathbf{u}_{(i,D)}$ is larger than d_{free} when $D > D_{\text{max}}$, where D_{max} is a value much larger than the constraint length of the constituent RSC code.

For the results in Table 5.7, we consider $K = 1000$, $m = 1$, $T_{\text{max}} = 2000$, and $D_{\text{max}} = 20$. The steps to obtain $\bar{A}(d_{\text{free}})$ are as follows.

Step 1. Initialize the free distance multiplicity counter $A = 0$.

Step 2. For $T = 1, 2, \ldots, T_{\text{max}}$, obtain the Hamming weight of codeword produced by weight-2 information sequence with the given information length and coupling ratio. Record the total free distance multiplicity.

1. Randomly generate turbo code interleaver and coupling indices.
2. Let $L = 2$. For $i = 1, 2, \ldots, K - (D_{\text{max}} - 1)$ and $0 < D \leq D_{\text{max}}$:
   (a) For CB$_1$, initialize $\mathbf{u}_1 = \mathbf{u}_{(i,D)}$.
   (b) For CB$_2$, initialize $\mathbf{u}_2$ as an all-zero sequence.
   (c) Apply coupling between CB$_1$ and CB$_2$, and encode CB$_1$ and CB$_2$ respectively.
   (d) Compute the Hamming weight of codeword: $d = 2 + w(\mathbf{v}_1) + w(\mathbf{v}_2)$.
   (e) If $d = d_{\text{free}}$, then increase counter A by 1.

Step 3. Compute the averaged free distance multiplicity $\bar{A}(d_{\text{free}}) = \frac{A}{T_{\text{max}}}$

Chapter 6

Generalized Spatially Coupled PCCs

In this chapter, we introduce the generalized spatially coupled parallel concatenated convolutional codes (GSC-PCCs), constructed by applying spatial coupling on parallel concatenated convolutional codes (PCCs) with a fraction of information bits repeated q times. GSC-PCCs can be seen as a generalization of the spatially-coupled parallel concatenated codes proposed in [38]. In addition, the PIC-TCs can also be seen as a subclass of GSC-PCCs. To characterize the asymptotic performance of GSC-PCCs, we derive the corresponding density evolution equations and compute their decoding thresholds. Both density evolution analysis and simulations show the superiority of the proposed codes over the existing spatially coupled turbo-like codes.

6.1 Introduction

In the last two chapters, we studied the partial coupling techniques and constructed various partially couple turbo codes. In the partially coupled scheme, a fraction of the codeword at one time instance is repeated two times and the replica becomes a part of the information sequence of the component encoder at the succeeding time instances. The partial coupling scheme where the information bits are repeated is referred to as the partial information coupling (PIC), while the scheme where the parity bits are repeated is referred to as the partial parity coupling (PPC).

In this chapter, we combine the ideas of spatially coupled parallel concatenated convolutional codes [38] (SC-PCCs) and PIC turbo codes [93, 94] (PIC-TCs) and construct the generalized spatially coupled parallel concatenated convolutional codes (GSC-PCCs) [104] by applying spatial coupling on parallel concatenated convolutional codes (PCCs) with a fraction of information bits repeated q times. The main contributions[1] of this chapter are as follows:

- We introduce the construction method for GSC-PCCs with an arbitrary PCC. First, a fraction λ of the information sequence at one time instance is repeated by q times. Then, the spatial coupling method used for constructing SC-PCCs is applied to the information sequence with partial repetition. In addition, we show that both SC-PCCs and PIC-TCs can be seen as the special subclasses of GSC-PCCs.

- The exact density evolution (DE) equations for the GSC-PCC ensembles on the binary erasure channel (BEC) are derived. For the rate-compatible GSC-PCC ensembles based on random puncturing on parity bits, we find the optimal repetition ratio λ that gives the largest BP decoding threshold for a given target code rate R, coupling memory m, and repetition factor q. The DE analysis results show that GSC-PCCs can surpass SC-PCCs and

---

[1] I was partially involved in the works presented in this chapter. I mainly contributed to the coupling indices design and the implementation of simulation programs.

PIC-TCs of the same rates. In particular, the GSC-PCCs with $q = 6$ and $m = 5$ can have a gap less than 0.005 to the BEC capacity for various code rates.

- We evaluate the error performance of GSC-PCC with randomly generated coupling indices over the BEC via simulation. Simulation results confirm our theoretical analysis and show significant error performance improvement over SC-PCCs and PIC-TCs. Furthermore, we show that the finite-length performance of GSC-PCCs can be further improved by carefully choosing the coupling indices.

6.2 Construction of GSC-PCCs

In this section, we first briefly introduce the PCCs with partial information repetition (PR-PCCs) which are used as the building blocks to construct GSC-PCCs. Then, we present the construction of GSC-PCCs. The constituent PCC can be arbitrary. For simplicity, we consider that the constituent code is a rate-$\frac{1}{3}$ PCC built with two rate-$\frac{1}{2}$ RSC codes throughout this chapter.

6.2.1 Encoding of PR-PCCs

The PR-PCCs are constructed by repeating a fraction of the information bits for q times. The encoder block diagram of PR-PCCs is depicted in Fig. 6.1. A length-K information sequence $\mathbf{u}$ is segmented into two subsequences $\mathbf{u}_{\mathrm{r}}$ and $\mathbf{u}_{\mathrm{o}}$. Then, q copies of $\mathbf{u}_{\mathrm{r}}$ are combined with $\mathbf{u}_{\mathrm{o}}$ to form a length-K' sequence $\mathbf{u}' = [\mathbf{u}_{\mathrm{r}}, \ldots, \mathbf{u}_{\mathrm{r}}, \mathbf{u}_{\mathrm{o}}]$. Next, $\mathbf{u}'$ and its reordered copy $\Pi(\mathbf{u}')$ is encoded by the upper and lower RSC encoders, producing the parity sequences $\mathbf{v}^{\mathrm{U}}$ and $\mathbf{v}^{\mathrm{L}}$, respectively. The codeword is given by $\mathbf{x} = [\mathbf{u}, \mathbf{v}^{\mathrm{U}}, \mathbf{v}^{\mathrm{L}}]$.

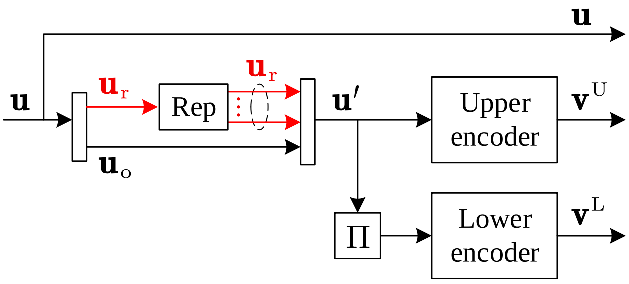


Figure 6.1: Encoder block diagram of PR-dTCs.

We define the repetition ratio as the ratio of the length of the repeated subsequence $\mathbf{u}_\mathrm{r}$ to the length of the encoder input sequence $\mathbf{u}'$, that is

$$\lambda \triangleq \frac{\|\mathbf{u}_\mathrm{r}\|}{\|\mathbf{u}'\|} = \frac{K'-K}{(q-1)K'}, \tag{6.1}$$

and $0 \leq \lambda \leq \frac{1}{q}$. This gives $K = (1-(q-1)\lambda)K'$, $\|\mathbf{u}_\mathrm{r}\| = \lambda K'$, $\|\mathbf{u}_\mathrm{o}\| = (1-q\lambda)K'$, and $\|x\| = 3K' - (q-1)\lambda K'$. The code rate of PR-PCCs is

$$R_0 = \frac{\|\mathbf{u}\|}{\|\mathbf{x}\|} = \frac{1-(q-1)\lambda}{3-(q-1)\lambda}. \tag{6.2}$$

6.2.2 Encoding of GSC-PCCs

We now construct GSC-PCCs by applying spatial coupling to PR-PCCs. As an example, we plot the encoder block diagram of GSC-PCCs with coupling memory $m = 1$ in Fig. 6.2.

An information sequence $\mathbf{u}$ is segmented into L subsequence $\mathbf{u}_1, \ldots, \mathbf{u}_L$ of length K. At time t, $\mathbf{u}_t$ is decomposed into $\mathbf{u}_{t,\mathrm{r}}$ and $\mathbf{u}_{t,\mathrm{o}}$, where $\|\mathbf{u}_{t,\mathrm{r}}\| = \lambda K'$ and $\|\mathbf{u}_{t,\mathrm{o}}\| = (1-q\lambda)K'$. Then, q copies of $\mathbf{u}_{t,\mathrm{r}}$ are combined with $\mathbf{u}_{t,\mathrm{o}}$ to form a length-K' sequence $\mathbf{u}'_t = [\mathbf{u}_{t,\mathrm{r}}, \ldots, \mathbf{u}_{t,\mathrm{r}}, \mathbf{u}_{t,\mathrm{o}}]$. Next, $\mathbf{u}'_t$ is decomposed into $m+1$ subsequences $\mathbf{u}^\mathrm{U}_{t,t}, \ldots, \mathbf{u}^\mathrm{U}_{t,t+m}$ of length $\frac{K'}{m+1}$, where $\mathbf{u}^\mathrm{U}_{t,t+j}$ will become a part of the input sequence to the upper encoder at time $t+j$, $j = 0, \ldots, m$. Meanwhile,

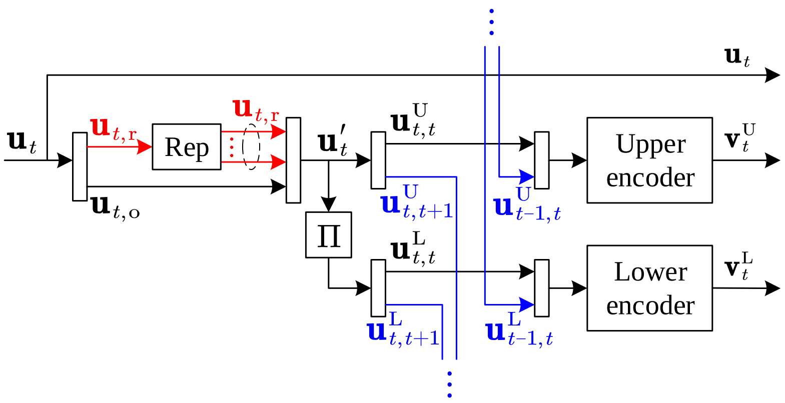

$\mathbf{u}_t$
$\mathbf{u}_t$
$\mathbf{u}_{t,\mathrm{r}}$
Rep
$\mathbf{u}_{t,\mathrm{r}}$
$\mathbf{u}_{t,\mathrm{o}}$
$\mathbf{u}'_t$
Π
$\mathbf{u}^{\mathrm{U}}_{t,t}$
$\mathbf{u}^{\mathrm{U}}_{t,t+1}$
$\mathbf{u}^{\mathrm{U}}_{t-1,t}$
$\mathbf{u}^{\mathrm{L}}_{t,t}$
$\mathbf{u}^{\mathrm{L}}_{t,t+1}$
$\mathbf{u}^{\mathrm{L}}_{t-1,t}$
Upper encoder
Lower encoder
$\mathbf{v}^{\mathrm{U}}_t$
$\mathbf{v}^{\mathrm{L}}_t$

Figure 6.2: Encoder block diagram of the GSC-dPCCs with $m = 1$ at time t.

$\Pi(\mathbf{u}'_t)$ is decomposed into $m+1$ subsequences $\mathbf{u}^{\mathrm{L}}_{t,t}, \ldots, \mathbf{u}^{\mathrm{L}}_{t,t+m}$ of length $\frac{K'}{m+1}$, where $\mathbf{u}^{\mathrm{L}}_{t,t+j}$ will become a part of the input sequence to the lower encoder at time $t+j$, $j = 0, \ldots, m$. Later, the upper encoder inputs $[\mathbf{u}^{\mathrm{U}}_{t-m,t}, \ldots, \mathbf{u}^{\mathrm{U}}_{t,t}]$ and produces the upper parity sequence $\mathbf{v}^{\mathrm{U}}_t$, while the lower encoder inputs $[\mathbf{u}^{\mathrm{L}}_{t-m,t}, \ldots, \mathbf{u}^{\mathrm{L}}_{t,t}]$ and produces the lower parity sequence $\mathbf{v}^{\mathrm{L}}_t$. Eventually, the codeword at time t is $\mathbf{x}_t = [\mathbf{u}_t, \mathbf{v}^{\mathrm{U}}_t, \mathbf{v}^{\mathrm{L}}_t]$.

To initialize and terminate the coupled chain, we can simply set $\mathbf{u}_t = \mathbf{0}$ for $t < 1$ and $t > L$. As a result, the code rate of the GSC-PCC is

$$R = \frac{\sum_{t=1}^{L} \|\mathbf{u}_t\|}{\sum_{t=1}^{L} \|\mathbf{x}_t\| + \sum_{t=L+1}^{L+m} \|\mathbf{v}^{\mathrm{U}}_t\| + \|\mathbf{v}^{\mathrm{L}}_t\|} = \frac{(1-(q-1)\lambda)K'L}{(3-(q-1)\lambda)K'L + 2mK'} \tag{6.3}$$

When $L \to \infty$, the code rate of a GSC-PCC approaches the code rate of its component PR-PCC, i.e.,

$$R \overset{L\to\infty}{=} \frac{1-(q-1)\lambda}{3-(q-1)\lambda}. \tag{6.4}$$

6.2.3 Comparison with Existing Codes

GSC-PCCs can be seen as a generalization of the conventional SC-PCCs. In particular, a GSC-PCC becomes an SC-PCC when $q = 1$ or $\lambda = 0$. GSC-PCCs have significant performance improvement over SC-PCCs. Moreover, GSC-PCCs have a better flexibility on code rate as they can reach any code rate between $\frac{1}{2q+1}$ and $\frac{1}{3}$ without additional rate-matching technique.

PIC-TCs can be seen as special cases of GSC-PCCs with $q = 2$. To be precise, we can obtain the PIC-TCs with $m = 1$ by setting $\mathbf{u}^{\mathrm{U}}_{t,t} = \mathbf{u}^{\mathrm{L}}_{t,t} = [\mathbf{u}_{t,\mathrm{r}}, \mathbf{u}_{t,\mathrm{o}}]$ and $\mathbf{u}^{\mathrm{U}}_{t,t+1} = \mathbf{u}^{\mathrm{L}}_{t,t+1} = \mathbf{u}_{t,\mathrm{r}}$. We can obtain PIC-TCs with $m \geq 1$ by setting $\mathbf{u}^{\mathrm{U}}_{t,t} = \mathbf{u}^{\mathrm{L}}_{t,t} = [\mathbf{u}_{t,\mathrm{r}}, \mathbf{u}_{t,\mathrm{o}}]$ and $\mathbf{u}^{\mathrm{U}}_{t,t+j} = \mathbf{u}^{\mathrm{L}}_{t,t+j} = \mathbf{u}_{t,\mathrm{r},j}$, where $1 \leq j \leq m$ and $\mathbf{u}_{t,\mathrm{r}} = [\mathbf{u}_{t,\mathrm{r},1}, \ldots, \mathbf{u}_{t,\mathrm{r},m}]$. For both PIC-TCs and GSC-PCCs with $q = 2$, coupled bits is protected by four RSC encoders. However, as the coupling of PIC-TCs is on the turbo code level, coupled bits is only protected by two time instants. The coupling of the GSC-PCCs is on the convolutional code level, so that coupled bits can be protected by no less than two time instants.

6.3 Density Evolution Analysis

In this section, we present the DE analysis of GSC-PCCs over the BEC. With the help of compact graphs, we first derive the DE equations for the uncoupled PR-PCCs and then derive the DE equations for the GSC-PCCs with coupling memory $m \geq 1$. Lastly, we evaluate the BP decoding thresholds of our codes over the BEC by using the derived DE equations.

Due to the symmetrical structure of PR-PCCs and GSC-PCCs, we only present the DE equations for the upper encoder to avoid repetition. In addition, as parity bits are not involved in the coupling, the DE equations for the parity

Table 6.1: Notations for the DE analysis of PR-PCCs and GSC-PCCs.

Notation	Definition
f^{U}	Upper decoder
f^{L}	Lower decoder
f^{R}	Partial repeater
$\bar{p}^{(i)}_{\mathrm{L},t}$	Information input erasure probability of f^{U}
$p^{(i)}_{\mathrm{U},t}$	Information output erasure probability of f^{U}
$\bar{p}^{(i)}_{\mathrm{U},t}$	Information input erasure probability of f^{L}
$p^{(i)}_{\mathrm{L},t}$	Information output erasure probability of f^{L}
$\bar{\varphi}^{(i)}_{\mathrm{U},t}$	Input erasure probability from f^{U} to f^{R}
$\bar{\varphi}^{(i)}_{\mathrm{U},t}$	Input erasure probability from f^{L} to f^{R}
$\varphi^{(i)}_{\mathrm{L},t}$	Output erasure probability from f^{R} to f^{U}
$\varphi^{(i)}_{\mathrm{L},t}$	Output erasure probability from f^{R} to f^{L}

bits of PR-PCCs and GSC-PCCs are the same as that of PIC-PCs (which can be found in Sec.4 Chapter 4), so we omit their details there. The notations involved in the DE analysis can be found in Table 6.1.

6.3.1 Density Evolution of PR-PCCs

Fig. 6.3a shows the compact graph of PR-PCCs. For f^{U} at the i-th iteration, both $\mathbf{u}_{\mathrm{o}}$ and $\mathbf{u}_{\mathrm{r}}$ can obtain extrinsic information from f^{L}. Additionally, $\mathbf{u}_{\mathrm{r}}$ can obtain extrinsic information from f^{U} via its $q-1$ replicas. Thus, the average erasure probability from $\mathbf{u}_{\mathrm{o}}$ to f^{U} is $\varepsilon \cdot p^{(i)}_{\mathrm{L},t}$, while the average erasure probability from $\mathbf{u}_{\mathrm{r}}$ to f^{U} is $\varepsilon \left(p^{(i)}_{\mathrm{L},t}\right)^q \cdot \left(p^{(i-1)}_{\mathrm{U},t}\right)^{q-1}$. The input erasure probability of f^{U} is

$$\bar{p}^{(i)}_{\mathrm{L},t} = \varepsilon \left(q\lambda \left(p^{(i)}_{\mathrm{L},t}\right)^q \cdot \left(p^{(i-1)}_{\mathrm{U},t}\right)^{q-1} + (1-q\lambda) \cdot p^{(i)}_{\mathrm{L},t} \right). \tag{6.5}$$

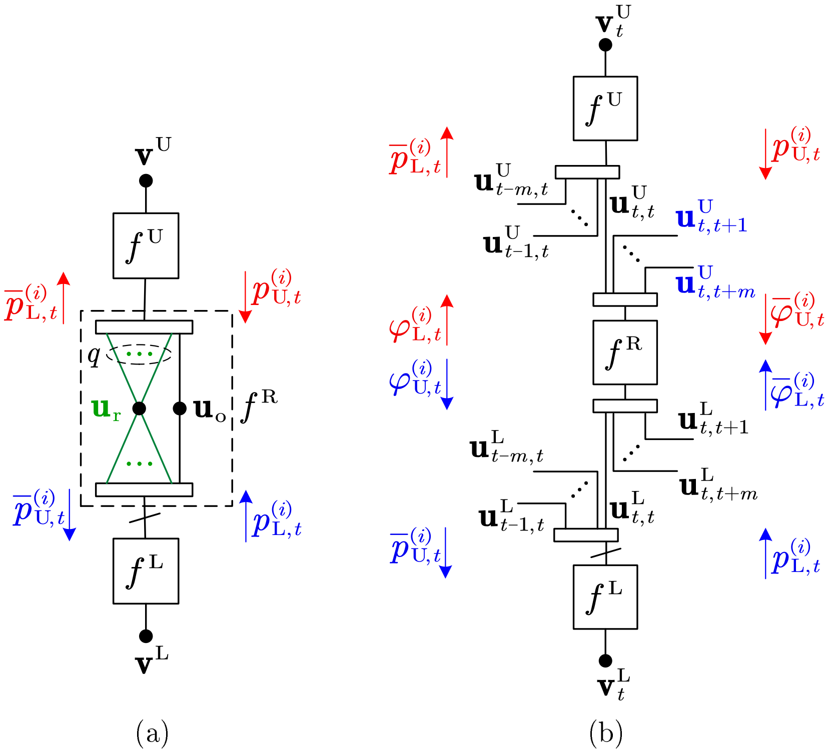

$\mathbf{v}^{\mathrm{U}}$
f^{U}
$\overline{p}^{(i)}_{\mathrm{L},t}$
$p^{(i)}_{\mathrm{U},t}$
q
$\mathbf{u}_{\mathrm{r}}$
$\mathbf{u}_{\mathrm{o}}$
f^{R}
$\overline{p}^{(i)}_{\mathrm{U},t}$
$p^{(i)}_{\mathrm{L},t}$
f^{L}
$\mathbf{v}^{\mathrm{L}}$
(a)
$\mathbf{v}^{\mathrm{U}}_t$
$\mathbf{u}^{\mathrm{U}}_{t-m,t}$
$\mathbf{u}^{\mathrm{U}}_{t,t}$
$\mathbf{u}^{\mathrm{U}}_{t\text{-}1,t}$
$\mathbf{u}^{\mathrm{U}}_{t,t+1}$
$\mathbf{u}^{\mathrm{U}}_{t,t+m}$
$\varphi^{(i)}_{\mathrm{L},t}$
$\varphi^{(i)}_{\mathrm{U},t}$
$\overline{\varphi}^{(i)}_{\mathrm{U},t}$
$\overline{\varphi}^{(i)}_{\mathrm{L},t}$
$\mathbf{u}^{\mathrm{L}}_{t-m,t}$
$\mathbf{u}^{\mathrm{L}}_{t,t+1}$
$\mathbf{u}^{\mathrm{L}}_{t,t+m}$
$\mathbf{u}^{\mathrm{L}}_{t\text{-}1,t}$
$\mathbf{u}^{\mathrm{L}}_{t,t}$
$\mathbf{v}^{\mathrm{L}}_t$
(b)

Figure 6.3: Compact graph of a) RP-PCCs and b) GSC-PCCs with $m \geq 1$.

6.3.2 Density Evolution of GSC-PCCs

Fig. 6.3b depicts the compact graph of GSC-PCCs, where the partial repetition operator is represented by a factor node f^{R}. For f^{U} at time t and the i-th iteration, $\mathbf{u}^{\mathrm{U}}_{t-m,t}, \ldots, \mathbf{u}^{\mathrm{L}}_{t,t}$ can obtain extrinsic information from f^{R} at time $t - m, \ldots, t$, respectively. Node f^{R} can obtain extrinsic information from both f^{U} and f^{L} at time $t, \ldots, t+m$. By taking into account the DE update inside f^{R}, the information input erasure probability of f^{U} is given by

$$\bar{p}_{\mathrm{L},t}^{(i)} = \frac{\varepsilon}{m+1}\sum_{j=0}^{m}\varphi_{\mathrm{L},t-j}, \tag{6.6a}$$

$$\varphi_{\mathrm{L},t-j}^{(i)} = q\lambda\left(\bar{\varphi}_{\mathrm{L},t-j}^{(i)}\right)^{q}\cdot\left(\bar{\varphi}_{\mathrm{U},t-j}^{(i-1)}\right)^{q-1} + (1-q\lambda)\cdot\bar{\varphi}_{\mathrm{L},t-j}^{(i)}, \tag{6.6b}$$

$$\bar{\varphi}_{\mathrm{L},t}^{(i)} = \frac{1}{m+1}\sum_{j=0}^{m}p_{\mathrm{L},t+j}^{(i)}, \tag{6.6c}$$

$$\bar{\varphi}_{\mathrm{U},t}^{(i-1)} = \frac{1}{m+1}\sum_{j=0}^{m}p_{\mathrm{U},t+j}^{(i-1)}. \tag{6.6d}$$

6.3.3 BP Decoding Thresholds

We now evaluate the BP decoding thresholds of our codes over the BEC by using the derived DE equations. We consider that the underlying component code is a rate-$\frac{1}{3}$ turbo code with generator $\mathbf{G} = [1, \frac{5}{7}]$. Given a target code rate R and repetition times q, we optimize the repetition ratio λ such that the BP decoding threshold is maximized. Same as Chapter 4 and Chapter 5, we consider the rate-matching design by randomly puncturing the parity bits.

From Table 6.2, it can be observed that the BP decoding thresholds of GSC-PCCs improve with increasing q for all considered code rates and coupling memories. Moreover, the thresholds of GSC-PCCs surpass those of SC-PCCs and PIC-TCs (see Table 4.3, Chapter 4) for the same coupling memories and same code rates.

6.4 Simulation Results

To verify our DE analysis, we simulate the error performance of our codes over the BEC. The error performance is measured in terms of bit erasure rate versus the erasure probability over the BEC. For all the simulated results, we set $K = 10^5$,

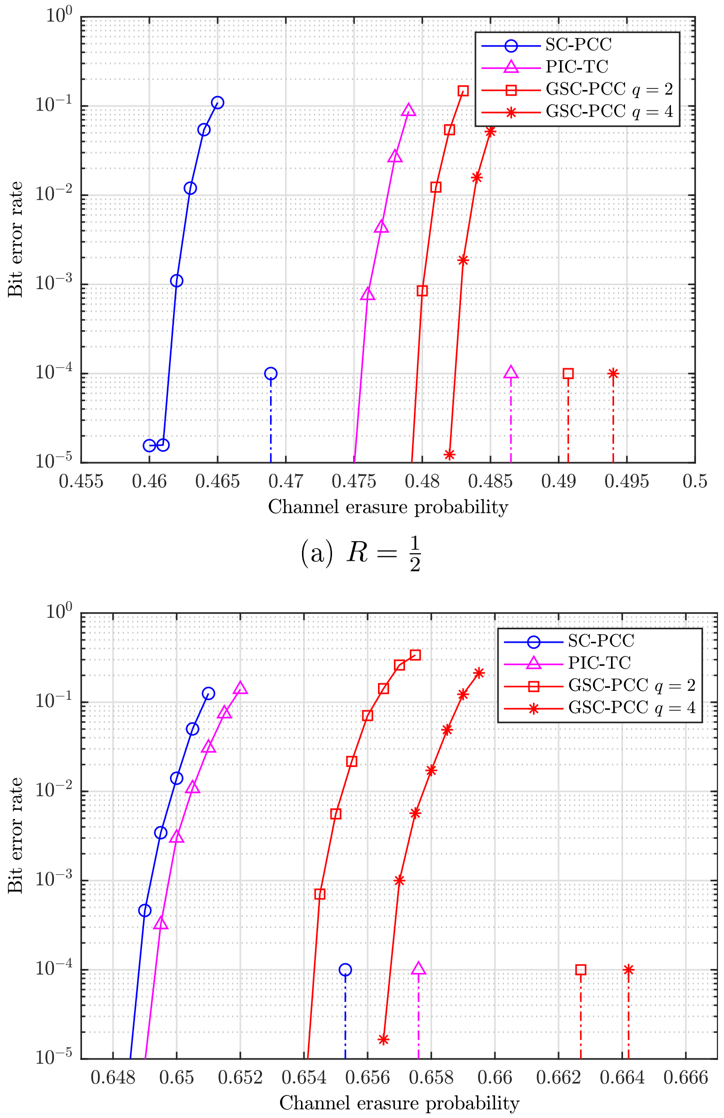
Table 6.2: BP decoding thresholds for GSC-PCCs.

Rate	q	$\varepsilon_{\text{BP}}^{(m=1)}$, λ	$\varepsilon_{\text{BP}}^{(m=3)}$, λ	$\varepsilon_{\text{BP}}^{(m=5)}$, λ
1/4	2	0.7478, [0.319,0.353]	0.7491, 0.5	0.7491, 0.5
	4	0.7487, [0.152,0.162]	0.7495, [0.216,0.229]	0.7497, 0.24
	6	0.7490, [0.112,0.116]	0.7496, [0.134,0.143]	0.7497, [0.143,0.158]
1/3	2	0.6627, [0.37,0.39]	0.6647, 0.5	0.6647, 0.5
	4	0.6642, [0.162,0.172]	0.6656, [0.216,0.229]	0.6660, 0.25
	6	0.6648, [0.121,0.122]	0.6658, [0.138,0.146]	0.6661, [0.151,0.158]
1/2	2	0.4907, 0.44	0.4938, 0.5	0.4938, 0.5
	4	0.4940, [0.187,0.188]	0.4969, 0.23	0.4978, 0.25
	6	0.4952, 0.131	0.4974, [0.150,0.151]	0.4982, [0.156,0.160]
3/4	2	0.2326, 0.5	0.2352, 0.5	0.2352, 0.5
	4	0.2380, [0.201,0.206]	0.2430, 0.24	0.2443, 0.25
	6	0.2406, 0.137	0.2442, [0.152,0.154]	0.2457, [0.162,0.163]

$m = 1$, and $L = 100$.

Fig. 6.4 shows the simulated performance and BP decoding thresholds of rate-$\frac{1}{2}$ and rate-$\frac{1}{3}$ GSC-PCCs, respectively. In the same figure, we also include the simulated performance and decoding thresholds of SC-PCCs and PIC-TCs for comparison, where the code rates, input information lengths, coupling lengths, and coupling memory are the same as those of GSC-PCCs. The interleavers and the coupling indices are randomly chosen.

For both rates, the simulation results show that the GSC-PCCs perform better than the SC-PCCs and PIC-TCs of the same rates. We observed that choosing $q = 2$ is sufficient to allow GSC-PCCs to outperform SC-PCCs and PIC-TCs. Furthermore, GSC-PCCs with $q = 4$ have a noticeable performance gain over the GSC-PCCs with $q = 2$. All these observations agree with the DE analysis results.



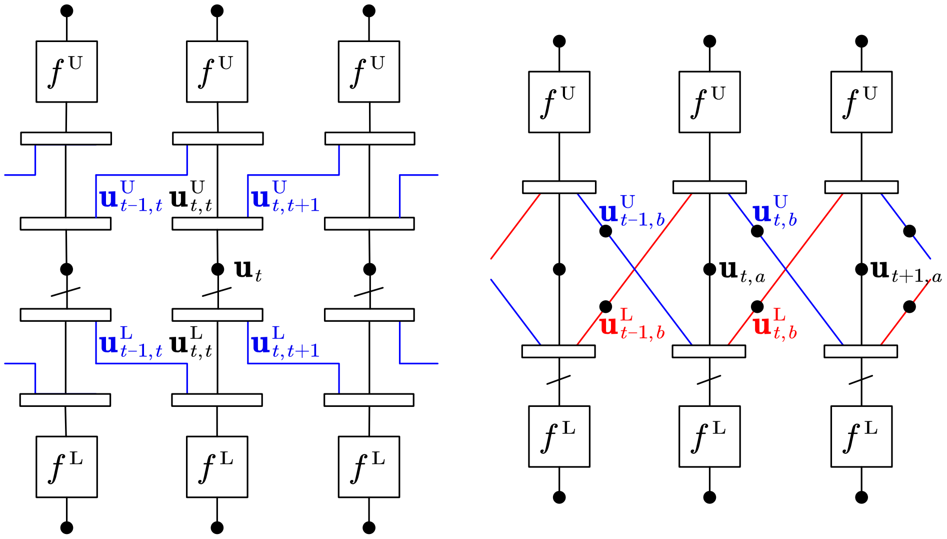
(a) $R = \frac{1}{2}$

(b) $R = \frac{1}{3}$

Figure 6.4: Error performance of GSC-PCCs over the BEC.

6.4.1 Design of Coupling Indices

In this section, we provide an effective method to choose the coupling indices to further enhance the error performance of the GSC-PCCs under finite block length. To be precise, coupling indices design refers how we decompose $\mathbf{u}'_t$ into subsequences $\mathbf{u}^{\mathrm{U}}_{t,t}, \ldots, \mathbf{u}^{\mathrm{U}}_{t,t+m}$ and decompose $\Pi(\mathbf{u}'_t)$ into $\mathbf{u}^{\mathrm{L}}_{t,t}, \ldots, \mathbf{u}^{\mathrm{L}}_{t,t+m}$.


f^{U}
f^{L}
$\mathbf{u}^{\mathrm{U}}_{t-1,t}$
$\mathbf{u}^{\mathrm{U}}_{t,t}$
$\mathbf{u}^{\mathrm{U}}_{t,t+1}$
$\mathbf{u}_t$
$\mathbf{u}^{\mathrm{L}}_{t-1,t}$
$\mathbf{u}^{\mathrm{L}}_{t,t}$
$\mathbf{u}^{\mathrm{L}}_{t,t+1}$
$\mathbf{u}^{\mathrm{U}}_{t-1,b}$
$\mathbf{u}^{\mathrm{U}}_{t,b}$
$\mathbf{u}_{t,a}$
$\mathbf{u}_{t+1,a}$
$\mathbf{u}^{\mathrm{L}}_{t-1,b}$
$\mathbf{u}^{\mathrm{L}}_{t,b}$

(a) (b)

Figure 6.5: Two equivalent compact graphs of SC-PCCs with $m = 1$.

Equivalent Compact Graph of SC-PCCs

Before we explain the design of coupling indices for GSC-PCCs, we first have a look at SC-PCCs (the GSC-PCCs with $q = 1$) with coupling memory $m = 1$. Fig. 6.5a depicts the compact graph of SC-PCCs given in [38]. We first consider time t. Note that $\mathbf{u}'_t = \mathbf{u}_t$ when $q = 1$. When we randomly decompose $\mathbf{u}'_t$ and $\Pi(\mathbf{u}'_t)$, there exists bits which belong to both $\mathbf{u}^{\mathrm{U}}_{t,t}$ and $\mathbf{u}^{\mathrm{L}}_{t,t}$ and are only encoded by the upper and lower encoder at time t. We call them the uncoupled bits of time t and denote them as $\mathbf{u}_{t,a,0} = \mathbf{u}^{\mathrm{U}}_{t,t} \cap \mathbf{u}^{\mathrm{L}}_{t,t}$.

Likewise, there exists bits which belong to both $\mathbf{u}^{\mathrm{U}}_{t,t+1}$ and $\mathbf{u}^{\mathrm{L}}_{t,t+1}$, denoted by $\mathbf{u}_{t+1,a,1} = \mathbf{u}^{\mathrm{U}}_{t,t+1} \cap \mathbf{u}^{\mathrm{L}}_{t,t+1}$. Effectively, such bits are only encoded by the upper and lower encoder at time $t+1$. Hence, we call them the uncoupled bits of time $t+1$ even though they are decomposed from $\mathbf{u}'_t$. For the same reason, we can tell that there is another set of uncoupled bits of time t, denoted by $\mathbf{u}_{t,a,1} = \mathbf{u}^{\mathrm{U}}_{t-1,t} \cap \mathbf{u}^{\mathrm{L}}_{t-1,t}$, which is decomposed from $\mathbf{u}'_{t-1}$. In summary, the set of uncoupled bits of time t is $\mathbf{u}_{t,a} = [\mathbf{u}_{t,a,0}, \mathbf{u}_{t,a,1}]$.

Then, we can classify the rest of the bits in $\mathbf{u}'_t$ into two subsets. One is the set of coupled bits shared between the upper encoder at time t and the lower encoder at time $t+1$, denoted by $\mathbf{u}^{\mathrm{U}}_{t,b} = \mathbf{u}^{\mathrm{U}}_{t,t} \cap \mathbf{u}^{\mathrm{L}}_{t,t+1}$. The other is the set of coupled bits shared between the lower encoder at time t and the upper encoder at time $t+1$, denoted by $\mathbf{u}^{\mathrm{L}}_{t,b} = \mathbf{u}^{\mathrm{L}}_{t,t} \cap \mathbf{u}^{\mathrm{U}}_{t,t+1}$.

With these new notations, we can rewrite the encoding process when the coupling indices are randomly selected. At time t, the input to the upper encoder is $[\mathbf{u}^{\mathrm{L}}_{t-1,b}, \mathbf{u}_{t,a}, \mathbf{u}^{\mathrm{U}}_{t,b}]$, while the input to the lower encoder is $[\mathbf{u}^{\mathrm{U}}_{t-1,b}, \mathbf{u}_{t,a}, \mathbf{u}^{\mathrm{L}}_{t,b}]$. The equivalent compact graph corresponding to the new notations are depicted in Fig. 6.5b. On average, the input sequence of the upper/lower encoder at time t consists of $\frac{1}{2}$ uncouple bits, $\frac{1}{4}$ coupled bits shared with time $t-1$, and $\frac{1}{4}$ coupled bits shared with time $t+1$.

Additionally, we can change the proportion of coupled bits in the input sequence by carefully choosing the coupling indices. For example, by setting $\mathbf{u}^{\mathrm{L}}_{t,t} = [u_i : u_i \in \mathbf{u}'_t, u_i \notin \mathbf{u}^{\mathrm{U}}_{t,t}]$ and $\mathbf{u}^{\mathrm{L}}_{t,t+1} = [u_i : u_i \in \mathbf{u}'_t, u_i \notin \mathbf{u}^{\mathrm{U}}_{t,t+1}]$ for each time t, the proportion of coupled bits becomes 1.

Design of Coupling Indices for GSC-PCCs

We now consider the GSC-PCCs with randomly selected coupling indices. At time t, around half of the bits in $\mathbf{u}_{t,\mathrm{o}}$ are encoded by the upper and lower encoders at the same time instance. Meanwhile, around $\frac{1}{2q^2}$ of the bits in $\mathbf{u}_{t,\mathrm{r}}$ (i.e., the information bits to be repeated) and their $q-1$ replicas are encoded by the upper and lower encoders at the same time instance. These uncoupled bits cannot benefit from coupling as no extrinsic information from the component codewords at other time instances can be obtained. To avoid the uncoupled bits, we choose the coupling indices such that each bit in $\mathbf{u}_{t,\mathrm{r}}$ and its replicas will be encoded by the encoders

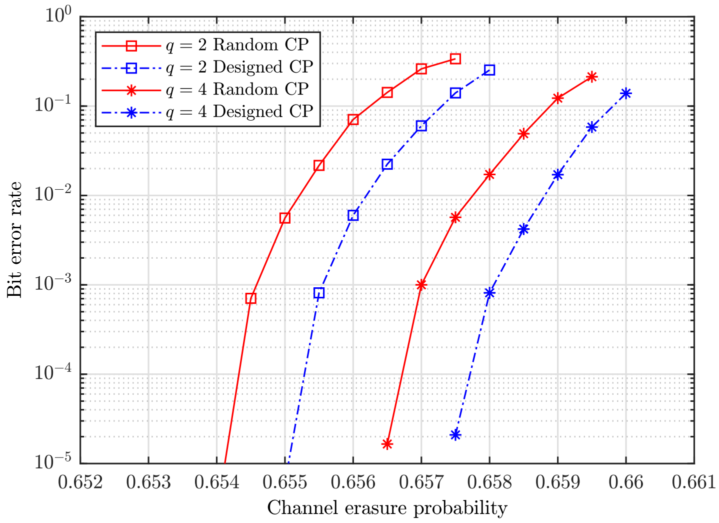
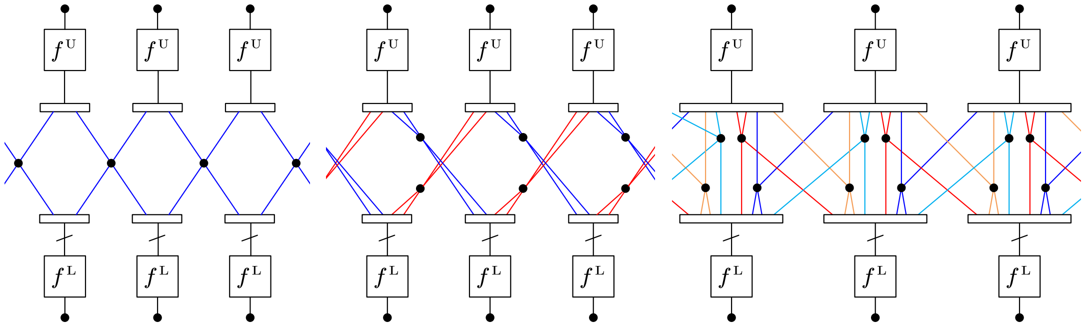


Figure 6.6: Error performance of rate-$\frac{1}{3}$ GSC-PCCs with designed coupling indices. $K=10^4$ and $L=100$.

at different time instances.

In Fig. 6.6, we plot the performance of GSC-PCCs with properly selected coupling indices which avoid uncoupled bits (labeled as “Designed CP”) and that of GSC-PCCs with randomly selected coupling indices (“labeled as “Random CP”). For both $q=2$ and $q=4$, we observe significant performance improvement when the designed coupling indices are used.

Apart from the above observations, we also note that the GSC-PCCs with $q \geq 2$ consist of multiple coupling methods. For example, the GSC-PCCs with $q=2$, $\lambda=0.5$, and $m=1$ consist of three coupling methods, whose compact graphs are shown in Fig. 6.7, respectively. The proportion of these coupling methods in GSC-PCCs can be adjusted via modifying the coupling indices. By deriving the DE equations for each coupling method individually, we find that they have different decoding thresholds. This suggests that we might further improve the finite-length performance of GSC-PCCs by adjusting the proportion of different coupling methods, which can be done by carefully choosing coupling indices.



(a) Type-1 (b) Type-2 (c) Type-3

Figure 6.7: Equivalent compact graph for GSC-PCCs with $q = 2$, $\lambda = 0.5$, and $m = 1$.

6.5 Conclusion

In this chapter, we introduced generalized spatially-coupled parallel concatenated codes, which are generalizations of the conventional SC-PCCs and also have a similar structure to PIC-TCs. We derived the density evolution equations for the newly proposed codes. We searched the maximum decoding thresholds via optimizing the fraction of repeated information bits. Simulation results under finite blocklength shown that our codes outperform some existing class of spatially-coupled codes built with the same component PCC. Additionally, an effective method of selecting coupled bits to further enhance the finite-length decoding performance was provided.

Chapter 7

Introduction to Polar Codes

In this chapter, we present the background on polar codes, including the construction of polar codes, various decoding algorithms, and performance analysis tools. The materials in this chapter are summarized from the literature and will be used as necessary background for the work throughout Chapters 8 – 9.

7.1 Notations and Definitions

We list the relevant notations and definitions in the following.

W	A binary-input discrete memoryless channel (B-DMC)
$W(y\|x)$	Transition probability of W with input x and output y
W_N	A vector channel consisting N subchannels
$W_N^{(i)}$	The i-th subchannel of W_N
$I(W)$	Symmetric capacity of W
x_1^N	A row vector $[x_1, \ldots, x_N]$
x_i^j	A subvector $[x_i, x_{i+1}, \ldots, x_j]$ with $i < j$. When $i = j$, $x_i^j = x_i$. When $i > j$, $x_i^j = \emptyset$.

$x_{\mathcal{A}}$	A subvector of x_1^N consisting of elements with indices in $\mathcal{A}$, and $\mathcal{A} \subset \{1, 2, \ldots, N\}$
$u_{1,o}^N$	A subvector of u_1^N consisting of all the elements with odd indices.
$u_{1,e}^N$	A subvector of u_1^N consisting of all the elements with even indices.
$\oplus$	Modulo-two addition
$\mathbf{A} \otimes \mathbf{B}$	Kronecker product of two matrices $\mathbf{A}$ and $\mathbf{B}$
$\mathbf{A}^{\otimes n}$	n-fold Kronecker power of matrix $\mathbf{A}$
$\mathbf{G}_N$	A matrix of size $N \times N$
$\mathbf{G}_{\mathcal{A}}$	A submatrix of $\mathbf{G}_N$ consisting of rows with indices in $\mathcal{A}$ and $\mathcal{A} \subset \{1, 2, \ldots, N\}$
$\mathbf{G}_{\mathcal{A}\mathcal{A}'}$	A submatrix of $\mathbf{G}_N$ consisting of elements $g_{i,j}$ with $i \in \mathcal{A}$, $j \in \mathcal{A}'$, and $\mathcal{A}, \mathcal{A}' \subset \{1, 2, \ldots, N\}$

Let $W : \mathcal{X} \to \mathcal{Y}$ represent a B-DMC with input alphabet $\mathcal{X} = \{0, 1\}$ and output alphabet $\mathcal{Y}$, and $W(y|x)$ denote the transition probability that $y \in \mathcal{Y}$ is received when $x \in \mathcal{X}$ is transmitted. The *symmetric capacity* of W is defined as

$$I(W) \triangleq \sum_{y \in \mathcal{Y}} \sum_{x \in \mathcal{X}} \frac{1}{2} W(y|x) \log \frac{W(y|x)}{\frac{1}{2}W(y|0) + \frac{1}{2}W(y|1)}, \tag{7.1}$$

which is the maximum code rate for reliable transmission when $\Pr\{x = 0\} = \Pr\{x = 1\} = \frac{1}{2}$. In the field of information theory, $I(W)$ is the mutual information between the input and output of channel W. For a symmetric channel, $I(W)$ equals the capacity of W. In this thesis, we only consider symmetric channels. Hence, for the convenience of presentation, we will call $I(W)$ the capacity of W

in Chapters 7 – 9.

7.2 Channel Polarization

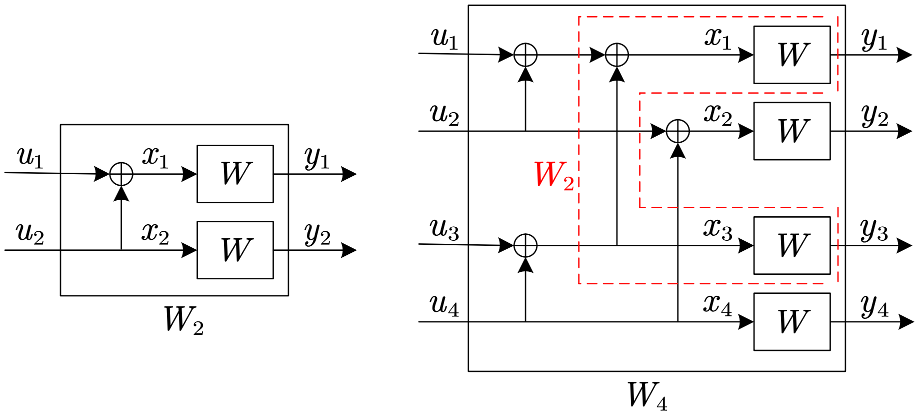
We now review the concept of *channel polarization* introduced in [27]. Channel polarization is a two-phase operation consisting of the *channel combining* phase and the *channel splitting* phase. With this operation, a set of N subchannels $\{W_N^{(i)} : 1 \leq i \leq N\}$ can be constructed from N independent copies of a B-DMC W. These subchannels are referred to as *bit-channels*. As N increases, $I(W_N^{(i)})$ converges to either 1 or 0. This phenomenon is referred to as the *polarization phenomenon*. By utilizing the channel polarization, polar codes achieve the symmetric capacity of any B-DMC.

7.2.1 Channel Combining

Channel combining is to combine N independent copies of W recursively and produce a vector channel $W_N : \mathcal{X}^N \to \mathcal{Y}^N$. The recursion begins with $n = 0$ and $W_1 \triangleq W$. The first level of recursion combines two independent copies of W. As shown in Fig. 7.1a, a vector channel $W_2 : \mathcal{U}^2 \to \mathcal{Y}^2$ is constructed by performing linear combinations

$$\begin{cases} x_1 = u_1 \oplus u_2, \\ x_2 = u_2. \end{cases} \tag{7.2}$$

This linear combination is the basic construction unit of a polar code and is called the kernel of polar code. Let $\mathbf{F} \triangleq \left[\begin{smallmatrix} 1 & 0 \\ 1 & 1 \end{smallmatrix}\right]$ be the kernel matrix, the matrix representation for (7.2) is $x_1^2 = u_1^2 \cdot \mathbf{F}$. In the next level of recursion, as shown in Fig. 7.1b, two independent copies of W_2 are combined to obtain channel W_4.



(a) (b)

Figure 7.1: Channel combining to obtain (a) W_2 and (b) W_4.

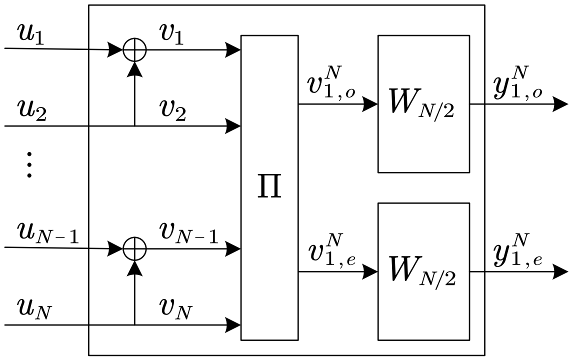


Figure 7.2: General representation of recursive channel combining.

The matrix representation for the linear combinations for constructing W_4 is $x_1^4 = u_1^4 \cdot \mathbf{F} \otimes \mathbf{F} = u_1^4 \cdot \mathbf{F}^{\otimes 2}$.

The general from for the n-th level of recursion is depicted in Fig. 7.2, where W_N is constructed by combining two independent copies of $W_{N/2}$. To be specific, u_1^N is the input vector of W_N, while $v_{1,o}^N$ and $v_{1,e}^N$ are the input vectors of the two copies of $W_{N/2}$, respectively. They are related by $v_{1,o}^N = u_{1,o}^N \oplus u_{1,e}^N$ and $v_{1,e}^N = u_{1,e}^N$. The matrix representation for such recursion is

$$x_1^N = u_1^N \cdot \mathbf{F} \otimes \mathbf{F}^{\otimes n-1} = u_1^N \cdot \mathbf{F}^{\otimes n}. \tag{7.3}$$

7.2.2 Channel Splitting

Channel splitting is to split the vector channel W_N back into N coordinate bit-channels. The i-th bit-channel is $W_N^{(i)} : \mathcal{X} \rightarrow \{\mathcal{Y}^N, \mathcal{X}^{i-1}\}$ with $1 \leq i \leq N$, and its transition probability is $W_N^{(i)}(y_1^N, u_1^{i-1}|u_i)$.

Consider splitting W_2 into $W_2^{(1)} : \mathcal{U}_1 \rightarrow \{\mathcal{Y}_1, \mathcal{Y}_2\}$ and $W_2^{(2)} : \mathcal{U}_2 \rightarrow \{\mathcal{Y}_1, \mathcal{Y}_2, \mathcal{U}_1\}$. The transition probabilities of these two bit-channels are

$$W_2^{(1)}(y_1^2|u_1) = \frac{1}{2}\sum_{u_2} W(y_1|u_1 \oplus u_2) \cdot W(y_2|u_2) \tag{7.4a}$$

$$W_2^{(2)}(y_1, y_2, u_1|u_2) = \frac{1}{2} W(y_1|u_1 \oplus u_2) \cdot W(y_2|u_2) \tag{7.4b}$$

Consider the general expression for splitting $W_N^{(i)}$ from W_N. When i is odd, the transition probability is

$$\begin{aligned} & W_N^{(i)}(y_1^N, u_1^{i-1}|u_i) \\ & \quad = \frac{1}{2}\sum_{u_{i+1}} W_{N/2}^{(i)}(y_{1,o}^N, u_{1,o}^{i-1} \oplus u_{1,e}^{i-1}|u_i \oplus u_{i+1}) \cdot W_{N/2}^{(i)}(y_{1,e}^N, u_{1,e}^{i-1}|u_{i+1}) \end{aligned} \tag{7.5a}$$

When i is even, the transition probability is

$$\begin{aligned} & W_N^{(i)}(y_1^N, u_1^{i-1}|u_i) \\ & \quad = \frac{1}{2} \cdot W_{N/2}^{(i)}(y_{1,o}^N, u_{1,o}^{i-2} \oplus u_{1,e}^{i-2}|u_{i-1} \oplus u_i) \cdot W_{N/2}^{(i)}(y_{1,e}^N, u_{1,e}^{i-2}|u_i) \end{aligned} \tag{7.5b}$$

7.2.3 Polarization of Capacity

During the channel polarization process, the total capacity is constant, while the capacities of bit-channels move away from $I(W)$ following

$$I(W_N^{(2i-1)}) + I(W_N^{(2i)}) = 2I(W_{N/2}^{(i)}), \tag{7.6a}$$

$$I(W_N^{(2i-1)}) \leq I(W_{N/2}^{(i)}) \leq I(W_N^{(2i)}). \tag{7.6b}$$

When the level of recursion increases, $I(W_N^{(i)})$ converges to either 0 or 1. We call the bit-channel whose capacity approaches 1 the capacity-achieving bit-channel. When N approaches infinity, the ratio between the number of capacity-achieving bit-channels and N approaches $I(W)$. Consequently, polar codes with infinite code length can approach the channel capacity.

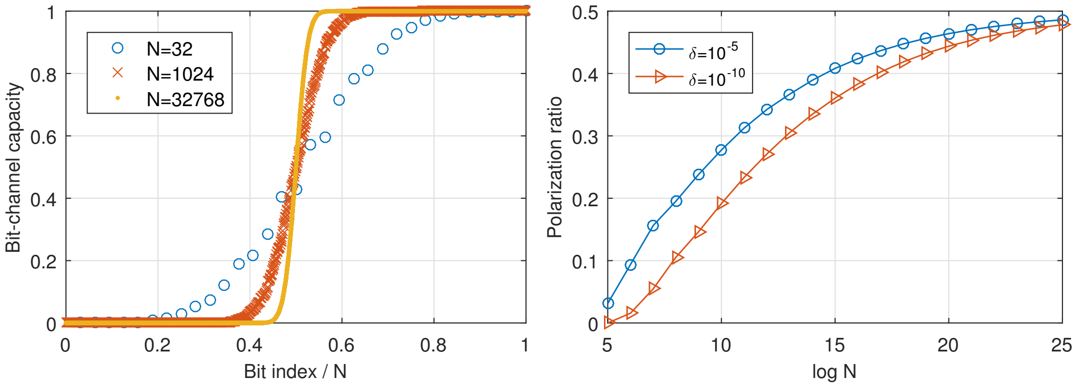
Example 7.1. To visualize the polarization phenomenon, we show the bit-channel capacities for various codeword lengths N in Fig. 7.3a over a BEC with erasure probability $\epsilon = 0.5$. Furthermore, we plot the polarization ratio against N in Fig. 7.3b. Given a threshold δ, the polarization ratio α_δ is computed as

$$\alpha_\delta = \frac{\|\{i : 1 - I(W_N^{(i)}) \leq \delta\}\|}{N}. \tag{7.7}$$

From these figures, we can see that the polarization ratio grows with N. ■

7.3 Polar Codes

Polar codes take advantage of the polarization phenomenon by transmitting information through the bit-channels with high capacity and leaving bit-channels with low capacity unused (i.e., frozen). We call these two types of bit-channels the information channels and the frozen channels, respectively. The indices



(a) Bit-channel capacity (b) Polarization ratio

Figure 7.3: Bit-channel polarization over BEC.

of information channels are called the information indices and are collectively denoted by $\mathcal{A}$, while the indices of frozen channels (frozen indices) are collectively denoted by $\mathcal{A}^c$.

In this section, we first present the encoding and decoding of polar codes. Then, we study the density evolution analysis. Lastly, the design of information indices is reviewed.

7.3.1 Encoding

The encoding of polar codes is to perform channel combining. The matrix representation of polar encoding is

$$x_1^N = u_1^N \cdot \mathbf{G}_N = u_1^N \cdot \mathbf{F}^{\otimes n} \tag{7.8}$$

where N is the codeword length, and $\mathbf{G}_N \triangleq \mathbf{F}^{\otimes n}$ is called polarization matrix, and $\mathbf{F} \triangleq \left[\begin{smallmatrix} 1 & 0 \\ 1 & 1 \end{smallmatrix}\right]$ is the kernel matrix. Each row of $\mathbf{G}_N$ represents a bit-channel of W_N. Let $\mathcal{A} \subset \{1, \dots, N\}$ be the information indices, and $\mathcal{A}^c = \{1, \dots, N\} \setminus \mathcal{A}$

be the frozen indices such that $u_{\mathcal{A}^c} = 0$, (7.8) can be rewritten as

$$x_1^N = u_{\mathcal{A}} \cdot \mathbf{G}_{\mathcal{A}}, \tag{7.9}$$

meaning that bit-channels $\{W_N^{(i)} : i \in \mathcal{A}^c\}$ are used to transmit information bits. The code rate of polar encoder is $R = \frac{K}{N}$, where $K = \|\mathcal{A}\|$ is the information length. We use $\mathscr{P}(N, K)$ to represent a polar code with information length K, and codeword length N.

Reed-Muller Codes

In [27, 105], it was shown that the generator matrices of length-N Reed-Muller (RM) codes are the submatrices of the polarization matrix $\mathbf{G}_N$. In particular, the order-k RM code is constructed by choosing $\mathcal{A} = \{i : w(\mathbf{g}_i) \geq 2^{n-k+1}\}$, where $\mathbf{g}_i$ is the i-th row of $\mathbf{G}_N$ and $w(\mathbf{g}_i)$ is the Hamming weight of $\mathbf{g}_i$. The minimum Hamming distance of the order-k RM code is 2^{n-k+1} and its code rate is

$$R = \frac{\sum_{i=k}^{n} \binom{n}{x}}{N}. \tag{7.10}$$

7.3.2 Decoding

Successive Cancellation Decoding

As the reverse operation of encoding, *successive cancellation* (SC) decoding is to perform channel splitting [27]. For a received signal y_1^N, the SC decoder outputs the hard-decision estimation $\hat{u}_1^N$ following

$$\hat{u}_i = \begin{cases} 0 & \text{if } i \in \mathcal{A}^c \\ h_i(y_1^N, \hat{u}_1^{i-1}) & \text{if } i \in \mathcal{A} \end{cases}, \tag{7.11a}$$

where $h_i(y_1^N, \hat{u}_1^{i-1})$ is the decision function defined as

$$h_i(y_1^N, \hat{u}_1^{i-1}) = \begin{cases} 0 & \text{if } \dfrac{W_N^{(i)}(y_1^N, \hat{u}_1^{i-1}|0)}{W_N^{(i)}(y_1^N, \hat{u}_1^{i-1}|1)} \geq 1 \\ 1 & \text{otherwise} \end{cases}, \tag{7.11b}$$

while the path metrics $W(y_1^N, \hat{u}_1^{i-1}|0)$ and $W(y_1^N, \hat{u}_1^{i-1}|1)$ are computed using (7.5). To be specific, the decoder needs to split $W_2^{(b_1)}, W_4^{(b_2)}, \ldots, W_N^{(b_n)}$ successively to determine $\hat{u}_i$, where $b_j = \left\lceil i \cdot \frac{2^j}{N} \right\rceil$ for $1 \leq j \leq n$. The complexity of SC decoding is $O(N \log_2 N)$.

The SC decoding can be described as the process to search for a correct path $\hat{u}_1^N$ in the *binary decoding tree* $\{0,1\}^K$ given a noisy observation y_1^N of the origin path u_1^N. In the i-th intermediate decoding stage, the decoder makes a decision on $\hat{u}_i = a, a \in \{0,1\}$, and all paths consisting $\hat{u}_i \neq a$ will be excluded from the searching scope for the rest of the decoding stage.

Due to the polarization phenomenon, SC decoding would give the optimal decoding performance under infinite blocklength. However, SC decoding suffers from the performance lost under finite blocklength because the bit-channels are partially polarized. To improve the finite-length performance of polar codes, various of SC-based decoding algorithms are proposed, including SC list (SCL) decoding [28], CRC-aided SC list (CASCL) decoding [28], SC bit-flipping decoding [106,107], SC stack decoding [108], Fano decoding [109], etc. In the following, we review the two most representative algorithms: SCL decoding and CASCL decoding.

Successive Cancellation List Decoding

The SCL decoding is proposed in [28]. It has been shown that SCL decoding can greatly improve the error performance of polar codes under finite blocklength. Unlike SC decoding, which only tracks one path in each intermediate decoding stage of the decoding tree, SCL decoding reserves up to L paths with the highest transition probabilities $W_N^{(i)}(y_1^N, \hat{u}_1^{i-1}|\hat{u}_i)$ in each intermediate decoding stage. Here, L is called the list size of the SCL decoder. In the last decoding stage, one path with the highest transition probability is determined as the decoder output. The complexity of SCL decoding is $O(LN \log N)$.

SCL decoding can be seen as a generalization of SC decoding. When $L = 1$, an SCL decoder becomes an SC decoder. When $L = 2^K$, an SCL decoder becomes an ML decoder.

CRC-aided Successive Cancellation List Decoding

In [28], the authors also proposed the CRC-aided polar codes and the associated CASCL decoding. CRC-aided polar codes are constructed by appending a few CRC-bits to the information sequence before polar encoding. To conduct CASCL decoding, SCL decoding is conducted first. In the last stage of SCL decoding, L surviving paths with the highest transition probabilities are produced. Then, CRC error detection is applied on the surviving paths in descending order with respect to the transition probabilities. The first path with no error detected is declared as the decoder output. If all the paths fail in the error detection, the path with the highest transition probability is considered as the decoder output.

There have been many works shown that polar codes under CASCL decoding have superior errors performance and can outperform various well-designed LDPC codes under finite block lengths. In various works, the performance under CASCL decoding is commonly used as a performance benchmark of polar codes.

BP-Based Decoding

Apart from SC-based decoding, another class of decoding approach is the BP-based decoding, which treats polar codes as a special class of LDPC codes. The decoding algorithms falling into this category include BP flooding [110], soft-cancellation [111], BP list decoding [112, 113], etc. Although BP-based decoding may not have as good error performance as CASCL decoding, it is widely studied due to high parallelism which brings high throughput and low latency.

7.3.3 Density Evolution Analysis

We now present the density evolution (DE) analysis of polar codes, which is used to design the information indices and to approximate the block error rate (BLER) of a polar code under SC decoding.

Factor Graph

Before we go to the details of DE analysis, we first introduce the factor graph of polar codes and some relevant notations. Fig. 7.4 shows the factor graph of the polarization kernel $\mathbf{F}$. The factor graph has two layers, the two VNs (x_1 and x_2) at layer-0 represent the bits transmitted via two copies of W, while the two VNs (u_1 and u_2) at layer-1 represent the bits transmitted via bit-channels $W_2^{(1)}$ and $W_2^{(2)}$, respectively. The Z-shaped VN-CN connection at layer-1 represents the linear combination $x_1^2 = u_1^2 \cdot \mathbf{F}$. Its corresponding channel transform is written as $(W, W) \to (W_1^{(1)}, W_2^{(2)})$.

The factor graph of length-N polar codes has $n + 1$ layers, where $n = \log N$. Let τ denote the layer, and $0 \leq \tau \leq n$. In layer-τ, there are $2^{n-\tau}$ copies

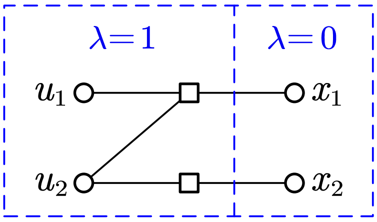


Figure 7.4: Factor graph of a polarization kernel.

of vector channel W_{2^τ}. Each copy of W_{2^τ} consists of 2^τ bit-channels. In the previous sections, we simply use $W_{2^\tau}^{(1)}, \ldots, W_{2^\tau}^{(2^\tau)}$ to represent the bit-channels from one copy of W_{2^τ}. In such notations, it is hard to distinguish the bit-channels from different copies of W_{2^τ}. To make it clear, we use the following bit-channel notations in the rest of this chapter.

- **Bit-channel notations**: Let $W_{2^\tau}^{((i-1)*2^{n-\tau}+j)}$ denote the i-th bit-channel of the j-th copy of W_{2^τ}, where $1 \le i \le 2^\tau$ and $1 \le j \le 2^{n-\tau}$. The channel transforms at layer-τ is written as $(W_{2^{\tau-1}}^{(i)}, W_{2^{\tau-1}}^{(j)}) \to (W_{2^\tau}^{(i)}, W_{2^\tau}^{(j)})$, where $i = (a-1)2^{n-\tau+1} + b$, $j = i + 2^{n-\tau}$, $1 \le a \le 2^{\tau-1}$, and $1 \le b \le 2^{n-\tau}$.

As an example, we plot the factor graph for $\mathscr{P}(8, K)$ in Fig. 7.5. We mark each VN with its corresponding bit-channel using the new notation. In each layer, VNs belonging to the same vector channel are filled with the same color.

DE for BEC

We use ϵ to represent the erasure probability of W, and use $\epsilon_N^{(i)}$ to represent the erasure probability of a bit-channel $W_N^{(i)}$. The evolution of erasure probabilities for channel transform $(W_N^{(i)}, W_N^{(j)}) \to (W_{2N}^{(i)}, W_{2N}^{(j)})$ is

$$\begin{cases} \epsilon_{2N}^{(i)} = \epsilon_N^{(i)} + \epsilon_N^{(j)} - \epsilon_N^{(i)} \epsilon_N^{(j)}, \\ \epsilon_{2N}^{(j)} = \epsilon_N^{(i)} \epsilon_N^{(j)}. \end{cases} \tag{7.12}$$

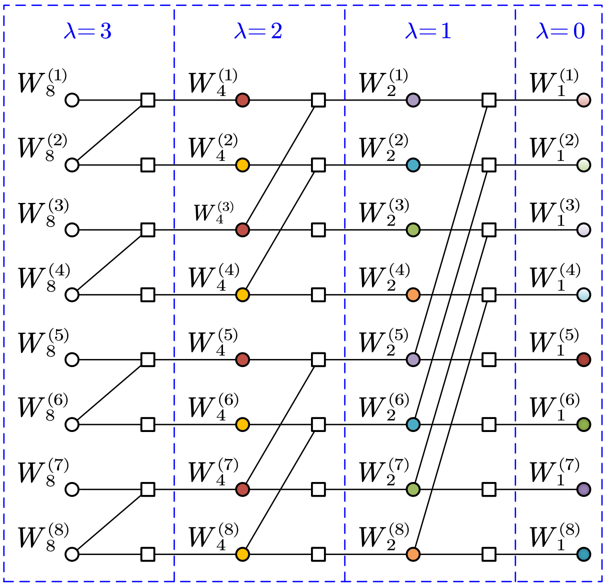


Figure 7.5: Factor graph of the polar code with $N = 8$.

Consider $\mathscr{P}(N, K)$, whose encoder takes input sequence u_1^N and generate codeword x_1^N. At layer-0, $W_1^{(i)}$ corresponds to the i-th coded bits x_i. At layer n, $W_N^{(i)}$ corresponds to the i-th input (information/frozen) bit u_i. The decoding starts from layer 0 and terminates at layer n. To analyze the decoding performance of SC decoder, we initialize $\epsilon_1^{(1)}, \dots, \epsilon_1^{(N)}$ with the erasure probability of the transmission channel. Then, we can track the evolution of erasure probability from layer 0 to layer n with (7.12), and eventually obtain $\epsilon_N^{(1)}, \dots, \epsilon_N^{(N)}$ at layer n. The error probability of u_i is

$$p^{(i)} = \frac{\epsilon_N^{(i)}}{2}. \tag{7.13}$$

GA for AWGN

Instead of digging into the details of DE analysis of polar codes in [114, 115], we review the Gaussian approximation (GA) analysis [116–118], which is a simplified DE analysis for AWGN channels and widely used due to high accuracy.

For an AWGN channel W, let σ^2 denote its noise variance per real dimension. With normalized signal power, the channel SNR is $\frac{E_s}{N_0} = \frac{1}{2\sigma^2}$, where σ^2 is the noise variance per dimension. The LLR density function of W is $\mathcal{N} \sim \{\frac{2}{\sigma^2}, \frac{4}{\sigma^2}\}$. Let μ denote the mean of the LLR density function of W, and $\mu_N^{(i)}$ denote that of $W_N^{(i)}$. For each channel transformation $\{W_N^{(i)}, W_N^{(j)}\} \rightarrow \{W_{2N}^{(i)}, W_{2N}^{(j)}\}$, the evolution of the LLR density function can be described by

$$\begin{cases} \mu_{2N}^{(i)} = f^{-1}\big(1 - \big(1 - f\big(\mu_N^{(i)}\big)\big)\big(1 - f\big(\mu_N^{(j)}\big)\big)\big) \\ \mu_{2N}^{(j)} = \mu_N^{(i)} + \mu_N^{(j)} \end{cases}, \tag{7.14}$$

where $f(x)$ can be approximated by [119]

$$f(x) \approx \begin{cases} 1 & \text{if } x = 0 \\ e^{(-0.4527x^{0.86}+0.0218)} & \text{if } 0 < x \leq 10 \\ \sqrt{\frac{\pi}{x}}\left(1 - \frac{10}{7x}\right) e^{-\frac{x}{4}} & \text{if } x > 10 \end{cases}. \tag{7.15}$$

To evaluate the error performance of $\mathscr{P}(N, K)$. We initialize $\mu_1^{(1)}, \ldots, \mu_1^{(N)}$ with the knowledge on the SNR of the transmission channel, and track the evolution of LLR from layer-0 to layer-n with (7.15) to obtain $\mu_N^{(1)}, \ldots, \mu_N^{(N)}$. The error probability of u_i can then be approximated by [116, 117]

$$p_N^{(i)} \approx Q\left(\sqrt{\mu_N^{(i)}/2}\right). \tag{7.16}$$

Block Error Rate Upper Bound Under SC decoding

Given information indices $\mathcal{A}$, the union bound of block error rate (BLER) of $\mathscr{P}(N, K)$ under SC decoding is

$$P_e \leq 1 - \prod_{i \in \mathcal{A}} (1 - p^{(i)}). \tag{7.17}$$

As this upper bound is tight at a high SNR (or a low erasure rate), it is commonly used as the approximation of BLER. Additionally, it can be used as a loose upper bound for polar codes under SCL decoding or CASCL decoding. Empirically, we can also draw a lower bound for polar codes under SC decoding by using the truncated union bound

$$P \geq 1 - \prod_{i \in \mathcal{M}} (1 - p_N^{(i)}) \tag{7.18}$$

where $\mathcal{M}$ is an indices set satisfies that $p_N^{(i)} > p_N^{(j)}$ for any $i \in \mathcal{M}$ and $j \in \mathcal{A} \setminus \mathcal{M}$, and $\|\mathcal{M}\| \ll K$.

Example 7.2. Fig. 7.6 shows the simulated BLER performance of $\mathscr{P}(1024, 512)$ under SC decoding and its performance bounds. For the truncated union bound, $\|\mathcal{M}\| \in \{10, 20\}$ is considered. ■

7.3.4 Design of Information Indices

As we can tell from (7.17), the design of information indices (also called rate profiling in some literature [109]) greatly affect the error performance of polar codes. Note that the criteria of choosing information indices may change depending on the decoding algorithm used. Generally speaking, there are two major design directions [27]: reliability-based and weight-based.

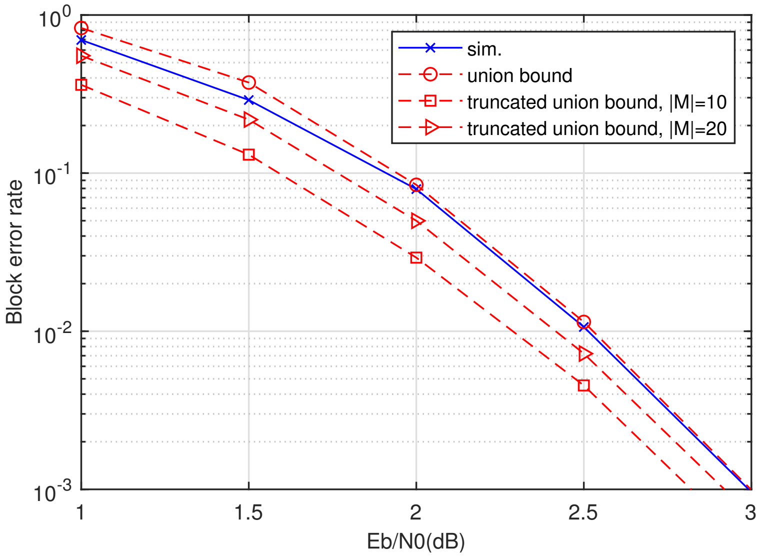


Figure 7.6: Performance bounds of $\mathscr{P}(1024, 512)$ under SC decoding.

The reliability-based design means to take advantage of the polarization phenomenon. The reliability order of bit-channels can be evaluated with different methods. One widely adopted method is to use density evolution to compute the error probability of each bit-channel. Besides, predetermined reliability lists are also studied [13, 120–122]. Most noticeably, 5G standard polar codes [13] use a reliability list to choose information indices for codeword length between 64 and 1024, as well as a large range of code rates. Apart from that, there are also some simulation based method to obtain the reliability order.

The weight-based design means to improve the distance properties of the polarization matrix to achieve good performance for very short codeword lengths, e.g., 256 or less [109, 123]. In particular, as we have reviewed in Sec. 7.3.1, RM codes are constructed with a weight-based design criteria on maximizing the minimum Hamming distance.

There also exist reliability-weight jointly designs [124–126]. A typical case for such design direction is the RM-polar method [124]. Given a weight threshold w_{th}, the information channels are chosen among the bit-channels with weight not lower than w_{th} according to their reliability order.

For consistency, we always use the DE analysis reviewed in Sec. 7.3.3 to design the information indices in the rest of the thesis.

7.4 Systematic Polar Codes

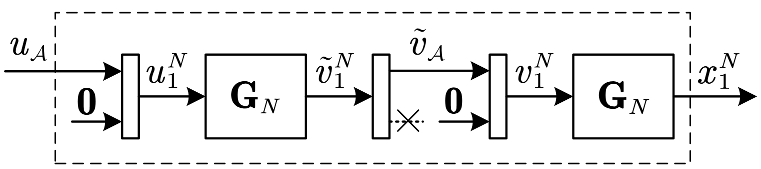
Systematic polar codes were introduced in [127]. Recall that the matrix representation of polar encoding is $x_1^N = u_{\mathcal{A}}\mathbf{G}_{\mathcal{A}}$. Consider that x_1^N is split into two subvectors $x_{\mathcal{A}}$ and $x_{\mathcal{A}^c}$ while $\mathbf{G}_{\mathcal{A}}$ is split into two submatrices $\mathbf{G}_{\mathcal{A}\mathcal{A}}$ and $\mathbf{G}_{\mathcal{A}\mathcal{A}^c}$. The submatrix $\mathbf{G}_{\mathcal{A}\mathcal{A}}$ consists of elements $g_{i,j}$ of $\mathbf{G}_N$ with $i, j \in \mathcal{A}$, while $\mathbf{G}_{\mathcal{A}\mathcal{A}^c}$ consists of elements $g_{i,j}$ of $\mathbf{G}_N$ with $i \in \mathcal{A}$ and $j \in \mathcal{A}^c$, the polar encoding process can be rewritten as

$$\begin{cases} x_{\mathcal{A}} = u_{\mathcal{A}}\mathbf{G}_{\mathcal{A}\mathcal{A}}, \\ x_{\mathcal{A}^c} = u_{\mathcal{A}}\mathbf{G}_{\mathcal{A}\mathcal{A}^c}. \end{cases} \tag{7.19}$$

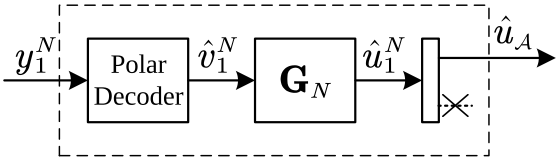
With (7.19), it can be seen that the codeword of a systematic polar code can be obtained by first mapping $u_{\mathcal{A}}$ into $v_{\mathcal{A}} = u_{\mathcal{A}}\mathbf{G}_{\mathcal{A}\mathcal{A}}^{-1}$, and then performing the polar encoding $x_1^N = v_{\mathcal{A}}\mathbf{G}_{\mathcal{A}}$. The resultant systematic codeword is rewritten as

$$\begin{cases} x_{\mathcal{A}} = v_{\mathcal{A}}\mathbf{G}_{\mathcal{A}\mathcal{A}} = u_{\mathcal{A}}\mathbf{G}_{\mathcal{A}\mathcal{A}}^{-1}\mathbf{G}_{\mathcal{A}\mathcal{A}} = u_{\mathcal{A}}\mathbf{I}_K, \\ x_{\mathcal{A}^c} = v_{\mathcal{A}}\mathbf{G}_{\mathcal{A}\mathcal{A}^c} = u_{\mathcal{A}}\mathbf{G}_{\mathcal{A}\mathcal{A}}^{-1}\mathbf{G}_{\mathcal{A}\mathcal{A}^c} = u_{\mathcal{A}}\mathbf{P}, \end{cases} \tag{7.20}$$

where $\mathbf{I}_K$ is the identity matrix of dimension K and $\mathbf{P} = \mathbf{G}_{\mathcal{A}\mathcal{A}}^{-1}\mathbf{G}_{\mathcal{A}\mathcal{A}^c}$ is the parity matrix of size $K \times (N - K)$. Note that the systematic bits $x_{\mathcal{A}}$ are interleaved with the parity bits $x_{\mathcal{A}^c}$, which is different from the conventional definition of systematic code, i.e., the information bits appear first in the output codeword and followed by parity bits. The matrix representation of systematic polar encoding



(a) Encoder



(b) Decoder

Figure 7.7: Systematic polar codes.

is

$$x_1^N = u_{\mathcal{A}} \mathbf{G}_{N,s} \mathbf{\Pi}, \tag{7.21}$$

where $\mathbf{G}_{N,s} = [\mathbf{I}_K \ \mathbf{P}]$ is of size $K \times N$, and $\mathbf{\Pi}$ is a permutation matrix of size $N \times N$ to ensure that $x_{\mathcal{A}} = u_{\mathcal{A}}$.

With the observation that $\mathbf{G}_{\mathcal{A}\mathcal{A}} = \mathbf{G}_{\mathcal{A}\mathcal{A}}^{-1}$, [128] proposed a simple implementation of the systematic encoder, which performs encoding as follows:

- **Systematic polar encoding:**

  1. Let u_1^N be the length-N input vector, where $u_{\mathcal{A}}$ carries the information bits and $u_{\mathcal{A}^c} = \mathbf{0}$.
  2. Compute $\tilde{v}_1^N = u_1^N \mathbf{G}_N$, and split $\tilde{v}_1^N$ into subvectors $\tilde{v}_{\mathcal{A}}$ and $\tilde{v}_{\mathcal{A}^c}$.
  3. Let v_1^N be a length-N vector. Let $v_{\mathcal{A}} = \tilde{v}_{\mathcal{A}}$ and $v_{\mathcal{A}^c} = \mathbf{0}$.
  4. Compute $x_1^N = v_1^N \mathbf{G}_N$.

In Fig. 7.7a, we illustrate the block diagram of the above systematic encoder.

The decoder block diagram of systematic polar codes is illustrated in Fig. 7.7b. As shown in the figure, the receiver inputs the received signal y_1^N and obtain $\hat{v}_1^N$, which is the hard-decision estimation of v_1^N, by using an arbitrary (non-systematic) polar decoding algorithm. Then, the receiver demaps $\hat{v}_1^N$ to $\hat{u}_1^N$ by computing $\hat{u}_1^N = \hat{v}_1^N \mathbf{G}_N$, where $\hat{u}_{\mathcal{A}}$ is the estimation of information bits $u_{\mathcal{A}}$. Knowing that $\hat{v}_{\mathcal{A}^c} = \mathbf{0}$, the demapping can also be expressed by $\hat{u}_{\mathcal{A}} = \hat{v}_{\mathcal{A}} \mathbf{G}_{\mathcal{A}\mathcal{A}}$.

7.5 Multi-Kernel Polar Codes

In the above sections, we have reviewed the classical polar codes constructed with binary kernel $\mathbf{F} \triangleq \left[\begin{smallmatrix} 1 & 0 \\ 1 & 1 \end{smallmatrix}\right]$. Although both theatrical analysis and simulation results have shown its superior error performance, polar codes have a drawback that its codeword length is restricted to $N = 2^n$. To overcome this limitation, various puncturing [129, 130] and shorting [130, 131] techniques have been proposed in the literature. However, a length-N' polar code with $N' < N$ constructed by puncturing or shorting a length-N polar code has high decoding complexity with respect to its codeword length N', because its decoding complexity is the same as the decoding complexity of its length-N mother code. Besides, puncturing and shorting may affect the polarization speed and consequently lead to a substantial performance loss.

To overcome the limitation on codeword length, multi-kernel polar codes [132–134] constructed by combining kernels of different sizes have been proposed in the literature. Let $\mathbf{F}_p$ represent a $p \times p$ kernel. In general, the polarization matrix of a multi-kernel polar code is given by

$$\mathbf{G}_N = \mathbf{F}_{p_1} \otimes \mathbf{F}_{p_2} \otimes \ldots \otimes \mathbf{F}_{p_n} \tag{7.22}$$

with $N = \prod_{\tau=1}^{n} p_\tau$. One representative multi-kernel polar code design is to use two kernels $\mathbf{F}$ and $\mathbf{F}_p$, and let $p_1 = p$ and $p_2, \ldots, p_n = 2$. The corresponding polarization matrix is

$$\mathbf{G}_N = \mathbf{F}_p \otimes \mathbf{F} \otimes \ldots \otimes \mathbf{F} = \mathbf{F}_p \otimes \mathbf{F}^{\otimes n-1} \tag{7.23}$$

with $N = p \cdot 2^{n-1}$. In the following, we use an example to show the structure of this type of multi-kernel polar codes.

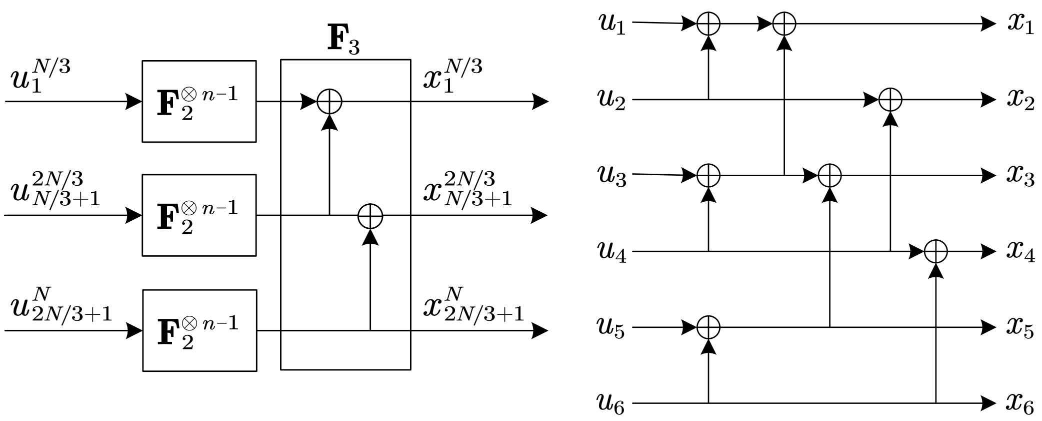
Example 7.3. Consider polarization matrix $\mathbf{G}_N = \mathbf{F}_3 \otimes \mathbf{F}^{\otimes n-1}$ with $\mathbf{F}_3 = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}$ and $N = 3 \cdot 2^{n-1}$. It can be rewritten as a product code

$$\mathbf{G}_N = (\mathbf{I}_3 \otimes \mathbf{F}^{\otimes n-1}) \cdot (\mathbf{G}_3 \otimes \mathbf{I}_{N/3}). \tag{7.24}$$

This expression divides the encoding process $x_1^N = u_1^N \mathbf{G}_N$ into two stages, described as follows:

1. **Stage 1:** parallel encoding with $\mathbf{F}^{\otimes n-1}$. The information sequence u_1^N is segmented into three subsequences $u_1^{N/3}$, $u_{N/3+1}^{2N/3}$, and $u_{2N/3+1}^{N}$. Each subsequence is encoded with $\mathbf{F}^{\otimes n-1}$, giving the coded subsequences $v_1^{N/3}$, $v_{N/3+1}^{2N/3}$, and $v_{2N/3+1}^{N}$ respectively.
2. **Stage 2:** parallel encoding with $\mathbf{F}_3$. The sequence v_1^N is segmented into $\frac{N}{3}$ (i.e., 2^{n-1}) subsequences, and each subsequence is encoded with $\mathbf{F}_3$. In particular, the i-th subsequence is $[v_i, v_{i+N/3}, v_{i+2N/3}]$, giving the coded subsequence $[x_i, x_{i+N/3}, x_{i+2N/3}]$, where $1 \leq i \leq \frac{N}{3}$.

Fig. 7.8a plots the general representation of the above encoding process. To give a simple example, we also plot the encoder for $N = 6$ in Fig. 7.8b, where the



(a) General representation (b) $\mathbf{G}_6$

Figure 7.8: Encoder structure of $\mathbf{G}_N = \mathbf{F}_3 \otimes \mathbf{F}^{\otimes n-1}$.

polarization matrix is given by

$$\mathbf{G}_6 = \mathbf{F}_3 \otimes \mathbf{F} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

■

In [27], it was pointed out that a large size kernel $\mathbf{F}_p$ with $p \geq 2$ may allow for the polarization effect. Later, in [135], the necessary and sufficient conditions for $\mathbf{F}_p$ to allow for the polarization effect are presented. In the same paper, the authors shown that the polarization speed of a polar code can be described by the polarization exponent of its kernels. That is, a larger polarization exponent brings a higher polarization speed. Furthermore, it is shown that the polarization exponent depends on the partial distance spectrum of kernel. A list of kernels with good polarization exponents is presented in [136].

7.6 Summary

In this chapter, we have reviewed the background on polar codes, including encoding, decoding algorithm, and performance analysis. In Chapters 8 – 9, we will present our spatially coupled polar codes design. In Chapter 8, we will introduce partially information coupled (PIC) polar codes. In Chapter 9, we will discuss the design of PIC-polar codes for high-order modulations. In particular, we design the coupling indices by taking the polarization property of finite-length polar codes into consideration.

Chapter 8

Partially Information Coupled Polar Codes

In this chapter, we propose the partially information coupled (PIC) polar codes to improve the transmission efficiency of the transport block (TB) based communication standards. In PIC-polar codes, every two consecutive systematic polar code blocks (CBs) in a TB are coupled by sharing a few systematic information bits. We propose a CB decoder that only uses the information associated with correctly decoded coupled bits to mitigate the serious error propagation problem in successive cancellation based polar code decoding algorithms. For the global decoding, we propose a sliding window decoder with a variable window size to achieve a flexible trade-off between the decoding performance and complexity. We derive a closed-form expression for the TB error rate (TBER) of PIC-polar codes. We further design the coupling scheme based on the derived TBER. Simulation results confirm the effectiveness of our TBER analysis and our coupling scheme design and show that PIC-polar codes significantly outperform the uncoupled polar codes with a slightly increased decoding complexity.

8.1 Introduction

Polar codes [27] are proved to be capacity-achieving for any binary-input discrete memoryless symmetric (B-DMS) channels. In 5G channel coding standards, polar codes were chosen as the coding scheme for control channels [13, 137]. Many works have reported that polar codes outperform some turbo codes and LDPC codes under specific code rates and code lengths [28, 138]. Thus, polar codes are considered as a strong candidate for transmitting long data sequences in future communication standards.

In this chapter, we construct PIC-polar codes [139] by applying the partial information coupling technique to polar codes. Though the code construction is straightforward, it is very challenging to obtain the promised coding gain compared to PIC-turbo codes [93] and PIC-LDPC codes [140]. In addition, the partial polarization problem of finite-length polar codes and the error propagation problem of SC based decoding algorithms ask for an analysis framework to design PIC-polar codes with good TBER performance. We aim to tackle these challenges in this work and the main contributions are summarized as follows.

- We propose a class of PIC-polar codes constructed by sharing a few systematic information bits between every two consecutive systematic polar CBs. A corresponding CB decoding scheme and a global decoding scheme are proposed. Different from PIC-turbo codes and PIC-LDPC codes which always use soft extrinsic information of coupled bits for CB decoding, PIC-polar codes only use the extrinsic information of correctly decoded coupled bits for CB decoding to mitigate the serious error prorogation in SC-based decoding algorithms of polar codes. In addition, the global decoding scheme realizes a sliding window decoder with a variable window size. It provides a flexible trade-off between decoding complexity, latency, and error performance for PIC-polar codes.

- We analyze the error performance of PIC-polar codes. In particular, we first derive the error probability of CB decoding for various decoding conditions. We then derive a closed-form expression and a lower bound for the TBER of PIC-polar codes.

- We design the coupling indices of PIC-polar codes. The coupling indices refer to the location of coupled bits in each CB. Due to the polarization effect, choosing different coupling indices would affect the TBER performance. Specifically, we consider different coupling indices design and use the derived TBER closed-form expression to find design that leads to the best TBER performance. The TBER analysis suggests that coupling the most unreliable bits of consecutive polar CBs provides the best TBER performance.

- We evaluate the TBER performance and the decoding complexity of PIC-polar codes via simulation. The simulation results confirm the effectiveness of our analysis and show that our PIC-polar codes outperform uncoupled polar codes for various code rates at a TBER above 10^{-3} with a slightly increased decoding complexity.

This chapter is organized as follows. In Sec. 8.2, we present the construction of PIC-polar codes by expressing the construction method in both encoding scheme and matrix representation. In Sec. 8.3, we present the decoding of PIC-polar codes. Sec. 8.4 presents the TBER analysis of PIC-polar codes and Sec. 8.5 focuses on the design of coupling indices. Numerical results are shown in Sec. 8.6. Sec. 8.7 draws the conclusion.

8.2 Construction of PIC-polar Codes

This section presents the construction of PIC-polar codes by applying coupling to systematic polar codes. We first present the encoding of PIC-polar codes. We then interpret the construction of PIC-polar codes with matrix representation.

8.2.1 Encoding of PIC-polar Codes

The encoder block diagram of PIC-polar codes is shown in Fig. 8.1. The encoding consists of two stages: 1) TB segmentation and coupling, and 2) polar CB encoding. The detailed processes are as follows.

Stage 1: TB segmentation and coupling

In this stage, a TB is segmented into multiple coupled information subsequences (CBs). This process in shown in Fig. 8.1a.

1. **Information segmentation:** Let $\mathbf{u}$ be the information sequence of a TB, and L be the number of CBs. The sequence $\mathbf{u}$ is segmented into L uncoupled sequences $\{\mathbf{u}_{t,t} : 1 \leq t \leq L\}$ and $L-1$ coupled sequences $\{\mathbf{u}_{t,t+1} : 1 \leq t \leq L-1\}$. The lengths of these two types of sequences are $\|\mathbf{u}_{t,t}\| = K'_u$ and $\|\mathbf{u}_{t,t+1}\| = K_c$, respectively.

2. **Padding bits insertion:** To initialize and terminate the coupling, two all-zero blocks $\mathbf{u}_{0,1}$ and $\mathbf{u}_{L,L+1}$ are attached before $\mathbf{u}_{1,1}$ and after $\mathbf{u}_{L,L}$ respectively. The lengths of both blocks are $\|\mathbf{u}_{0,1}\| = \|\mathbf{u}_{L,L+1}\| = K_c$.

3. **Information coupling:** At time $t = 1, \ldots, L$, the t-th polar CB (CB$_t$) takes $\mathbf{u}_t = [\mathbf{u}_{t-1,t}, \mathbf{u}_{t,t}, \mathbf{u}_{t,t+1}]$ as its information sequence.

4. **CRC attachment:** For CB_t, a K_{crc}-bit CRC $\mathbf{c}_t$ is computed for $\mathbf{u}_t$ and attached to the end $\mathbf{u}_t$.

Stage 2: CB encoding

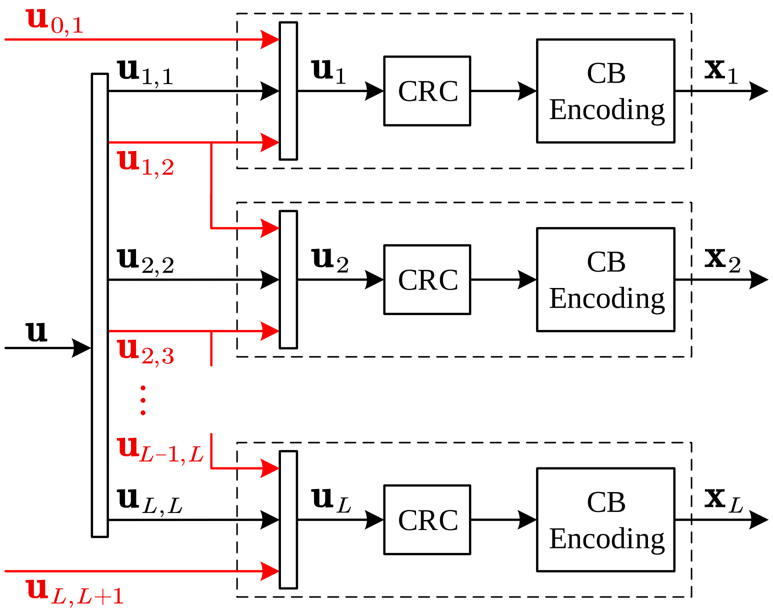
In this stage, each information subsequence $\mathbf{u}_t$ is encoded individually, producing the codeword $\mathbf{x}_t$. The structure of CB encoder is depicted in Fig. 8.1b. The component encoder is a systematic polar encoder, denoted as $\mathbf{G}_{N,s}$, with input vector u_1^N and output vector x_1^N.

1. **Interleaving:** Let $\mathcal{A}$ be the information indices, and $|\mathcal{A}| = K$. Let $\mathcal{B} = \{\mathcal{B}_{\mathrm{H}}, \mathcal{B}_{\mathrm{T}}\} \subset \mathcal{A}$ be the coupling indices, $\mathcal{B}_{\mathrm{H}} \cap \mathcal{B}_{\mathrm{T}} = \emptyset$, and $|\mathcal{B}_{\mathrm{H}}| = |\mathcal{B}_{\mathrm{T}}| = K_c$. The indices sets $\mathcal{B}_{\mathrm{H}}$ and $\mathcal{B}_{\mathrm{T}}$ are the *head coupling indices* and the *tail coupling indices* associated with $\mathbf{u}_{t-1,t}$ and $\mathbf{u}_{t,t+1}$ for CB_t, respectively. At time t, the sequence $\mathbf{u}_t$ is interleaved so that the input vector u_1^N satisfies $u_{\mathcal{B}_{\mathrm{H}}} = \mathbf{u}_{t-1,t}$, $u_{\mathcal{B}_{\mathrm{T}}} = \mathbf{u}_{t,t+1}$, and $u_{\mathcal{A}\setminus\mathcal{B}} = \{\mathbf{u}_{t,t}, \mathbf{c}_t\}$.

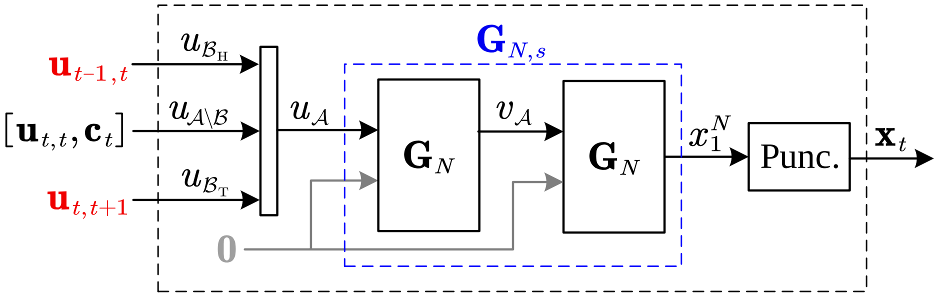
2. **Systematic encoding:** The systematic polar encoder takes input vector u_1^N and produces the coded output vector x_1^N.

3. **Puncturing:** For CB_t, the coded sequence x_1^N satisfies $x_{\mathcal{B}_{\mathrm{H}}} = \mathbf{u}_{t-1,t}$ and $x_{\mathcal{B}_{\mathrm{T}}} = \mathbf{u}_{t,t+1}$. At time $t = 1, \ldots, L-1$, $x_{\mathcal{B}_{\mathrm{H}}}$ is punctured so that the codeword of CB_t is $\mathbf{x}_t = x_1^N \setminus x_{\mathcal{B}_{\mathrm{H}}}$. At time $t = L$, both $x_{\mathcal{B}_{\mathrm{H}}}$ and $x_{\mathcal{B}_{\mathrm{T}}}$ are punctured so that the codeword of CB_L is $\mathbf{x}_L = x_1^N \setminus x_{\mathcal{B}}$.

In this encoder, the padding sequence $\mathbf{u}_{0,1}$ and $\mathbf{u}_{L,L+1}$ are not transmitted while the coupled blocks $\{\mathbf{u}_{t,t+1} : 1 \le t \le L-1\}$ are only transmitted once even though each of them is encoded by twice. The effective code rate of PIC-polar



(a) TB segmentation and coupling.



(b) CB encoding.

Figure 8.1: Encoding scheme of PIC-polar codes.

codes is

$$R = \frac{\sum_{t=1}^{L} \|\mathbf{u}_{t,t}\| + \sum_{t=1}^{L-1} \|\mathbf{u}_{t,t+1}\|}{\sum_{t=1}^{L} \|\mathbf{x}_t\|} = \frac{LK - (L+1)K_c}{LN - (L+1)K_c}, \tag{8.1}$$

where $K = K_u + 2K_c$ is information length of polar encoder, $K_u = K_u' + K_{crc}$ is the length of uncoupled bits of a CB, and $R_0 = \frac{K}{N}$ is the code rate of polar encoder[1]. We define $\lambda \triangleq \frac{K_c}{K}$ as coupling ratio of PIC-polar codes.

Remark 8.1. In encoding stage 2, the information sequence $\mathbf{u}_t$ is interleaved before entering the systematic polar encoder. The interleaving pattern is determined by coupling indices $\mathcal{B}_{\mathrm{H}}$ and $\mathcal{B}_{\mathrm{T}}$. When decoding, the coupling indices decide how many reliable decoding messages can be spread between CBs, and consequently affect the TBER performance of PIC-polar codes. The design of coupling indices will be presented in Sec. 8.5. ◀

PIC-Polar Codes Versus PIC-Turbo Codes

One advantage of PIC-polar codes over PIC-turbo codes is that PIC-polar codes can provide better flexibility on the target code rates. A turbo encoder has a constant code rate R_0. Due to the rate loss brought by partial information coupling, a PIC-turbo code built with the rate-R_0 turbo code has a code rate $R < R_0$. Thus, additional rate-matching techniques are required to construct a PIC-turbo code with $R \geq R_0$. In contrast, a polar encoder supports a large range of code rates, i.e, $\frac{1}{N} \leq R_0 \leq \frac{N-1}{N}$. Given a target code rate R, we can choose a component code rate $R_0 > R$ to ensure that the code rate after coupling is exact R. Thus, PIC-polar codes can reach a large range of code rates without additional rate-matching techniques.

In another aspect, unlike PIC-turbo codes, which can support almost arbitrary blocklengths, PIC-polar codes have a constraint on blocklength. This is because the codeword length of a polar code is restricted to $N = 2^n$. Therefore, PIC-polar codes are less preferred for a system that needs to support a wide range of different blocklengths.

---

[1] We treat CRC bits as a part of information sequence here since CRC is always employed in practical applications for the receiver side error detection. In addition, CRC only introduces a negligible rate loss because CRC is generally much shorter than the information sequence

Another advantage of PIC-polar codes over PIC-turbo codes is that PIC-polar codes do not suffer from error floor. As PIC technique does not modify the structure of the underlying component code, PIC-turbo codes are affected by error floor due to the nature of turbo codes. On the contrary, PIC-polar codes are not affected by error floor because polar codes do not have error floor [141].

8.2.2 Matrix Representation of PIC-polar Codes

As we have reviewed in Sec. 7.4, the generator matrix of a polar code can be written in the systematic form of $\mathbf{G}_{N,s} = [\mathbf{I}_K \ \mathbf{P}]$, where $\mathbf{I}_K$ is the $K \times K$ identity matrix and $\mathbf{P} = \mathbf{G}_{\mathcal{A}\mathcal{A}}^{-1}\mathbf{G}_{\mathcal{A}\mathcal{A}^c}$ is the parity matrix of size $K \times N - K$. The systematic generator matrix can be further rewritten into a form as

$$\mathbf{G}_{N,s} = \begin{bmatrix} \mathbf{I}_{K_c} & 0 & 0 & \mathbf{P}_1 \\ 0 & \mathbf{I}_{K_u} & 0 & \mathbf{P}_2 \\ 0 & 0 & \mathbf{I}_{K_c} & \mathbf{P}_3 \end{bmatrix} \Leftrightarrow \begin{bmatrix} \mathbf{I}_{K_c} & 0 & \mathbf{P}_1 & 0 \\ 0 & \mathbf{I}_{K_u} & \mathbf{P}_2 & 0 \\ 0 & 0 & \mathbf{P}_3 & \mathbf{I}_{K_c} \end{bmatrix} \tag{8.2}$$

In the above expression, $\mathbf{I}_{K_c}$ and $\mathbf{I}_{K_u}$ are identity matrices of size $K_c \times K_c$ and $K_u \times K_u$, $\mathbf{P}_1$ and $\mathbf{P}_3$ are the submatrices of $\mathbf{P}$ of size $K_c \times (N - K)$, and $\mathbf{P}_2$ is the submatrix of $\mathbf{P}$ of size $K_u \times (N - K)$. The operator $\Leftrightarrow$ means that the R.H.S. matrix is obtained by swapping the columns of the L.H.S. matrix. In the R.H.S. matrix of (8.2), the top-left submatrix $\mathbf{I}_{K_c}$ and the bottom-right submatrix $\mathbf{I}_{K_c}$ of the lower matrix are the same. PIC-polar codes can then be constructed by sharing the submatrix $\mathbf{I}_{K_c}$ between consecutive polar CBs. In this view, the encoding of PIC-polar codes is expressed as follows

$$\mathbf{u}\mathbf{G}_{\mathrm{PIC}} = \begin{bmatrix} \vdots \\ \mathbf{u}_{t-1,t} \\ \mathbf{u}_{t,t} \\ \mathbf{u}_{t,t+1} \\ \mathbf{u}_{t+1,t+1} \\ \mathbf{u}_{t+1,t+2} \\ \vdots \end{bmatrix}^{\mathrm{T}} \cdot \begin{bmatrix} \ddots & & & & & & & & \\ & \mathbf{I}_{K_c} & 0 & \mathbf{P}_1 & 0 & & & & \\ & 0 & \mathbf{I}_{K_u} & \mathbf{P}_2 & 0 & & & & \\ & 0 & 0 & \mathbf{P}_3 & \mathbf{I}_{K_c} & 0 & \mathbf{P}_1 & 0 & \\ & & & & 0 & \mathbf{I}_{K_u} & \mathbf{P}_2 & 0 & \\ & & & & 0 & 0 & \mathbf{P}_3 & \mathbf{I}_{K_c} & \\ & & & & & & & & \ddots \end{bmatrix}, \quad (8.3)$$

where $\mathbf{u}$ is an infinite-long input information sequence, and $\mathbf{G}_{PIC}$ is the generator matrix of the PIC-polar code. The subvector $[\mathbf{u}_{t-1,t}, \mathbf{u}_{t,t}, \mathbf{u}_{t,t+1}]$ constitute the information sequence of CB_t, and $[\mathbf{u}_{t,t+1}, \mathbf{u}_{t+1,t+1}, \mathbf{u}_{t+1,t+2}]$ constitute the information sequence of CB_{t+1}. CB_t and CB_{t+1} are coupled via the shared sequence $\mathbf{u}_{t,t+1}$. To initialize and terminate the PIC-polar code, the first and the last (L-th) polar CBs are shortened by setting the first and the last K_c bits in $\mathbf{u}$ to zeros so that $\mathbf{u}_{0,1} = \mathbf{0}$ and $\mathbf{u}_{L,L+1} = \mathbf{0}$.

8.3 Decoding of PIC-polar Codes

In this section, we present the decoding of PIC-polar codes. Generally speaking, each polar CB can be decoded separately, while the global decoding of PIC-polar codes is conducted by passing the extrinsic information of coupled sequences between consecutive CBs. We first present the decoding of a CB. Then, we propose a sliding window decoding scheme to reduce the decoding latency.

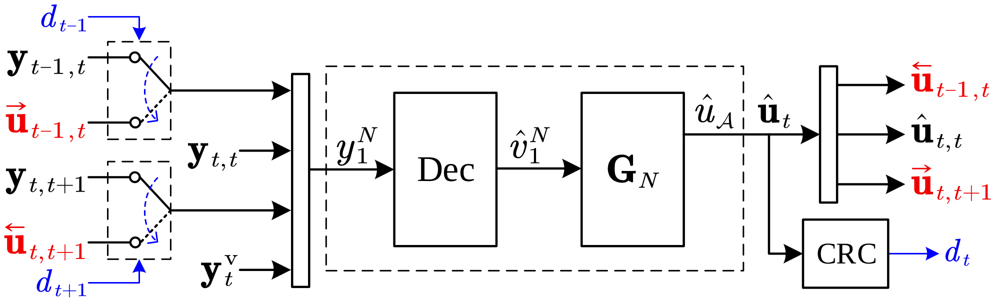

$d_{t\text{-}1}$
$\mathbf{y}_{t\text{-}1,t}$
$\overleftarrow{\mathbf{u}}_{t\text{-}1,t}$
$\mathbf{y}_{t,t+1}$
$\overleftarrow{\mathbf{u}}_{t,t+1}$
d_{t+1}
$\mathbf{y}_{t,t}$
$\mathbf{y}_t^{\mathrm{v}}$
y_1^N
Dec
$\hat{v}_1^N$
$\mathbf{G}_N$
$\hat{u}_{\mathcal{A}}$
$\hat{\mathbf{u}}_t$
$\overleftarrow{\mathbf{u}}_{t\text{-}1,t}$
$\hat{\mathbf{u}}_{t,t}$
$\overrightarrow{\mathbf{u}}_{t,t+1}$
CRC
d_t

Figure 8.2: PIC-polar CB decoding.

8.3.1 Decoding of Polar Code Blocks

Fig. 8.2 depicts the block diagram for decoding CB$_t$. We use $\mathbf{y}_{t-1,t}$, $\mathbf{y}_{t,t}$, and $\mathbf{y}_{t,t+1}$ to represent the channel LLR of head coupled bits $\mathbf{u}_{t-1,t}$, uncoupled bits $\{\mathbf{u}_{t,t}, \mathbf{c}_t\}$, and tail coupled bits $\mathbf{u}_{t,t+1}$, respectively. We use $\mathbf{y}_t^{\mathrm{v}}$ to represent the channel LLR of parity bits $\mathbf{v}_t$. A systematic polar decoder, which is implemented as a serial concatenation of a polar decoder (represented by "Dec" in Fig. 8.2) and a polar encoder (represented by "$\mathbf{G}_N$" in Fig. 8.2), takes in the log-likelihood-ratio (LLR) vector y_1^N and outputs the hard-decision estimation vector $\hat{u}_{\mathcal{A}}$. The hard-decision output $\hat{\mathbf{u}}_t$ consists of $\overleftarrow{\mathbf{u}}_{t-1,t}$, $\hat{\mathbf{u}}_{t,t}$, $\overrightarrow{\mathbf{u}}_{t,t+1}$, and $\hat{\mathbf{c}}_t$, which are the hard-decision estimation of $\mathbf{u}_{t-1,t}$, $\mathbf{u}_{t,t}$, $\mathbf{u}_{t,t+1}$, and $\mathbf{c}_t$, respectively. By performing CRC check against $\hat{\mathbf{u}}_t$, the decoding indicator d_t is produced by

$$\mathbf{d}_t = \begin{cases} 1 & \text{if CRC detects no error,} \\ 0 & \text{if CRC detects errors.} \end{cases} \tag{8.4}$$

For any spatially coupled codes, the decoding of a CB can use the extrinsic information of coupled sequences to improve the decoding performance. In our proposed polar CB decoding scheme, we only use the perfect extrinsic information of coupled sequences which are believed to be decoded correctly[2]. In particular,

---

[2] We assume that the length of the CRC is long enough to provide a sufficiently low undetectable error probability.

the input LLRs $y_{\mathcal{B}_{\mathrm{H}}}$ and $y_{\mathcal{B}_{\mathrm{T}}}$ of the polar decoder are given by

$$y_{\mathcal{B}_{\mathrm{H}}} = \begin{cases} \mathbf{y}_{t-1,t} & \text{if } d_{t-1} = 0 \\ \infty \cdot (1 - 2\overrightarrow{\mathbf{u}}_{t-1,t}) & \text{if } d_{t-1} = 1 \end{cases}, \tag{8.5a}$$

$$y_{\mathcal{B}_{\mathrm{T}}} = \begin{cases} \mathbf{y}_{t,t+1} & \text{if } d_{t+1} = 0 \\ \infty \cdot (1 - 2\overleftarrow{\mathbf{u}}_{t,t+1}) & \text{if } d_{t+1} = 1 \end{cases}, \tag{8.5b}$$

while $y_{\mathcal{A}\setminus\mathcal{B}} = \mathbf{y}_{t,t}$ and $y_{\mathcal{A}^c} = \mathbf{v}_t$. The indicators d_{t-1} and d_{t+1} are produced by (8.4) during the decoding of CB_{t-1} and CB_{t+1}. The extrinsic information $\overrightarrow{\mathbf{u}}_{t-1,t}$ is the hard-decision estimation of $\mathbf{u}_{t-1,t}$ produced during decoding of CB_{t-1}, while $\overleftarrow{\mathbf{u}}_{t,t+1}$ is the hard-decision estimation of $\mathbf{u}_{t,t+1}$ produced during decoding of CB_{t+1}.

Remark 8.2. We only use the perfect extrinsic information of coupled bits for CB decoding. If the perfect extrinsic information is not available, the CB decoder only uses the channel LLRs of the coupled bits as the input. This is motivated by the observation that SC based decoding algorithms are very sensitive to error propagation. By propagating the extrinsic information with errors to a CB, there is a high probability that it will introduce more errors in the target CB.

In addition, our analysis in Sec. 8.5 will suggest that coupling the unreliable bits between consecutive CBs can obtain more coupling gain. This means that when the decoding of a CB fails, it is highly likely that errors would appear in the coupled sequences. Therefore, conservatively exchanging extrinsic information should be more beneficial under this coupling strategy.

Apart from that, we would also like to emphasize that the CB decoder only outputs a single codeword even if list decoding algorithms, e.g., SCL and CASCL, are used therein. Therefore, only the extrinsic information related to this codeword will be exchanged between consecutive CBs. No list will be exchanged between consecutive CBs. ◄

8.3.2 Global Decoding of PIC-polar Codes

Based on the above polar CB decoding scheme, a global decoding scheme is required to exploit the coupled information between consecutive CBs. Generally, several decoding scheduling and decoding schemes can be employed. One is the fully parallel decoding scheme, which decodes all polar CBs simultaneously and then exchanges the extrinsic information of coupled bits for a few iterations. It has a low decoding latency for the whole TB, but suffers from a high decoding complexity. The other is the fully serial decoding scheme (FF-FB), which decodes CBs one by one from the first CB to the last CB (feed-forward decoding) and then proceeds in the opposite direction (feed-back decoding) for several iterations. It has a low decoding complexity but a high decoding latency.

In this section, we propose a new *look-back and go-back* (LB-GB) scheme based on FF-FB scheme to reduce the decoding latency and decoding complexity without sacrificing decoding performance. To see this, consider the case that CB_t was decoded successfully while the decoding of CB_{t-1} failed in the feed-forward (FF) phase. In the feed-back (FB) phase, the perfect extrinsic information $\overleftarrow{\mathbf{u}}_{t-1,t}$ will be used for decoding CB_{t-1} according to (8.4) and (8.5b). Therefore, the decoding results for CB_{t+1} – CB_L are irrelevant to the decoding of CB_{t-1} in the FB phase since $\overleftarrow{\mathbf{u}}_{t-1,t}$ only depends on CB_t. Based on this observation, we propose to immediately go back to decode CB_{t-1} instead of proceeding to CB_{t+1} when the above case happens. This LB-GB decoding process is performed continuously. If there is a decoding failure in the go-back phase, the decoding of the whole TB terminates since any further FF-FB decoding will not provide more information to the CB which cannot be decoded in the go-back phase. This early termination property of the LB-GB scheme saves the computation resources of the receiver and enables an earlier retransmission request. Therefore, it leads to a lower TB transmission latency. We also note that it sacrifices no decoding performance compared to the fully serial decoding scheme since any decoding failures in the go-back phase will also happen in the FB phase of the fully serial decoding scheme.

The LB-GB scheme is described as below:

---

1. **Initialization**: Let $\mathbf{y}_{0,1} = \mathbf{y}_{L,L+1} = \infty$, $[d_1, \dots, d_L] = \mathbf{0}$, $d_0 = 1$, and $M = 0$.

2. **Feed-forward Decoding**: The decoder performs CB decoding from CB$_{M+1}$ to CB$_L$. After the decoding of CB$_t$, the decoder checks the decoding indicators $[d_1, \dots, d_L]$ to decide whether to stay in the FF phase or switch to the FB phase.

   (a) If $[d_1, \dots, d_L] = \mathbf{1}$, go to Step 4.

   (b) If $d_t = 1$ and $d_{t-1} = 0$, update $M = t$, go to Step 3.

   (c) If $t = L$ and $d_L = 0$, exit decoding and declare a TB error.

   (d) Otherwise, continue the FF decoding process.

3. **Feed-back Decoding**: The decoder performs CB decoding from CB$_{M-1}$ to CB$_1$. After the decoding of CB$_t$, the decoder checks the decoding indicators $[d_1, \dots, d_L]$ to decide whether to stay in the FB phase or switch to the FF phase.

   (a) If $[d_1, \dots, d_L] = \mathbf{1}$, go to Step 4.

   (b) If $d_t = 0$, exit decoding and declare a TB error (**early termination**).

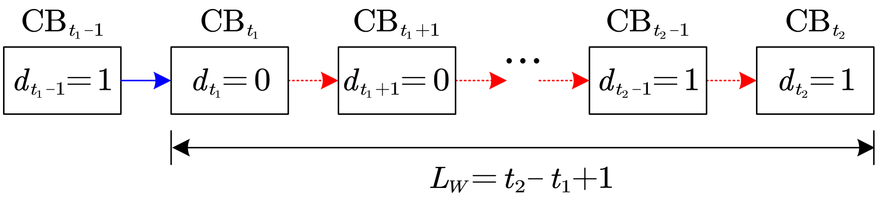


Figure 8.3: Decoding window of the proposed decoding scheme.

(c) If $d_t = 1$ and $d_{t-1} = 1$, go to Step 2.

(d) Otherwise, continue the FB decoding process.

4. **CB concatenation**: The decoder collects and concatenates all the hard-decision estimations, and outputs the estimation of the whole TB.

Remark 8.3. LB-GB scheme can be seen as a windowed decoding scheme with a variable window size. The decoding window starts from the first CB which was not decoded successfully in the FF phase and ends with the CB which has been decoded successfully in the FF phase. An example is shown in Fig. 8.3. The window size in this case is $L_W = t_2 - t_1 + 1$.

In addition, the decoding complexity can be further reduced by limiting the window size to L_W. To be specific, the decoder terminates decoding in the FF phase if the decoding of L_W consecutive CBs has failed. Unless otherwise specified, we always use $L_W = L$ in this chapter. ◀

8.4 Performance Analysis of PIC-polar Codes

In this section, we analyze the TBER performance of the PIC-polar codes under LB-GB scheme. To this end, we first review the Gaussian approximation (GA) [116–119] analysis for polar codes under SC decoding over the AWGN channels. Then, we define four possible error events for the CB decoding of PIC-polar codes.

We also present the methods to obtain the possibilities of these four error events based on the GA analysis for polar codes or via simulations. With these error event possibilities, we derive the CBERs of PIC-polar codes in the FF and FB decoding processes, respectively, and eventually obtain the TBER of PIC-polar codes.

8.4.1 Gaussian Approximation Analysis of Polar Codes

We first recall the GA analysis we have viewed in Sec. 7.3.3. Consider a AWGN channel W. With normalized signal power, we have $\frac{E_s}{N_0} = \frac{1}{2\sigma^2}$, where σ^2 is the noise variance per real dimension. The LLR density function of W is $\mathcal{N} \sim \{\frac{2}{\sigma^2}, \frac{4}{\sigma^2}\}$. Let μ denote the mean of the LLR density function of W, and $\mu_N^{(i)}$ denote that of $W_N^{(i)}, 1 \leq i \leq N$. The evolution of LLR density function for each channel transformation $(W_N^{(i)}, W_N^{(j)}) \to (W_{2N}^{(i)}, W_{2N}^{(j)})$ is

$$\begin{cases} \mu_{2N}^{(i)} = f^{-1}\big(1 - \big(1 - f\big(\mu_N^{(i)}\big)\big)\big(1 - f\big(\mu_N^{(j)}\big)\big)\big) \\ \mu_{2N}^{(j)} = \mu_N^{(i)} + \mu_N^{(j)} \end{cases}, \tag{8.6}$$

where $f(x)$ is approximated by [119]

$$f(x) \approx \begin{cases} 1 & \text{if } x = 0 \\ e^{(-0.4527x^{0.86}+0.0218)} & \text{if } 0 < x \leq 10 \\ \sqrt{\frac{\pi}{x}}\left(1 - \frac{10}{7x}\right)e^{-\frac{x}{4}} & \text{if } x > 10 \end{cases} \cdot \tag{8.7}$$

To evaluate the performance of a polar code under SC decoding, we first initialized $\mu_1^{(1)}, \dots, \mu_1^{(N)}$ with the knowledge on the SNR of the transmission channel, and track the evolution of LLR via GA to obtain $\mu_N^{(1)}, \dots, \mu_N^{(N)}$. The block error rate of polar code can then be bounded by

$$P_e \leq 1 - \prod_{i \in \mathcal{A}} (1 - p_N^{(i)}), \tag{8.8}$$

where

$$p_N^{(i)} \approx Q\left(\sqrt{\mu_N^{(i)}/2}\right). \tag{8.9}$$

8.4.2 CBER of PIC-polar Codes

Different from uncoupled polar codes in which the CB decoding solely depends on the channel LLRs, the CBER of polar CB with coupling also depends on the availability of the extrinsic information. To derive the TBER performance of PIC-polar codes, the CBER of each polar CB under different decoding conditions should be characterized first. Here, the decoding conditions refer to that whether the extrinsic information of the head and/or tail coupled sequence is known by the CB decoder. To describe the error probability of the CB decoding for different decoding conditions, we define four CB error events that the decoding of CB$_t$ fails with

- no extrinsic information available,
- only head extrinsic information $\overrightarrow{\mathbf{u}}_{t-1,t}$ available,
- only tail extrinsic information $\overleftarrow{\mathbf{u}}_{t,t+1}$ available, and
- both head extrinsic information $\overrightarrow{\mathbf{u}}_{t-1,t}$ and tail extrinsic information $\overleftarrow{\mathbf{u}}_{t,t+1}$ available.

In the following, we denote the probabilities of these four events by $E_{0,0}$, $E_{1,0}$, $E_{0,1}$, and $E_{1,1}$, respectively.

For PIC-polar codes, the padding sequences are known by the encoder and decoder in advance. This is equivalent to transmitting the padding sequences over a noiseless channel. By the same token, if the head and/or tail extrinsic information bits are available at the decoder, it is equivalent to transmit these bits over a noiseless channel since only the correctly decoded extrinsic information of the coupled bits is used in the proposed polar CB decoding scheme.

Assuming that the extrinsic information of coupled bits (or padding bits) at indices $\widetilde{\mathcal{B}} \in \{\{\phi\}, \{\mathcal{B}_{\mathrm{H}}\}, \{\mathcal{B}_{\mathrm{T}}\}, \{\mathcal{B}_{\mathrm{H}}, \mathcal{B}_{\mathrm{T}}\}\}$ is available by the CB decoder, $E_{0,0}$, $E_{1,0}$, $E_{0,1}$, and $E_{1,1}$ can be calculated through (8.6) – (8.9) by setting $\mu_1^{(i)} = \infty$ for $i \in \widetilde{\mathcal{B}}$ and $\mu_1^{(i)} = \frac{1}{\sigma^2}$ for $i \in \widetilde{\mathcal{B}}^c$ when SC decoding is used. When other decoding algorithms are used, $E_{0,0}$, $E_{1,0}$, $E_{0,1}$, and $E_{1,1}$ can be obtained by using Monte-Carlo simulation.

CBER in the Feed-Forward Phase

We first consider the FF phase of LB-GB decoding, where the CB decoder processes serially from CB$_1$ to CB$_L$. From CB$_1$ to CB$_{L-1}$, CB decoder has no tail extrinsic information since their succeeding CBs have not been decoded yet. Therefore, for CB$_1$ which has padding bits $\mathbf{u}_{0,1}$ known by CB decoder, CB decoding has only one error event: decoding fails with only head extrinsic information available. From CB$_2$ to CB$_{L-1}$, CB decoding has two possible error events: 1) decoding fails with no extrinsic information available, and 2) decoding fails with only head extrinsic information available. For CB$_L$, with the existence of the tail padding bits $\mathbf{u}_{L,L+1}$, CB decoding has two possible error events: 1) decoding fails with only tail extrinsic information available, and 2) decoding fails with both head and tail extrinsic information available.

Based on the above analysis, the CBER of CB$_t$ in the proposed PIC-polar codes in the FF phase can be calculated by

$$P_t^{\mathrm{F}} = \begin{cases} E_{1,0} & \text{if } t = 1 \\ \left(1 - P_{t-1}^{\mathrm{F}}\right) E_{1,0} + P_{t-1}^{\mathrm{F}} E_{0,0} & \text{if } 2 \leq t \leq L-1 \cdot \\ \left(1 - P_{t-1}^{\mathrm{F}}\right) P_2 + P_{t-1}^{\mathrm{F}} E_{0,1} & \text{if } t = L \end{cases} \tag{8.10}$$

CBER in the Feed-Back Phase

We now consider the FB phase of LB-GB decoding. The probability that CB_t cannot be decoded in the FB phase is

$$\Pr\{\text{FB fail}\} = \Pr\{\text{FF fail}\} - \Pr\{\text{FB success}|\text{FF fail}\}, \tag{8.11}$$

where $\Pr\{\text{FB fail}\}$ is the probability that the decoding of CB_t fails in the FB phase, $\Pr\{\text{FF fail}\}$ is the probability that the decoding of CB_t fails in the FF phase computed by (8.10), and $\Pr\{\text{FB success}|\text{FF fail}\}$ is the conditional probability that CB_t is successfully decoded in the FB phase given that it was not decoded in the FF phase. In the following, we derive the conditional probability $\Pr\{\text{FB success}|\text{FF fail}\}$ in (8.11).

The event $\{\text{FB success}|\text{FF fail}\}$ appears when the following subevents happen in serial:

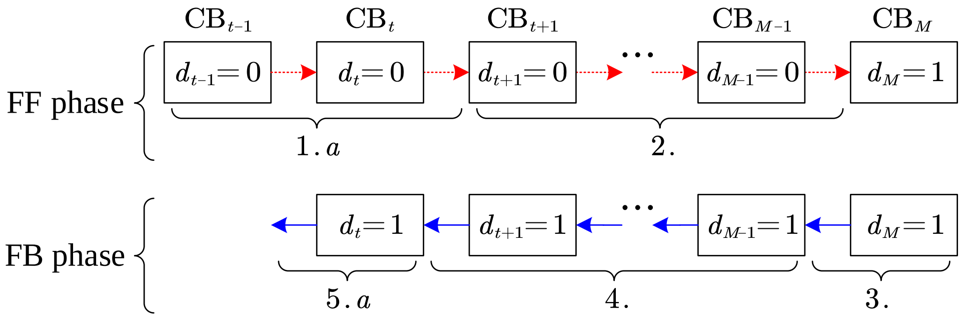
1. Decoding of CB_t fails in the FF phase with
    (a) no extrinsic information available or
    (b) only head extrinsic information $\overrightarrow{\mathbf{y}}_{t-1,t}$ available.
2. Decoding of CB_{t+1} – CB_{M+1} fails in the FF phase, where $t+1 \leq M \leq L$.
3. Decoding of CB_M succeeds in the FF phase.

4. Decoding of CB$_{M-1}$ – CB$_{t+1}$ succeeds in the FB phase.

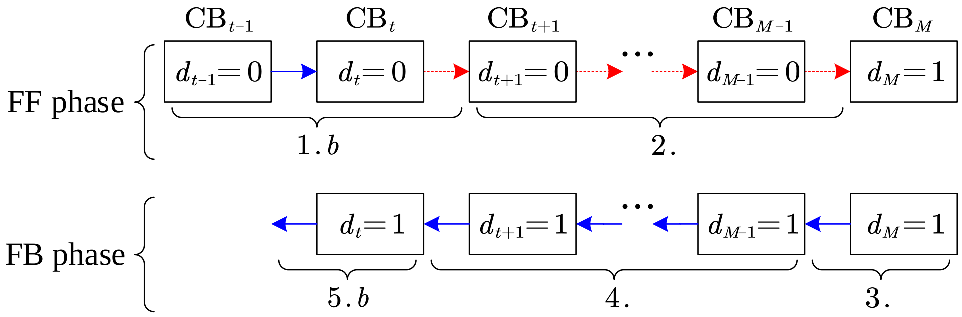
5. Decoding of CB$_t$ succeeds in the FB phase with

    (a) only tail extrinsic information $\overleftarrow{\mathbf{y}}_{t,t+1}$ available or

    (b) both head and tail extrinsic information $\{\overrightarrow{\mathbf{y}}_{t-1,t}, \overleftarrow{\mathbf{y}}_{t,t+1}\}$ available.



(a) Case A: no extrinsic information is available in the FF phase.



(b) Case B: head extrinsic information is available in the FF phase.

Figure 8.4: Examples of decoding success in the FB phase.

For a better understanding, we depict the appearance of these subevents in Fig. 8.4. In the figure, the directions of the arrows denote the order of these events. An arrow with solid line means that extrinsic information is passed between CBs, while an arrow with dashed line means that no extrinsic information is passed between CBs.

In event {FB success|FF fail}, the first three subevents happen in the FF phase. Therefore, the probabilities of these subevents can be derived based on

(8.10) and $E_{0,0}$, $E_{1,0}$, $E_{0,1}$, $E_{1,1}$. In particular, the probabilities corresponding to the first three subevents are

$$\Pr\{1a\} = P^{\mathrm{F}}_{t-1} E_{0,0} \tag{8.12a}$$

$$\Pr\{1b\} = \left(1 - P^{\mathrm{F}}_{t-1}\right) E_{1,0} \tag{8.12b}$$

$$\Pr\{2\} = \left(E_{0,0}\right)^{M-t-1} \tag{8.12c}$$

$$\Pr\{3\} = \begin{cases} 1 - E_{0,0} & \text{if } M < L \\ 1 - E_{0,1} & \text{if } M = L \end{cases} \tag{8.12d}$$

The last two subevents happen in the FB phase given that the decoding of the associated CBs fails in the FF phase. To derive the probabilities corresponding to these two subevents, we first obtain the probability that the decoding of CB_t fails in the FB phase given that it has also failed in the FF phase. Note that CB_t will be decoded in the FB phase only when it was not successfully decoded in the FF phase and obtains the tail extrinsic information $\overleftarrow{\mathbf{u}}_{t,t+1}$ from CB_{t+1}. When CB_t does not obtain $\overleftarrow{\mathbf{u}}_{t,t+1}$, the decoding of CB_t will always fail in the FB phase. There are two cases when a CB will be decoded in the FB phase:

- Decoding in the FF phase fails with no extrinsic information and tail extrinsic information is available in the FB phase, and
- Decoding in the FF phase fails with head extrinsic information and both head and tail extrinsic information are available in the FB phase.

The probability that the decoding of a CB fails in the FB phase for these two cases is

$$E_{0,1|0,0} = \frac{E_{0,1}}{E_{0,0}}, \tag{8.13a}$$

$$E_{1,1|1,0} = \frac{E_{1,1}}{E_{1,0}}, \tag{8.13b}$$

respectively. Then, the probability for the last two subevents is

$$\Pr\{4\} = (1 - E_{1,0})^{M-t-1} \tag{8.14a}$$

$$\Pr\{5a\} = 1 - E_{0,1|0,0} \tag{8.14b}$$

$$\Pr\{5b\} = 1 - E_{1,1|1,0} \tag{8.14c}$$

With (8.12) – (8.14), the conditional probability Pr{FB success|FF fail} for CB$_t$ in the case where no extrinsic information is available in the FF phase is

$$\Pr\{A\} = \sum_{M=t+1}^{L} \Big(\Pr\{1a\} \Pr\{5a\} \prod_{j=2}^{4} \Pr\{j\} \Big). \tag{8.15}$$

In this expression, the summation runs from $t+1$ to L since the FB phase may be triggered from any CB between CB$_{t+1}$ and CB$_L$ in the proposed decoding scheme. Similarly, Pr{FB success|FF fail} for CB$_t$ in the case where head extrinsic information is available in the FF phase is

$$\Pr\{B\} = \sum_{M=t+1}^{L} \Big(\Pr\{1b\} \Pr\{5b\} \prod_{j=2}^{4} \Pr\{j\} \Big). \tag{8.16}$$

Eventually, the probability that the decoding of CB$_t$ in the FB phase fails can be calculated by

$$P_t^{\mathrm{B}} = \Big(\Pr\{1a\} - \Pr\{A\} \Big) + \Big(\Pr\{1b\} - \Pr\{B\} \Big). \tag{8.17}$$

8.4.3 TBER of PIC-polar Codes

With (8.17), the TBER of PIC-polar codes under LB-GB decoding scheme is given by

$$P_{\mathrm{TB}} = 1 - \prod_{1 \le t \le L} (1 - P_t^{\mathrm{B}}). \tag{8.18}$$

In addition, the CBER of the PIC-polar codes is lower bounded by the error rate of the polar CB with both head and tail extrinsic information available, i.e., $P_t^{\mathrm{B}} \ge E_{1,1}$. Therefore, the TBER is lower bounded by

$$P_{\mathrm{TB}} \ge 1 - (1 - E_{1,1})^L. \tag{8.19}$$

We will show in Sec. 8.6 that the TBER derived in (8.18) and the TBER lower bound derived in (8.19) match the simulation results. In the next section, (8.18) and (8.19) are adopted to describe the TBER performance of PIC-polar codes with different coupling indices designs.

8.5 Design of Coupling Indices

In this section, we present the design of coupling indices for PIC-polar codes to improve their coding gains over the uncoupled polar codes. To achieve this, we first generate the coupling indices corresponding to various design criteria. Then, we use the TBER derived in (8.18) to evaluate which coupling indices design results in the lowest TBER.

Recall that for a sufficiently large blocklength, the capacities of bit-channels converge to either 0 or 1. However, for a finite blocklength, there is a non-negligible fraction of partially polarized bit-channels whose capacity approaches neither 0 nor 1 [142–144]. Thus, for a finite-length polar code, a few information bits are transmitted through the partially polarized bit-channels. Such bits are less reliable than those being transmitted via the fully polarized bit-channels. Hence, they dominate the CBER performance of polar codes.

Intuitively, it is beneficial to couple the most unreliable information bits to combat the partial polarization problem of finite-length polar codes. We consider two coupling indices designs related to the most unreliable bits:

- couple the most unreliable bits between consecutive CBs, and
- couple the most unreliable bits of one CB to the most reliable bits of a succeeding CB.

In addition, we also consider coupling the most reliable bits between consecutive CBs for comparison. Here, we use "U-U", "U-R", and "R-R" to represent the above three coupling indices designs, respectively.

To implement the above three coupling indices designs, we need to evaluate the reliability orders of all information bits and identify the most unreliable and the most reliable bits, respectively. By using GA, the indices of the most unreliable bits are chosen as

$$\mathcal{U} = \{i : \mu_N^{(i)} < \mu_N^{(j)}, i \in \mathcal{A}, j \in \mathcal{A} \setminus \mathcal{U}\}, \tag{8.20}$$

while the indices of the most reliable bits are chosen as

$$\mathcal{R} = \{i : \mu_N^{(i)} > \mu_N^{(j)}, i \in \mathcal{A}, j \in \mathcal{A} \setminus \mathcal{R}\}. \tag{8.21}$$

For "U-U", we let $|\mathcal{U}| = 2K_c$ and $\mathcal{B} = \mathcal{U}$. For "R-R", we let $|\mathcal{R}| = 2K_c$ and $\mathcal{B} = \mathcal{R}$. Let $b_1, \ldots, b_{2K_c}$ represent the $2K_c$ elements in $\mathcal{B}$. For both "U-U" and

"R-R", we have $\mathcal{B}_{\mathrm{H}} = \{b_1, b_3, \dots, b_{2K_c-1}\}$ and $\mathcal{B}_{\mathrm{T}} = \{b_2, b_4, \dots, b_{2K_c}\}$. For "U-R", the coupling indices are $\mathcal{B}_{\mathrm{H}} = \mathcal{U}$ and $\mathcal{B}_{\mathrm{T}} = \mathcal{R}$, with $|\mathcal{U}| = |\mathcal{R}| = K_c$.

The derived TBER and TBER lower bound for the considered coupling indices designs can be obtained via (8.18) and (8.19) respectively. Both the derived TBER and the TBER lower bound suggest that "U-U" should provide the best TBER performance. This evidence the previous intuition that coupling between the unreliable bits can partly solve the partial polarization problem of finite-length polar codes, and therefore brings a better coupling gain. In the following, "U-U" will be considered for coupling indices design of PIC-polar codes. We also use Monte-Carlo simulations to compare the TBER performance for these three coupling indices designs. The related results will be shown in Sec. 8.6.

Example 8.1. In Fig. 8.5, we show the derived TBERs and TBER lower bounds[3] of the rate-$\frac{1}{3}$ PIC-polar codes for the three considered coupling indices designs under SC decoding. In specific, $N = 1024$, $K = 383$, $K_c = 57$, and $L = 10$. It is observed that "U-U" gives the lowest TBER. ■

As to the coupling ratio, although a larger coupling ratio brings more extrinsic information during decoding, it also introduces a larger rate loss. We observe that the extra extrinsic information might not be enough to compensate for the rate loss when the coupling ratio is too large. According to the derived TBERs as well as the simulation results, an empirically good coupling ratio range is between 0.15 – 0.25.

---

[3] Instead of (8.8), we use the empirical inequality $P \geq 1 - \prod_{i \in \mathcal{M}} (1 - p_N^{(i)})$ to derive a lower bound $P_{2,LB}$ of $E_{1,1}$. Here, $\mathcal{M}$ is an indices set satisfies that $p_N^{(i)} > p_N^{(j)}$ for any $i \in \mathcal{M}$ and $j \in \mathcal{A} \setminus \mathcal{M}$, and $|\mathcal{M}| = 10$. Then, we use $P_{\mathrm{TB}} \geq 1 - (1 - P_{2,LB})^L$ as the TBER lower bound. This is because the CBER derived by (8.8) tends to be slightly greater than the simulated CBER due to the approximation error of GA.

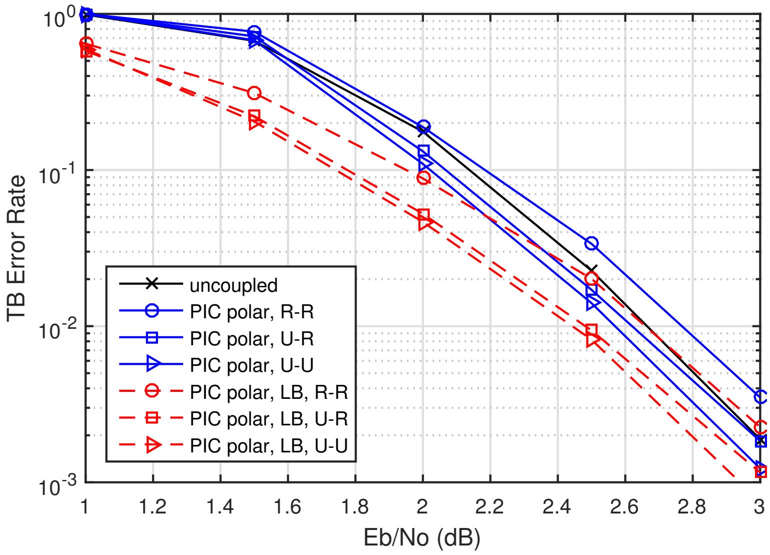


Figure 8.5: TBER and TBER lower bound of rate-$\frac{1}{3}$ PIC-polar codes with different coupling indices design under SC decoding. $\lambda = 0.15$, $L_W = 10$, and "LB" refers to lower bound.

8.6 Numerical Results

In this section, we first confirm that the derived TBER of PIC-polar codes in Sec. 8.4 complies with the simulation results. Then, we use the simulated TBER performance for different coupling indices to evaluate the superiority of the proposed coupling indices design. The TBER performance of PIC-polar codes with different code rates is simulated and compared to that of uncoupled polar codes, as well as the PIC-turbo codes [93], for the same code length and code rate. We also evaluate the decoding complexity and TBER performance trade-off for LB-GB scheme.

For all the simulations, we use $N = 1024$ and $L = 10$ for both uncoupled polar codes and PIC-polar codes. For the construction of PIC-polar codes, $\lambda = 0.15$ is used if not specified. As to the decoding, unless otherwise specified, we use CASCL decoder with a maximum list size of 8 for CB decoding and the LB-GB scheme with $L_W = 10$ for global decoding.

8.6.1 Derived TBER vs Simulated TBER

To evaluate the TBER analysis derived in (8.10) – (8.18), we obtain the TBER of PIC-polar codes via simulations and compare it with the derived TBER. Fig. 8.6 shows the TBER of PIC-polar code under SC decoding and CASCL decoding respectively. Note that the conditional CB error probabilities $\{E_{0,0}, E_{1,0}, E_{0,1}, P_2\}$ are obtained via GA for SC decoding, and via simulation for CASCL decoding.

It can be seen that the derived TBERs are very close to the simulated TBERs under both SC and CASCL decoding. At TBER of 10^{-3}, the gap between the derived TBER and the simulated TBER under SC decoding is less than 0.1 dB. The gap between the derived TBER and the simulated TBERs under CASCL decoding is smaller than that for SC decoding. This is because the conditional CB error probabilities are more accurate for the CASCL decoding since they are obtained via simulation.

In the same figure, we also plot the TBER lower bound derived in (8.19). Under both SC decoding and CASCL decoding, the simulated TBER approaches to the TBER lower bound when $\frac{E_b}{N_0}$ increases. This observation suggests that LB-GB scheme can effectively exploit the coupling gain of the PIC-polar codes.

8.6.2 TBER for Various Parameters

To confirm the effectiveness of the proposed coupling indices design, we obtain the TBER performance of the codes used in Fig. 8.5 with different coupling indices designs via simulation. Both SC decoding and CASCL decoding are considered. The results are shown in Fig. 8.7. The simulation results show that "U-U" provides the best TBER performance for both SC and CASCL decoding, which complies with the TBER analysis results in Sec. 8.5.

In Fig. 8.8, we evaluate TBER performance of PIC-polar codes for $R \in$

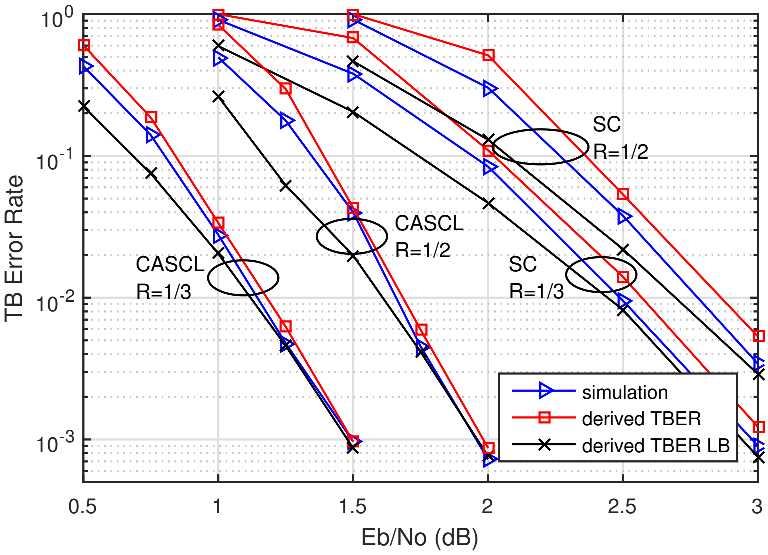


Figure 8.6: Derived and simulated TBER of PIC-polar codes under SC and CASCL decoding. $\lambda = 0.15$ and $L_W = 10$.

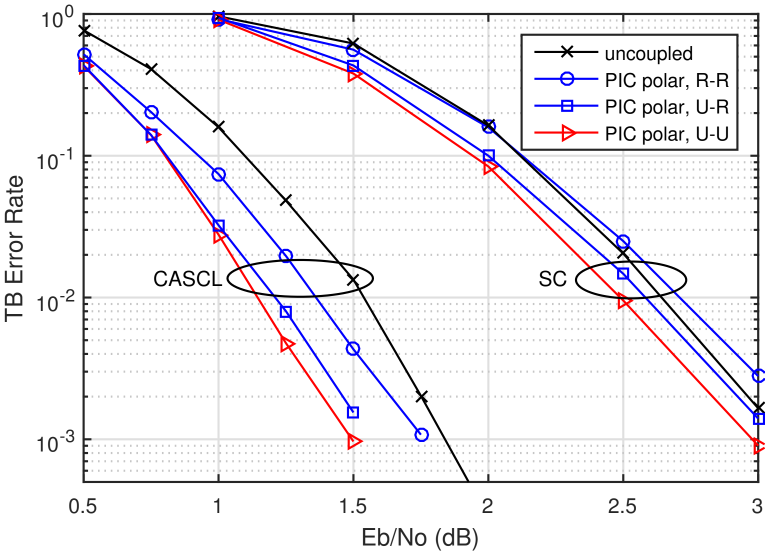


Figure 8.7: TBER of PIC-polar codes with different coupling indices designs. $\lambda = 0.15$ and $L_W = 10$.

$\{\frac{1}{4}, \frac{1}{3}, \frac{1}{2}, \frac{2}{3}\}$. It can be seen that PIC-polar codes can bring a consistent coding gain of $0.25 - 0.3$ dB over uncoupled polar codes at TBER between 10^{-2} and 10^{-3}. In the same figure, we also plot the TBER performance of PIC-turbo codes for $R \in \{\frac{1}{3}, \frac{1}{2}, \frac{2}{3}\}$. The LTE standard rate-$\frac{1}{3}$ turbo code is used for constructing the

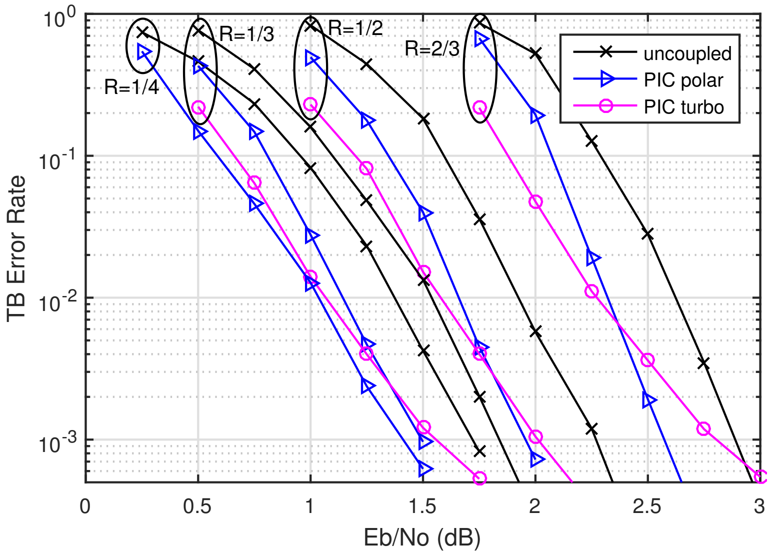


Figure 8.8: TBER of PIC-polar codes for various code rates under CASCL decoding. $\lambda = 0.2$ for $R = \frac{2}{3}$, $\lambda = 0.15$ for $R \in \{\frac{1}{4}, \frac{1}{3}, \frac{1}{2}\}$, and $L_W = 10$.

PIC-turbo codes. In each turbo CB, the coded sequence is punctured to ensure $N = 1024$. Although the PIC-turbo codes outperform the PIC-polar codes at a TBER of 10^{-2}, when $\frac{E_b}{N_0}$ increases, the PIC-turbo codes reach the error floor. Hence, the PIC-polar codes outperform the PIC-turbo codes at a TBER of 10^{-3}.

As we discussed in Sec. 8.3.2, LB-GB decoding scheme can be seen as a windowed decoding scheme with a variable window size. In this subsection, we evaluate how the maximum window size L_W affects the performance of the proposed decoder. In particular, we evaluate the TBER performance of rate-$\frac{1}{3}$ PIC-polar codes with coupling ratios $\lambda \in \{0.15,\ 0.25\}$ for various maximum window sizes. The results are shown in Fig. 8.9. In general, when λ increases, a larger L_W is required to approach the best TBER performance given by $L_W = L$. To be specific, for $\lambda = 0.15$, $L_W = 2$ exhibits obvious TBER performance loss, while $L_W = 3$ almost approaches the TBER given by $L_W = 10$. For $\lambda = 0.25$, $L_W = 3$ shows a considerable TBER performance degradation, while $L_W = 5$ provides a TBER close to the TBER given by $L_W = 10$.

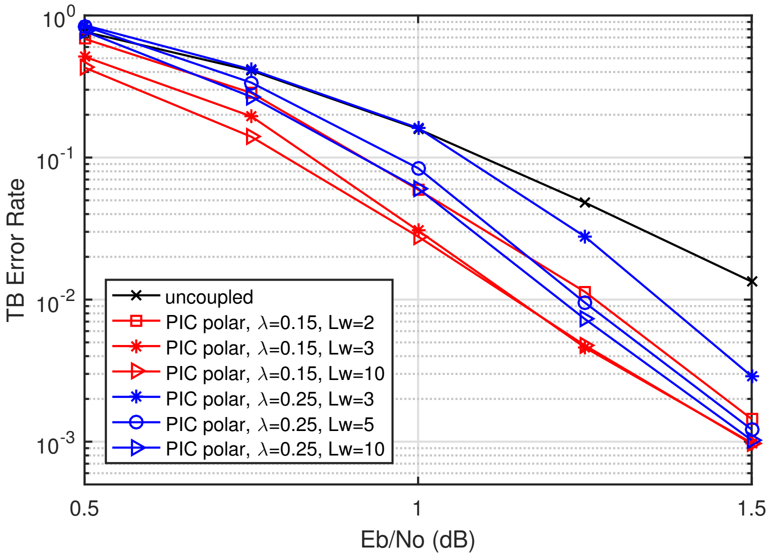


Figure 8.9: TBER of the rate-$\frac{1}{3}$ PIC-polar codes with various maximum decoding window sizes. CASCL decoding is considered.

8.6.3 Decoding Complexity

To evaluate the decoding complexity of PIC-polar codes, we compare it with that of the uncoupled polar codes with the same TB length and code rate. For simplicity, we normalize the decoding complexity of a CB to 1. In addition, for the uncoupled codes, a fully parallel decoding scheme and a fully serial decoding (with early termination) scheme are considered as the benchmark schemes. We investigate the decoding complexity in two scenarios: 1) single transmission and 2) with TB based retransmission.

Let us consider the first scenario. Consider uncoupled polar codes with L CBs, the fully parallel decoding scheme has a constant decoding complexity of

$$\bar{D}_P = L \tag{8.22}$$

as each CB is decoded once. The fully serial decoding scheme terminates global decoding once a CB decoding error is detected. Its average decoding complexity is given by

$$\bar{D}_S = L(1-P_u)^{L-1} + \sum_{t=1}^{L-1} tP_u(1-P_u)^{t-1}, \tag{8.23}$$

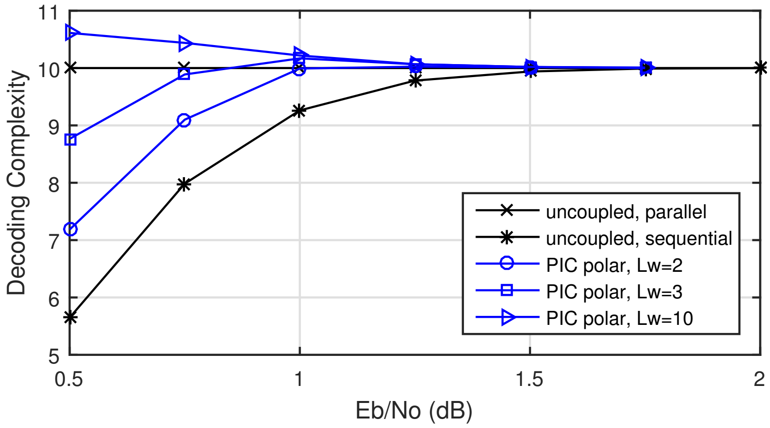
where P_u is the CBER of an uncoupled polar CB. For a PIC-polar code with L CBs, the decoding complexity is written as

$$D = \sum_{t=1}^{L} D_t, \tag{8.24}$$

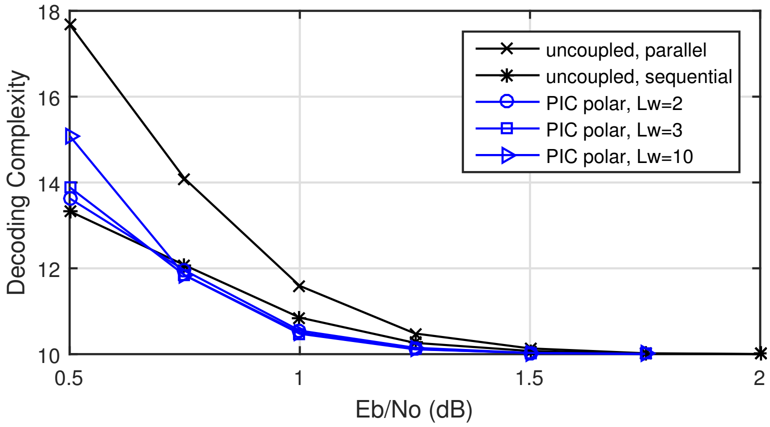
where D_t is the number that CB_t is processed by the CB decoder, which are collected through simulations. Here, $D_t \in \{0, 1, 2\}$ since any CB will be decoded at most twice in LB-GB scheme.

Fig. 8.10a compares the average decoding complexity of a rate-$\frac{1}{3}$ PIC-polar code to that of a rate-$\frac{1}{3}$ uncoupled polar codes for $L = 10$. At a low $\frac{E_b}{N_0}$, the uncoupled polar code with fully serial decoding has the lowest decoding complexity since early termination happens frequently. On the other hand, the PIC-polar code has a higher decoding complexity than the uncoupled polar code with fully serial decoding since some CBs are decoded repeatedly to obtain a better TBER performance. In addition, we observe that the decoding complexity of the PIC-polar code is significantly reduced by reducing the maximum window size L_W. At a high $\frac{E_b}{N_0}$, the probability that most CBs can be decoded at the first decoding attempt increases. Therefore, the decoding complexity of both the PIC-polar code and the uncoupled polar code with fully serial decoding approaches L.

Now, we consider the second scenario in which the TB based retransmission is taken into account for evaluating the decoding performance of both codes. In a practical $\frac{E_b}{N_0}$ range, when the received signal power is doubled, the TBER rate



(a) Scenario 1: single transmission.



(b) Scenario 2: with TB based retransmission.

Figure 8.10: Normalized decoding complexity of rate-$\frac{1}{3}$ PIC-polar codes under CASCL decoding. $\lambda = 0.15$. The corresponding TBER performance are plotted in Fig. 8.9.

will drop significantly. Therefore, we assume that at most one retransmission will happen. By the same token, we also assume that when retransmission happens, the decoding complexity of the retransmitted TB is L. In this scenario, the average decoding complexity for the uncoupled polar code with fully parallel and fully serial decoding schemes is

$$\bar{D}'_P = \bar{D}_P + L \cdot P_{u,\mathrm{TB}}, \tag{8.25a}$$

$$\bar{D}'_S = \bar{D}_S + L \cdot P_{u,\mathrm{TB}}, \tag{8.25b}$$

respectively. Here, $P_{u,\mathrm{TB}}$ is the TBER of the uncoupled polar code in the first transmission. The average decoding complexity with retransmission for the PIC-polar code is written as

$$\bar{D}' = \bar{D} + L \cdot P_{c,\mathrm{TB}}, \tag{8.26}$$

where $\bar{D}$ is the average decoding complexity in the first transmission, and $P_{c,\mathrm{TB}}$ is the TBER of PIC-polar code in the first transmission.

Fig. 8.10b shows the average decoding complexity for the PIC-polar code and the uncoupled polar code when TB based retransmission is employed. At a low $\frac{E_b}{N_0}$, the decoding complexity of the PIC-polar code is slightly higher than the uncoupled polar code with fully serial decoding. As $\frac{E_b}{N_0}$ increases, the decoding complexity of the PIC-polar code becomes lower than the uncoupled polar code with fully serial decoding. For all the $\frac{E_b}{N_0}$ values, the PIC-polar code always has a lower decoding complexity than the uncoupled polar code with fully parallel decoding. With these observations, we can conclude that PIC-polar codes achieve a better TBER performance with a slightly increased or even lower decoding complexity compared to the uncoupled polar code under the simulated code parameters.

8.7 Conclusion

In this chapter, we constructed PIC-polar codes to improve the transmission reliability for long information sequences. A closed-form expression of TBER for the PIC-polar codes under SC decoding was derived, and was applied to design the coupling indices. Simulation results confirmed the effectiveness of our analysis and demonstrated the improvement on the TBER performance of PIC-polar codes over uncoupled polar codes.

Chapter 9

PIC-BIPCM

In this chapter, we propose the partially information coupled bit-interleaved polar coded modulation (PIC-BIPCM) scheme, which is the PIC-polar coding scheme designed for 2^Q-ary modulations. Specifically, we consider PIC-BIPCM schemes constructed with three BIPCM schemes: direct BIPCM, punctured BIPCM, and multi-kernel BIPCM, respectively. We analyze the error performance of the proposed PIC-BIPCM over the binary erasure channel (BEC) via density evolution. With the analysis as a guideline, we jointly design the positions of coupled bits and the modulation bit-mapper by taking into account the partial polarization of finite-length polar codes as well as the unequal error protection of high order modulations. Simulation results demonstrate a significant performance improvement of the proposed PIC-BIPCM over the uncoupled BIPCM on both BEC and AWGN channels.

9.1 Introduction

In Chapter 8, we proposed PIC-polar codes (PIC-PCs) [139], which employ systematic polar codes as component codes. In the PIC-polar coding scheme, the consecutive CBs are coupled by sharing some information bits between them. The decoding of PIC-PCs is conducted in a CB-wise manner, and the hard-decision extrinsic information is passed between the CBs via the shared bits. The implementation of PIC-PCs preserves the architecture of conventional (non-systematic) polar codes. Hence, the component decoder can adopt various decoding algorithms designed for polar codes, including the CRC-aided successive cancellation list (CASCL) decoding [28], whose superior decoding performance has been shown in the literature.

However, the previous PIC-PCs only consider the binary-input channel, i.e., the modulation order is one. In many wireless communication systems, high order modulations are employed to increase the spectral efficiency [145,146]. Due to the partial polarization of finite-length polar codes, the error probabilities of information bits in the same codeword are diverse so that some information bits have significantly higher error probabilities than others. Vaiours spatially coupled polar codes [66–69, 147–151], including our PIC-PCs, exhibit such a property. As modulation would affect the polarization of the underlying component polar codes, the spatially coupled polar coding schemes proposed for the binary-input channel may not promise good error performance when directly used in conjunction with high-order modulations, which needs to tackled in the research.

To reap the benefits of spatial coupling and coded modulation for polar codes, we propose partially information coupled bit-interleaved polar coded modulation (PIC-BIPCM) [152] for 2^Q-ary modulations, where $Q \geq 2$. Note that directly combining PIC-PCs designed for the binary input channel and BIPCM does not guarantee superior performance over the uncoupled BIPCM without a careful design. The design of PIC-BIPCM needs to jointly consider the coupling indices (the positions of shared information bits at the input of polar code encoder) and the bit-mapper (the positions of coded bits at the input of the modulator in the BIPCM scheme). We will show that the benefits of spatial coupling and coded modulation can be jointly attained by the proposed PIC-BIPCM.

The main contributions of this work are summarized as follows.

- We propose a class of PIC-BIPCM scheme for 2^Q-ary modulations. To allow our scheme to be adapted to any modulation order $Q \geq 2$, we consider three different PIC-BIPCM schemes with direct BIPCM, punctured BIPCM, and multi-kernel BIPCM, respectively. The encoding and decoding methods for the PIC-BIPCM schemes with coupling memory $m \geq 1$ and various modulation orders are provided.

- We analyze the error performance of PIC-BIPCM for any given coupling indices and bit-mapper over the binary erasure channel (BEC) by using density evolution. In specific, we derive a closed-form expression of the CB error rate (CBER) for coupling memory $m = 1$, and a CBER lower bound for coupling memory $m \geq 1$.

- We jointly design the coupling indices and bit-mapper to obtain a large coding gain via a two-step approach. We first provide the criteria of designing coupling indices for any given bit-mapper. On one hand, designing coupling indices needs to consider that the unequal error protection of high order modulations would affect the polarization of the underlying component polar codes. On the other hand, designing coupling indices also needs to take into consideration their impact on the bit-mapping. We then present the method for determining the bit-mapper for PIC-BIPCM constructed with direct BIPCM, punctured BIPCM, and multi-kernel BIPCM, respectively. Specifically, we use the derived CBER closed-form expression to find the bit-mapper that leads to the lowest CBER while incorporating the design criteria of coupling indices.

- We evaluate the CBER performance of PIC-BIPCM with various modulations over both the BEC and the AWGN channel. The simulation results verify the CBER performance analysis and show that the designed PIC-BIPCM can achieve a performance gain up to 0.8 dB at a CBER of 10^{-3} over the uncoupled BIPCM.

This chapter is organized as follows. Sec. 9.2 reviews the existing BIPCM schemes. Sec. 9.3 introduces the encoding and decoding of the proposed PIC-BIPCM scheme. In Sec. 9.4, we analyze the CBER performance of PIC-BIPCM over the BEC. Sec. 9.5 focuses on designing coupling indices and bit-mapper for PIC-BIPCM. Numerical results are shown in Sec. 9.6. Lastly, Sec. 9.7 draws the conclusion.

9.1.1 Notations

In this chapter, as high-order modulations are used, we introduce some new notations in correspondence. In addition, we redefine some vector representations. The details are as follows.

$[i:j]$	An integer set $\{i, i+1, \ldots, j\}$, $i < j$.
$x_{[1:N]}$	A row vector $[x_1, x_2, \ldots, x_N]$
$x_{[i:j]}$	A subvector $[x_i, x_{i+1}, \ldots, x_j]$ of $x_{[1:N]}$, $1 \leq i < j \leq N$.
$W^{(1:q)}$	A set of q parallel B-DMS channels $[W^{(1)}, \ldots, W^{(q)}]$.
$W^{(i:j)}$	A subset of parallel B-DMS channels $[W^{(i)}, \ldots, W^{(j)}]$, $i < j$.
$\bar{I}(W^{(1:q)})$	Average capacity of $W^{(1:q)}$, $\bar{I}(W^{(1:q)}) \triangleq \sum_{i=1}^{q} \frac{I(W^{(i)})}{q}$.

9.2 Bit-Interleaved Polar Coded Modulation

This section provides the necessary background related to BIPCM. We first present a brief overview on coded modulation schemes. We then review the BICM system model. Lastly, we summary the existing BIPCM schemes.

9.2.1 Coded Modulation Overview

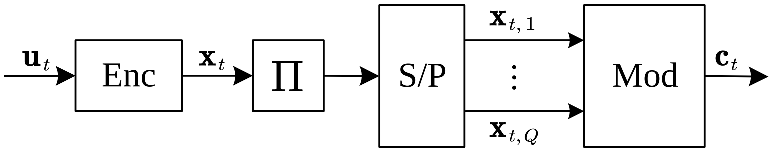
There are two broad categories of code modulation schemes. One is multilevel coding (MLC) [51], where multiple codewords with different code rates are modulated simultaneously such that each codeword is transmitted via one of the bit-positions of the modulated symbols. Another is bit-interleaved coded modulation (BICM) [52], which only uses a single channel code. Each codeword is independently modulated so that each bit-position of modulated symbols transmits one subblock of the codeword. Their corresponding schemes using polar codes as the constituent codes are referred to as multi-level polar coding (MLPC) [153–155] and bit-interleaved polar coded modulation (BIPCM) [156–158] respectively.

9.2.2 Bit-Interleaved Coded Modulation

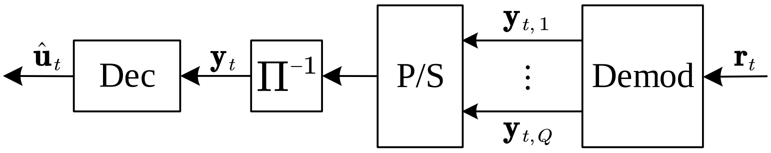
Here, we review the BICM system model introduced in [52]. The transmitter and receiver block diagrams are depicted in Fig. 9.1.

Fig. 9.1a shows the transmitter block diagram. At time t, a K-bit information sequence $\mathbf{u}_t$ is encoded, results in codeword $\mathbf{x}_t$ of length N. After interleaving (Π) and a serial-to-parallel (S/P) module, $\mathbf{x}_t$ is segmented into m subblocks of length $\frac{N}{m}$, denoted by $\mathbf{x}_{t,1}, \ldots, \mathbf{x}_{t,Q}$. Then, the modulator takes these subblocks as input, and produces symbol sequence $\mathbf{c}_t$ of length $\frac{N}{m}$.

Fig. 9.1b shows the receiver block diagram. At time t, the receiver obtains



(a) Transmitter.



(b) Receiver.

Figure 9.1: BICM system block diagram.

$\mathbf{r}_t = \mathbf{c}_t + \mathbf{n}_t$, where $\mathbf{n}_t$ is the complex-value Gaussian noise. The demodulator takes $\mathbf{r}_t$ as input and outputs the LLR sequences $\mathbf{y}_{t,1}, \ldots, \mathbf{y}_{t,Q}$, which are the channel LLRs of $\mathbf{x}_{t,1}, \ldots, \mathbf{x}_{t,Q}$. After the parallel-to-serial (P/S) module and deinterleaving (Π^{-1}), the LLR sequences are reordered into $\mathbf{y}_t$. Lastly, the decoder produces the hard-decision estimation $\hat{\mathbf{u}}_t$.

Let $W : \mathcal{X} \to \mathcal{Y}$ be a DMC with modulation symbol set $\mathcal{X}$ and output alphabet $\mathcal{Y}$. Given a label set $\mathcal{L}$, a BICM scheme maps W to a set of Q synthesized channels: $W \to W^{(1:Q)}$. The i-th synthesized channel, denoted by $W^{(i)}$, corresponds to the i-th bit-position of the modulated symbols. The capacity of $W^{(i)}$ with channel input $b \in \{0, 1\}$ and output $y \in \mathcal{Y}$ is computed as [52]

$$I(W^{(i)}) = 1 + \mathop{\mathbb{E}}_{b,y}\left[\log \frac{\sum_{x \in \mathcal{X}_{i,b}} W(y|x)}{\sum_{x \in \mathcal{X}} W(y|x)}\right], \tag{9.1}$$

where $\mathcal{X}_{j,b}$ is the constellation subset of $\mathcal{X}$ consisting of all the symbols whose j-th bit-label is $b \in \{0, 1\}$ for any $1 \leq i \leq Q$. Assume an ideal interleaver, when W is an AWGN channel, we consider that its synthesized channels are a set of parallel BI-AWGN channels. Fig. 9.2 shows the equivalent BICM system model with synthesized channels, where each subblock of $\mathbf{x}_t$ is mapped to one of the

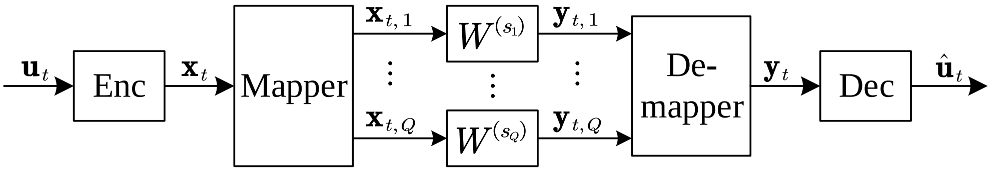


Figure 9.2: BICM equivalent system model.

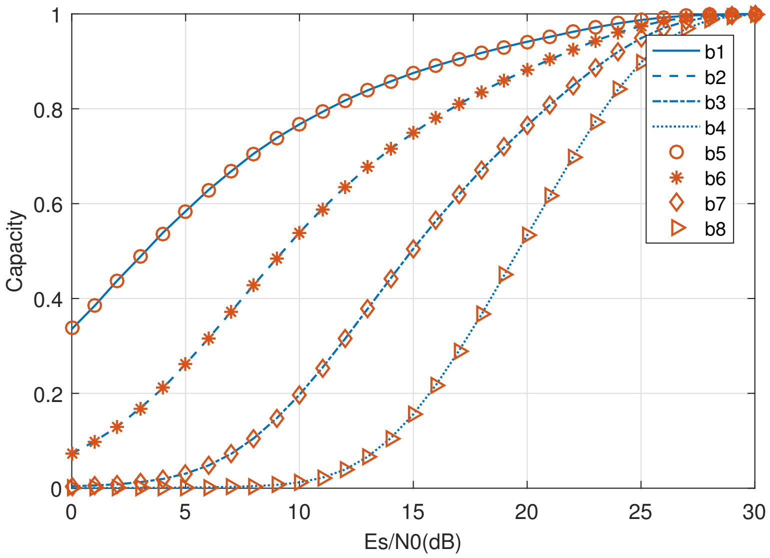


Figure 9.3: Synthesized channel capacities for BICM system with Gray-labeled 256-QAM.

synthesized channel. Specifically, given a bit-mapper $\mathbf{s} = [s_1, \ldots, s_Q]$, subblock $\mathbf{x}_{t,i}$ is mapped to $W^{(s_i)}$.

For the code design of a BICM system, we may further simplify the equivalent model. Let q be the number of synthesized channels that are taken into consideration for the design. In most cases, we need to consider all $q = Q$ synthesized channels. When Gray-labeled 2^Q-QAM is used, where Q is an even number, the capacities of synthesized channels satisfy that $I(W^{(i)}) = I(W^{(i+\frac{Q}{2})})$ for $1 \leq i \leq \frac{Q}{2}$. For the code design in such case, we only need to consider $q = \frac{Q}{2}$ synthesized channels: $W^{(1)}, \ldots, W^{(\frac{Q}{2})}$ by using $W^{(i)}$, $1 \leq i \leq \frac{Q}{2}$, to represent both the i-th bit position and the $(i + \frac{Q}{2})$-th bit position of the modulated symbols.

Example 9.1. Consider a BICM system with Gray-labeled 256-QAM, i.e., $Q =$

8. Let b_i represents the i-th bit-position. We plot the capacity of each bit-position in Fig. 9.3. There are four pairs of bit-positions with equal capacity: (b_1, b_5), (b_2, b_6), (b_3, b_7), and (b_4, b_8). Therefore, we use $q = 4$ synthesized channels $W^{(1)}$, $W^{(2)}$, $W^{(3)}$, and $W^{(4)}$ to represent each pair of bit-positions, respectively. ■

9.2.3 BIPCM schemes

Various BIPCM schemes have been proposed in the literature. In [156], the authors proposed a generalized BIPCM scheme where a multi-kernel [132–134] polar code design is adopted such that a capacity-achieving performance can be obtained for any 2^Q-ary modulation. In [157], a multi-kernel polar code consisting of a 2×2 kernel and a 3×3 kernel is adopted to construct BIPCM for 64-QAM. In [158], the authors proposed punctured BIPCM, where zero-capacity virtual channels are introduced to adapt any 2^Q-ary modulation with a fixed 2×2 kernel. The punctured BIPCM and the multi-kernel BIPCM have comparable error performance, while the former has a lower construction complexity and the latter has a lower decoding complexity.

In the following, we classify the existing BIPCM schemes into three classes, and summary their construction methods, respectively.

Direct BIPCM Scheme [156,158,159]

Let n be a positive integer and $N = 2^n$. The polarization matrix for a length-N polar code is given by $\mathbf{G}_N = \mathbf{F}^{\otimes n}$, where $\mathbf{F} \triangleq \left[\begin{smallmatrix} 1 & 0 \\ 1 & 1 \end{smallmatrix}\right]$. Note that $\mathbf{G}_N$ can be seen as the product code of $\mathbf{G}_{N/q}$ and $\mathbf{G}_q$, written as

$$\mathbf{G}_N = \mathbf{G}_q \otimes \mathbf{G}_{N/q} = (\mathbf{I}_q \otimes \mathbf{G}_{N/q}) \cdot (\mathbf{G}_q \otimes \mathbf{I}_{N/q}) \tag{9.2}$$

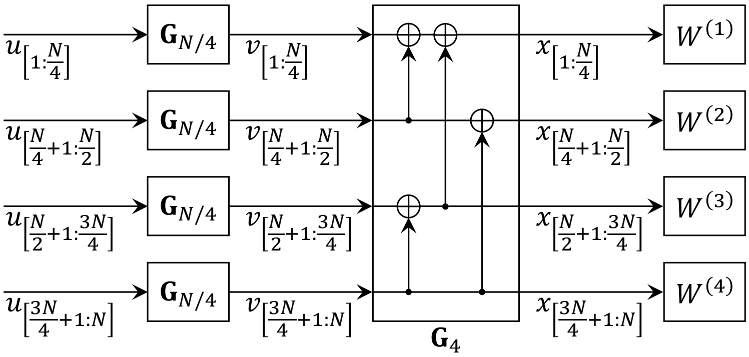


Figure 9.4: Direct BIPCM for $N = 2^n$, $q = 4$, $\mathbf{s} = [1, 2, 3, 4]$.

Given input sequence $u_{[1:N]}$, the polar encoding process $x_{[1:N]} = u_{[1:N]}\mathbf{G}_N$ can be rewritten as

$$\begin{aligned} x_{[1:N]} &= u_{[1:N]}\mathbf{G}_N \\ &= \left[u_{[1:\frac{N}{q}]}, \ldots, u_{[\frac{(q-1)N}{q}+1:N]}\right] \cdot (\mathbf{I}_q \otimes \mathbf{G}_{N/q}) \cdot (\mathbf{G}_q \otimes \mathbf{I}_{N/q}) \\ &= \left[u_{[1:\frac{N}{q}]}\mathbf{G}_{N/q}, \ldots, u_{[\frac{(q-1)N}{q}+1:N]}\mathbf{G}_{N/q}\right] \cdot (\mathbf{G}_q \otimes \mathbf{I}_{N/q}) \\ &= \left[x_{[1:\frac{N}{q}]}, \ldots, x_{[\frac{(q-1)N}{q}+1:N]}\right]. \end{aligned} \tag{9.3}$$

We call $x_{[\frac{(i-1)\cdot N}{q}+1:\frac{i\cdot N}{q}]}$, $1 \le i \le q$, the i-th sub-codeword of $x_{[1:N]}$.

Consider that q is a power of 2, e.g. Gray-labeled 16-QAM ($q = 2$) and Gray-labeled 256-QAM ($q = 4$). The polar codeword $x_{[1:N]}$ can be directly fitted into channel $W^{(1:q)}$ by segmenting $x_{[1:N]}$ to q sub-codewords of length $\frac{N}{q}$. Given a bit-mapper $\mathbf{s} = [s_1, \ldots, s_q]$, the i-th sub-codeword is mapped to $W^{(s_i)}$. In other words, the encoding of direct BIPCM over $W^{(1:q)}$ is conducted in two steps:

1. Length-$\frac{N}{q}$ polar encoding: The input sequence $u_{[1:N]}$ is segmented into q information blocks, where the i-th information block is $u_{[\frac{(i-1)N}{q}+1:\frac{iN}{q}]}$, $1 \le i \le q$. Then, the i-th information block is encoded with $\mathbf{G}_{N/q}$, producing the codeword:

$$v_{[\frac{(i-1)N}{q}+1:\frac{iN}{q}]} = u_{[\frac{(i-1)N}{q}+1:\frac{iN}{q}]}\mathbf{G}_{N/q}.$$

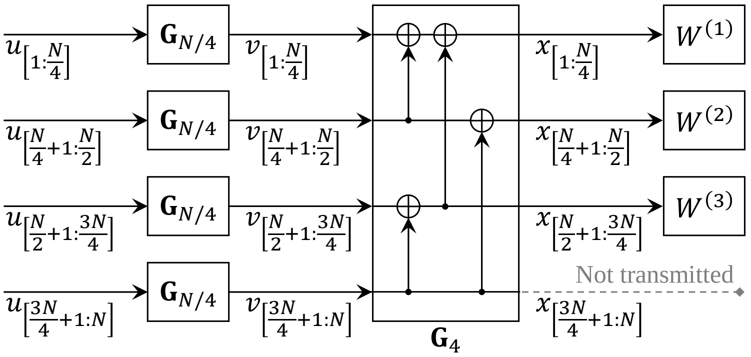


Figure 9.5: Punctured BIPCM for $N = 2^n$, $q = 3$, $q' = 4$, $\mathbf{s} = [1, 2, 3, 4]$. Sub-codeword $x_{[\frac{3N}{4}+1:N]}$ is mapped to a virtual subchannel and consequently not transmitted.

2. Length-q BIPCM: For $1 \le j \le \frac{N}{q}$, conducting length-q polar encoding:

$$[x_j, x_{j+\frac{N}{q}}, \ldots, x_{j+\frac{(q-1)N}{q}}] = [v_j, v_{j+\frac{N}{q}}, \ldots, v_{j+\frac{(q-1)N}{q}}]\mathbf{G}_q.$$

   Consider a bit-mapper $\mathbf{s} = [s_1, \ldots, s_q]$. For $1 \le i \le q$, the i-th coded bit $x_{j+\frac{(i-1)N}{q}}$ is mapped to $W^{(s_i)}$.

An example of a direct BIPCM scheme is shown in Fig. 9.4.

Punctured BIPCM scheme [158]

Consider that q is not a power of 2, e.g. quasi-Gray-labeled 32-QAM ($q = 5$) and Gray-labeled 64-QAM ($q = 3$). The polar codeword $x_{[1:N]}$ cannot be directly fitted into channel $W^{(1:q)}$ as the codeword length is not an integer multiple of the modulation order. One scheme to deal with this case is the punctured BIPCM. Let $q' = 2^{\lceil \log_2 q \rceil}$. The punctured BIPCM introduces $q' - q$ zero-capacity virtual subchannels, resulted in $W^{(1:q')} = [W^{(1:q)}, W^{(q+1)}, \ldots, W^{(q')}]$. Given a bit-mapper $\mathbf{s} = [s_1, \ldots, s_{q'}]$, $x_{[1:N]}$ is split into q' sub-codewords of length $\frac{N}{q'}$. For $1 \le i \le q$, the i-th sub-codeword $x_{[\frac{(i-1)\cdot N}{q'}+1:\frac{i\cdot N}{q'}]}$ is mapped into subchannel $W^{(s_i)}$. The sub-codewords mapped to the virtual subchannels $W^{(q+1:q')}$ are punctured, while the rest of the sub-codewords are mapped to the modulation symbols. An example

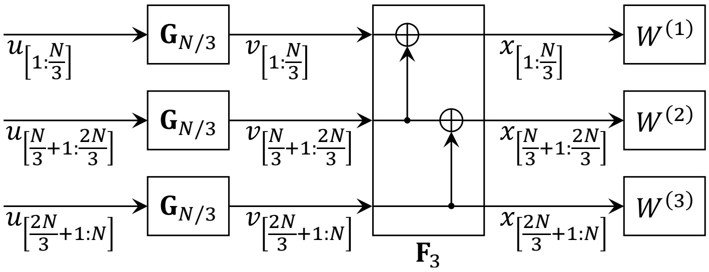


Figure 9.6: Multi-kernel BIPCM for $N = 3 \cdot 2^{n-1}$, $q = 3$, $\mathbf{s} = [1, 2, 3]$.

of the punctured BIPCM scheme is shown in Fig. 9.5.

Multi-kernel BIPCM scheme [156, 157]

The multi-kernel BIPCM can also be employed when q is not a power of 2. It adopts a multi-kernel polar code which is constructed by combining a conventional polar code and a $q \times q$ kernel $\mathbf{F}_q$. The generator matrix constructed with n recursion layers is $\mathbf{G}_N = \mathbf{F}_q \otimes \mathbf{F}^{\otimes n-1}$, where $N = q \cdot 2^{n-1}$. Given a bit-mapper $\mathbf{s} = [s_1, \ldots, s_q]$, the codeword $x_{[1:N]}$ is segmented into q sub-codewords of length $\frac{N}{q}$. For $1 \leq i \leq q$, the i-th sub-codeword $x_{[\frac{(i-1)\cdot N}{q}+1:\frac{i\cdot N}{q}]}$ is mapped into subchannel $W^{(s_i)}$. An example of the multi-kernel BIPCM scheme is shown in Fig. 9.6.

Let $\widetilde{N}$ be the transmission length of a polar codeword in BIPCM scheme. When direct BIPCM or multi-kernel BIPCM is adopted, $\widetilde{N} = N$ because all the bits in a codeword are transmitted. When punctured BIPCM is adopted, $\widetilde{N} = \frac{qN}{q'}$ because $q' - q$ sub-codewords are punctured. The punctured BIPCM of length $N = q' \cdot 2^{n-\log_2 q'}$ and the multi-kernel BIPCM of length $N = q \cdot 2^{n-1}$ have the same transmission length $\widetilde{N} = q \cdot 2^{n-1}$. They also have similar level of error performance [158]. The punctured BIPCM has lower construction complexity as it only consists of a fixed kernel $\mathbf{F}$, while the multi-kernel BIPCM consists of two kernels $\mathbf{F}$ and $\mathbf{F}_q$. In another aspect, the punctured BIPCM has a decoding complexity of $O(N \cdot \log_2 N)$, and the multi-kernel BIPCM has a lower decoding complexity of $O(N \cdot (\log_2 \frac{N}{q} + 1))$. In particular, when $q = 3$, the decoding complexities of these two scheme are $O(2^{n+1} \cdot (n+1))$ and $O(3 \cdot 2^{n-1} \cdot n)$, respectively.

In this chapter, all above three BIPCM schemes are considered in our design of PIC-BIPCM.

9.3 PIC-BIPCM for 2^Q-ary Modulation

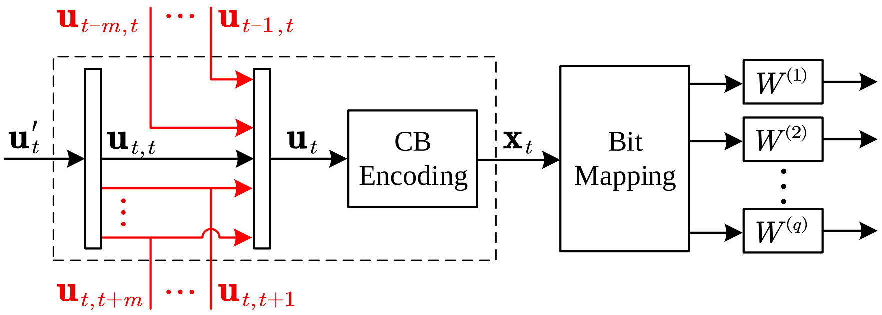
In this section, we first provide the necessary backgrounds related to polar codes. Then, we review the BICM channel model and the existing BIPCM schemes. Later, we use three different BIPCM schemes to respectively construct the PIC-BIPCM. Lastly, the decoding of PIC-BIPCM is presented.

9.3.1 Construction of PIC-BIPCM

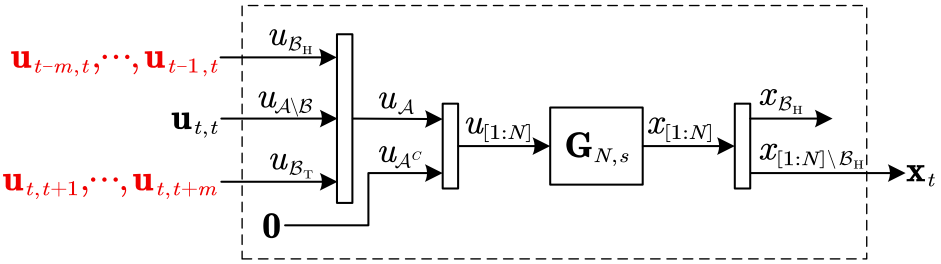
In this section, we introduce the architecture of the PIC-BIPCM with coupling memory $m \geq 1$. Let L denote the number of CBs in a TB. At time t, $1 \leq t \leq L$, the t-th CB (CB_t) is coupled with m preceding CBs (from CB_{t-m} to CB_{t-1}) and m succeeding CBs (from CB_{t+1} to CB_{t+m}).

Information Segmentation

Let $\mathbf{u}$ denote the information sequence of a TB. First, $\mathbf{u}$ is segmented into L sub-sequences $\mathbf{u}'_1, \ldots, \mathbf{u}'_L$. As shown in Fig. 9.7a, at time t, $\mathbf{u}'_t$ is decomposed into $m+1$ sub-sequences $\mathbf{u}_{t,t}, \mathbf{u}_{t,t+1}, \ldots, \mathbf{u}_{t,t+m}$, where $\mathbf{u}_{t,t}$ is the uncoupled information of CB_t, and $\mathbf{u}_{t,t+i}$, $1 \leq i \leq m$, is the coupled information shared between CB_t and CB_{t+i}. To initialize and terminate coupling, zero-paddings are applied to both ends of the information sequence such that $\mathbf{u}_{t-i,t} = \mathbf{0}$ when $t - i < 1$ and



(a) PIC-BIPCM encoder block diagram at time t.


$\mathbf{u}_{t-m,t},\cdots,\mathbf{u}_{t-1,t}$
$u_{\mathcal{B}_\text{H}}$
$\mathbf{u}_{t,t}$
$u_{\mathcal{A}\backslash\mathcal{B}}$
$\mathbf{u}_{t,t+1},\cdots,\mathbf{u}_{t,t+m}$
$u_{\mathcal{B}_\text{T}}$
$\mathbf{0}$
$u_{\mathcal{A}}$
$u_{\mathcal{A}^c}$
$u_{[1:N]}$
$\mathbf{G}_{N,s}$
$x_{[1:N]}$
$x_{\mathcal{B}_\text{H}}$
$x_{[1:N]\backslash\mathcal{B}_\text{H}}$
$\mathbf{x}_t$

(b) CB encoder block diagram at time t.

Figure 9.7: Construction of PIC-BIPCM.

$\mathbf{u}_{t,t+i} = \mathbf{0}$ when $t + i > L$.

CB Encoding

At time t, the CB encoder takes $\mathbf{u}_t = [\mathbf{u}_{t-m,t}, \ldots, \mathbf{u}_{t-1,t}, \mathbf{u}'_t]$ as input, and produces the codeword $\mathbf{x}_t = [\mathbf{u}'_t, \mathbf{v}_t]$, where $\mathbf{v}_t$ is the parity sequence of CB_t. The sequences $\mathbf{u}_{t-m,t}, \ldots, \mathbf{u}_{t-1,t}$ are referred to as the head coupled sequences of CB_t. The sequences $\mathbf{u}_{t,t+1}, \ldots, \mathbf{u}_{t,t+m}$, which are the sub-sequences of $\mathbf{u}'_t$, are referred to as the tail coupled sequences of CB_t. Note that for any $1 \le i \le m$, $\mathbf{u}_{t-i,t}$ is the head coupled sequence of CB_t, as well as the tail coupled sequence of CB_{t-i}. In the following, we present the structure of the CB encoder.

The structure of the CB encoder is shown in Fig. 9.7b. The component code of the CB encoder is a systematic polar code of length N, denoted as $\mathbf{G}_{N,s}$ in Fig. 9.7b. The component encoder inputs $u_{[1:N]}$, and outputs $x_{[1:N]} = u_{[1:N]} \cdot \mathbf{G}_{N,s}$.

Given information indices $\mathcal{A}$, $u_{\mathcal{A}}$ takes the information bits, and $u_{\mathcal{A}^c}$ takes the frozen bits. In the output vector $x_{[1:N]}$, $x_{\mathcal{A}}$ consists of the systematic bits (i.e., $x_i = u_i$ for $i \in \mathcal{A}$), and $x_{\mathcal{A}^c}$ consists of the parity bits.

Let $\mathcal{B}_{\mathrm{H}}$ and $\mathcal{B}_{\mathrm{T}}$ denote two disjoint subsets of $\mathcal{A}$, representing indices subsets for the head and tail coupled sequences, respectively. We also define $\mathcal{B} \triangleq \{\mathcal{B}_{\mathrm{H}}, \mathcal{B}_{\mathrm{T}}\}$. The head coupled sequences, the tail coupled sequences, and the uncoupled sequences of CB_t are assigned to the input vectors $u_{\mathcal{B}_{\mathrm{H}}}$, $u_{\mathcal{B}_{\mathrm{T}}}$, and $u_{\mathcal{A}\setminus\mathcal{B}}$, respectively. In the output vector $x_{[1:N]}$, the systematic bits corresponding to head coupled sequences (i.e., $x_{\mathcal{B}_{\mathrm{H}}} = u_{\mathcal{B}_{\mathrm{H}}}$) are punctured, and $x_{[1:N]\setminus\mathcal{B}_{\mathrm{H}}}$ is output as the codeword of CB_t. That is, although the coupled sequences are encoded by two CB encoders, they are only transmitted in one CB.

Bit-mapping

At time t, the codeword $\mathbf{x}_t$ is mapped to the equivalent subchannels $W^{(1:q)}$ of the 2^Q-ary symbols. When the direct BIPCM or the multi-kernel BIPCM is adopted, $\mathbf{x}_t$ is segmented into q sub-CBs $\mathbf{x}_{t,1}, \ldots, \mathbf{x}_{t,q}$ of equal length, sequentially w.r.t the indexing in $x_{[1:N]}$ in ascending order. Given a bit-mapper $\mathbf{s} = [s_1, \ldots, s_q]$, for $1 \leq i \leq q$, $\mathbf{x}_{t,i}$ is mapped to $W^{(s_i)}$. An example for the CB segmentation of direct PIC-BIPCM is as follows.

Example 9.2. For direct PIC-BIPCM, consider $N = 8$, $q = 2$, and $\mathcal{B}_{\mathrm{H}} = \{3, 4\}$. The first sub-CB $\mathbf{x}_{t,1}$ consists of $[x_1, x_2, x_5]$, and the second sub-CB $\mathbf{x}_{t,2}$ consists of $[x_6, x_7, x_8]$. ■

When the punctured BIPCM is used, given a bit-mapper $\mathbf{s} = [s_1, \ldots, s_{q'}]$ with $q' = 2^{\lceil \log_2 q \rceil}$, the i-th sub-codeword $x_{\left[\frac{(i-1)N}{q'}+1:\frac{iN}{q'}\right]}$ in $\mathbf{x}_t$ is punctured for $i \in \{i : s_i > q, 1 \leq i \leq q\}$. The resultant codeword $\mathbf{x}'_t$ is segmented into q sub-CBs. Then, these sub-CBs are mapped to $W^{(1:q)}$ according to the bit-mapper, respectively. An example for the CB segmentation and bit-mapping of punctured PIC-BIPCM is as follows.

Example 9.3. Consider $q = 3$ and $\mathbf{s} = [s_1, s_2, s_3, s_4]$. If $s_1 = 4$, $x_{[1:\frac{N}{4}]}$ in $\mathbf{x}_t$ is punctured to obtain the punctured codeword $\mathbf{x}'_t = x_{[\frac{N}{4}+1:N]\backslash\mathcal{B}_\mathrm{H}}$. Then, $\mathbf{x}'_t$ is decomposed into three sub-CBs $\mathbf{x}_{t,2}$, $\mathbf{x}_{t,3}$, and $\mathbf{x}_{t,4}$, which are mapped to $W^{(s_2)}$, $W^{(s_3)}$, and $W^{(s_4)}$, respectively. Likewise, if $s_2 = 4$, $\mathbf{x}'_t = x_{\left([1:\frac{N}{4}]\cup[\frac{N}{2}+1:N]\right)\backslash\mathcal{B}_\mathrm{H}}$. Then, $\mathbf{x}'_t$ is decomposed into $\mathbf{x}_{t,1}$, $\mathbf{x}_{t,3}$, and $\mathbf{x}_{t,4}$, which are mapped to $W^{(s_1)}$, $W^{(s_3)}$, and $W^{(s_4)}$, respectively. ■

The information length of component encoder is $K = \|\mathbf{u}_t\|$, and the component code rate is $R_0 = \frac{K}{N}$. The coupling length is $K_c = \sum_{i=1}^{m}\|\mathbf{u}_{t-i,t}\|$ with $\|\mathbf{u}_{t-i,t}\| = \frac{K_c}{m}$. We define $\lambda \triangleq \frac{K_c}{K} \in [0, 0.5]$ as the coupling ratio. The length of a codeword is $\|\mathbf{x}_t\| = N - K_c$. When punctured PIC-BIPCM is used, the length of a punctured codeword is $\|\mathbf{x}'_t\| = \frac{qN}{q'} - K_c$. Let $\widetilde{\mathbf{x}}_t$ represent the transmitted codeword, and $\|\widetilde{\mathbf{x}}_t\| = \widetilde{N} - K_c$, where $\widetilde{N} \le N$ and it depends on the adopted BIPCM scheme. Specifically, $\widetilde{\mathbf{x}}_t = \mathbf{x}_t$ and $\widetilde{N} = N$ for direct PIC-BIPCM and multi-kernel PIC-BIPCM, while $\widetilde{\mathbf{x}}_t = \mathbf{x}'_t$ and $\widetilde{N} = \frac{qN}{q'}$ for punctured PIC-BIPCM. Note that K_c needs to be an integer multiple of modulation order Q so that $\widetilde{N} - K_c$ is also an integer multiple of Q, which ensures that each CB can be mapped into an integer number of the modulated symbols. The code rate of the PIC-BIPCM with coupling ratio λ and coupling memory m is

$$\begin{aligned} R &= \frac{\sum_{t=1}^{L}\|\mathbf{u}'_t\| - \sum_{t=L-m+1}^{L}\sum_{i=L-t+1}^{m}\|\mathbf{u}_{t,t+i}\|}{\sum_{t=1}^{L}\|\widetilde{\mathbf{x}}_t\| - \sum_{t=L-m+1}^{L}\sum_{i=L-t+1}^{m}\|\mathbf{u}_{t,t+i}\|} \\ &= \frac{L(K - K_c) - \frac{(m+1)}{2}K_c}{L(\widetilde{N} - K_c) - \frac{(m+1)}{2}K_c} \\ &\overset{L\to\infty}{=} \frac{(1-\lambda)K}{\widetilde{N} - \lambda K}. \end{aligned} \tag{9.4}$$

Remark 9.1. In PIC scheme, although each coupled information sequence is encoded by two CB encoders, it is only transmitted in one CB. Thus, PIC codes exploit spatial coupling gain instead of repetition gain to obtain better performance than the uncoupled codes. ◀

9.3.2 Decoding of PIC-BIPCM

Now we introduce the decoding for PIC-BIPCM. The received PIC-BIPCM signal is firstly demodulated, where the demodulation steps are the same as the uncoupled BIPCM [156–158]. Then, the global decoding of the PIC-BIPCM is accomplished by a feed-forward and feedback (FF-FB) decoding iteratively [139]. Specifically, the FF-FB decoding employs a serial scheduling by decoding in the forward direction from CB_1 to CB_L during the FF phase and then starts decoding in the backward direction from CB_L to CB_1 during the FB phase. The component CB decoder can take arbitrary polar code decoding algorithms, as PIC does not modify the underlying encoding and decoding architecture. With the FF-FB decoding scheme, the global decoding process terminates when the number of CB decoding errors in the current iteration and the previous iteration are the same, or the maximum number of iterations is reached.

A windowed decoding scheme can also be employed for the global decoding. The window size satisfies $L_W \geq 2m + 1$. At time t, the decoding window is from CB_t to CB_{t+L_W-1}. The decoder performs forward and backward decoding operations within the decoding window. After that, the receiver outputs the hard-decision estimation of $\mathbf{u}'_t$, and moves the decoding window forward. It is observed that when L_W is sufficiently large, the windowed decoding scheme can have a similar error performance as the FF-FB decoding scheme.

Now, we introduce the CB decoding. At time t and $i = 1, \dots, m$, let $\mathbf{y}_{t-i,t}$, $\mathbf{y}_{t,t}$, and $\mathbf{y}_{t,t+i}$ represent the LLR associated with the information sequences $\mathbf{u}_{t-i,t}$,

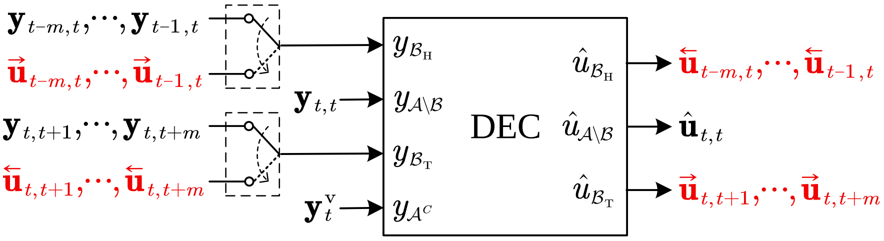

$\mathbf{y}_{t-m,t},\cdots,\mathbf{y}_{t-1,t}$
$\overrightarrow{\mathbf{u}}_{t-m,t},\cdots,\overrightarrow{\mathbf{u}}_{t-1,t}$
$\mathbf{y}_{t,t+1},\cdots,\mathbf{y}_{t,t+m}$
$\overleftarrow{\mathbf{u}}_{t,t+1},\cdots,\overleftarrow{\mathbf{u}}_{t,t+m}$
$\mathbf{y}_{t,t}$
$\mathbf{y}_t^{\mathrm{v}}$
$y_{\mathcal{B}_{\mathrm{H}}}$
$y_{\mathcal{A}\backslash\mathcal{B}}$
$y_{\mathcal{B}_{\mathrm{T}}}$
$y_{\mathcal{A}^c}$
DEC
$\hat{u}_{\mathcal{B}_{\mathrm{H}}}$
$\hat{u}_{\mathcal{A}\backslash\mathcal{B}}$
$\hat{u}_{\mathcal{B}_{\mathrm{T}}}$
$\overleftarrow{\mathbf{u}}_{t-m,t},\cdots,\overleftarrow{\mathbf{u}}_{t-1,t}$
$\hat{\mathbf{u}}_{t,t}$
$\overrightarrow{\mathbf{u}}_{t,t+1},\cdots,\overrightarrow{\mathbf{u}}_{t,t+m}$

Figure 9.8: CB decoder block diagram at time t.

$\mathbf{u}_{t,t}$, and $\mathbf{u}_{t,t+i}$, respectively. Let $\mathbf{y}_t^{\mathrm{v}}$ represent the LLR associated with the parity sequence $\mathbf{v}_t$. Furthermore, let $\overrightarrow{\mathbf{u}}_{t-i,t}$ denote the extrinsic information (i.e., the hard-decision estimation of $\mathbf{u}_{t-i,t}$) passed from CB$_{t-i}$ to CB$_t$, and $\overleftarrow{\mathbf{u}}_{t,t+i}$ denote the extrinsic information passed from CB$_{t+i}$ to CB$_t$, respectively. We use d_t to denote the decoding status of CB$_t$, where $d_t = 0$ refers to that the decoding of CB$_t$ has failed, or CB$_t$ has not been decoded yet, and $d_t = 1$ refers to that the decoding of CB$_t$ is successful.

As shown in Fig. 9.8, the component CB decoder has an N-bit input $y_{[1:N]}$, where $y_{\mathcal{A}}$ takes the channel LLR of $\mathbf{u}_{t,t}$, $y_{\mathcal{A}^c}$ takes the channel LLR of $\mathbf{v}_t$, $y_{\mathcal{B}_{\mathrm{H}}}$ and $y_{\mathcal{B}_{\mathrm{T}}}$ are associated with the head coupled sequences and tail coupled sequences, respectively. In specific, for $1 \leq i \leq m$, the decoder takes

$$y_{\mathcal{B}_{\mathrm{H},i}} = \begin{cases} \mathbf{y}_{t-i,t} \ , & \text{if } d_{t-i} = 0 \\ \infty \cdot (1 - 2\overrightarrow{\mathbf{u}}_{t-i,t}) \ , & \text{if } d_{t-i} = 1 \end{cases}, \tag{9.5a}$$

$$y_{\mathcal{B}_{\mathrm{T},i}} = \begin{cases} \mathbf{y}_{t,t+i} \ , & \text{if } d_{t+i} = 0 \\ \infty \cdot (1 - 2\overleftarrow{\mathbf{u}}_{t,t+i}) \ , & \text{if } d_{t+i} = 1 \end{cases} \tag{9.5b}$$

as inputs, respectively. In other words, $y_{\mathcal{B}_{\mathrm{H},i}}$[1] only takes the extrinsic information from CB$_{t-i}$ if no errors are detected in CB$_{t-i}$. If errors are detected in CB$_{t-i}$,

---

[1] For $m \geq 1$, $\mathcal{B}_{\mathrm{H}} = \{\mathcal{B}_{\mathrm{H},1}, \ldots, \mathcal{B}_{\mathrm{H},m}\}$ and $\mathcal{B}_{\mathrm{T}} = \{\mathcal{B}_{\mathrm{T},1}, \ldots, \mathcal{B}_{\mathrm{T},m}\}$. At time t and $i = 1, \ldots, m$, $\mathcal{B}_{\mathrm{H},i}$ represent the indices subset for head coupling sequence $\mathbf{u}_{t-i,t}$, and $\mathcal{B}_{\mathrm{T},i}$ represent the indices subset for head coupling sequence $\mathbf{u}_{t,t+i}$.

$y_{\mathcal{B}_{\mathrm{H},i}}$ only takes the corresponding channel LLRs as the decoder input. The same procedures also apply for $y_{\mathcal{B}_{\mathrm{T},i}}$. After decoding, the receiver decomposes the hard-decision estimation of $\mathbf{u}_t$ into $\overleftarrow{\mathbf{u}}_{t-m,t}, \ldots, \overleftarrow{\mathbf{u}}_{t-1,t}, \hat{\mathbf{u}}_{t,t}, \overrightarrow{\mathbf{u}}_{t,t+1}, \ldots, \overrightarrow{\mathbf{u}}_{t,t+m}$. Later, for $1 \leq i \leq m$, $\overleftarrow{\mathbf{u}}_{t-i,t}$ will be passed to CB_{t-i}, and $\overrightarrow{\mathbf{u}}_{t,t+i}$ will be passed to CB_{t+i}, respectively.

9.4 Performance Analysis of PIC-BIPCM

In this section, we analyze the CBER performance of the PIC-BIPCM for any given coupling indices and bit-mapper over the BEC. We first review the density evolution (DE) of finite-length polar codes over the BEC [27]. Then, by looking into the graph model of the PIC-BIPCM, we derive a closed-form expression of CBER for the PIC-BIPCM with $m = 1$ over the BEC, which is also a CBER upper bound of the PIC-BIPCM with $m > 1$. We also present a CBER lower bound of the PIC-BIPCM with $m \geq 1$. The CBER analysis over the BEC will provide a guideline for designing coupling indices and bit-mapper over the AWGN channel. In Sec. V, we will show that the CBER analysis over the BEC is effective for the PIC-BIPCM design over the AWGN channel.

9.4.1 Density Evolution Analysis of Polar Codes

As we have reviewed in Chapter 7, polar codes have a layered structure. Consider a polar code with codeword length $N = 2^n$ and information indices $\mathcal{A}$. The encoder takes $u_{[1:N]}$ and generate codeword $x_{[1:N]}$. For $\tau = 0, \ldots, n$ and $i = 1, \ldots, N$, the vector channel at layer τ is denoted by W_{2^τ}, and the i-th subchannel in this layer is denoted by $W_{2^\tau}^{(i)}$. At layer 0, $W_1^{(i)}$ corresponds to the i-th coded bits x_i. At layer n, $W_N^{(i)}$ corresponds to the i-th input (information/frozen) bit u_i. Let

$\epsilon_{2^\tau}^{(i)}$ denote the erasure probability of $W_{2^\tau}^{(i)}$, and let $\epsilon_{2^\tau}^{(1:N)}$ collectively denote the erasure probabilities of $W_{2^\tau}^{(1)}, \ldots, W_{2^\tau}^{(N)}$.

The decoding of the polar code starts from layer 0 and terminates at layer n. To analyze the decoding performance of SC decoder, we initialize $\epsilon_1^{(1:N)}$ with the erasure probability of the transmission channel. Knowing that the DE equations corresponding to channel transform $(W_N^{(i)}, W_N^{(j)}) \to (W_{2N}^{(i)}, W_{2N}^{(j)})$ is [27]

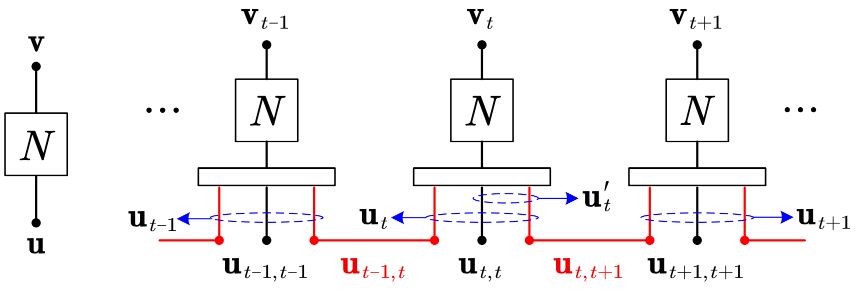
$$\begin{cases} \epsilon_{2N}^{(i)} = 1 - \left(1 - \epsilon_N^{(i)}\right)\left(1 - \epsilon_N^{(j)}\right) \\ \epsilon_{2N}^{(j)} = \epsilon_N^{(i)} \epsilon_N^{(j)} \end{cases}, \tag{9.6}$$

we can track the evolution of erasure probability from layer 0 to layer n, and obtain $\epsilon_N^{(1:N)}$ in layer n. The error probability of u_i is $p_N^{(i)} = \frac{\epsilon_N^{(i)}}{2}$. The block error probability P_e of the polar code is bounded by $P_e \leq 1 - \prod_{i \in \mathcal{A}} (1 - p_N^{(i)})$. As this upper bound is tight at a high channel capacity, we will use it as the approximation of the block error probability and evaluate the CBER performance bounds of the PIC-BIPCM in the following subsection.

9.4.2 Graph Model for the PIC-BIPCM with $m = 1$

To analyze the error performance of PIC-BIPCM, we first start with the graph model representation of uncoupled polar codes. Following the graph model for turbo codes in [38], we plot the simplified factor graph of uncoupled systematic polar codes in Fig. 9.9a. In the factor graph, all systematic bits are represented by a single variable node, and all parity bits are represented by a single parity node. The multi-layer parity check structure of a length-N polar code is represented by a single function node.

For the factor graph of PIC-BIPCM, we treat the uncoupled sequences, head coupled sequences, and tail coupled sequences separately. Fig. 9.9b shows the



(a) (b)

Figure 9.9: Factor graph of (a) length-N polar codes, and (b) the PIC-BIPCM with $m = 1$ constructed by length-N polar codes.

factor graph of the PIC-BIPCM with coupling memory $m = 1$. In CB$_t$, we use three variable nodes to represent $\mathbf{u}_{t,t}$, $\mathbf{u}_{t-1,t}$, and $\mathbf{u}_{t,t+1}$, respectively. As $\mathbf{u}_{t-1,t}$ is shared between CB$_t$ and CB$_{t-1}$, we connect the variable nodes corresponding to $\mathbf{u}_{t-1,t}$ in CB$_{t-1}$ and that in CB$_t$ with a horizontal line, meaning that $\mathbf{u}_{t-1,t}$ is encoded twice. Similarly, we connect the variable nodes corresponding to $\mathbf{u}_{t,t+1}$ in CB$_t$ and that in CB$_{t+1}$ with a horizontal line. During the decoding process, $\overrightarrow{\mathbf{u}}_{t-1,t}$ and $\overleftarrow{\mathbf{u}}_{t-1,t}$ are passed between CB$_{t-1}$ and CB$_t$ via node $\mathbf{u}_{t-1,t}$, and $\overrightarrow{\mathbf{u}}_{t,t+1}$ and $\overleftarrow{\mathbf{u}}_{t,t+1}$ are passed between CB$_t$ and CB$_{t+1}$ via node $\mathbf{u}_{t,t+1}$, respectively.

9.4.3 CBER Analysis for the PIC-BIPCM with $m = 1$

In this subsection, we analyze the CBER of the PIC-BIPCM with coupling memory $m = 1$ and decoded by the FF-FB decoding scheme. In this case, only one FF phase and one FB phase are required. When the decoding of CB$_t$ fails in the FB phase of the first iteration, the receiver would not pass extrinsic information $\overleftarrow{\mathbf{u}}_{t-1,t}$, $\overrightarrow{\mathbf{u}}_{t,t+1}$ to their destination CBs. In such case, even if the receiver processes the FF-FB decoding for more iterations, the decoding error of CB$_t$ would be the same. Therefore, the CBER of the PIC-BIPCM with $m = 1$ can be determined by the probability that the decoding of CB$_t$ fails in the FB phase of the first iteration[2], assuming that L approaches infinity and CB_t is located in the middle of the coupling chain.

The error probability of CB_t is jointly determined by the subchannel capacity, the error probability of CB_{t-1} in the FF phase, and the error probability of CB_{t+1} in the FB phase. Specifically, we consider that the receiver processes decoding in three instances as follows:

- Instance 1: Receiver decodes in the forward direction from CB_1 to CB_{t-1}.
- Instance 2: Receiver decodes in the backward direction from CB_L to CB_{t+1}.
- Instance 3: Receiver decodes CB_t.

The error probability of CB_t at instance 3 determines the CBER of PIC-BIPCM with FF-FB decoding scheme. Let $P^{\mathrm{F}}_{e,t-1}$ denote the error probability of CB_{t-1} at instance 1, and $P^{\mathrm{B}}_{e,t+1}$ denote the error probability of CB_{t+1} at instance 2, respectively. Let $E_{d_{t-1},d_{t+1}}$[3] be the conditional probability that the decoding of CB_t fails when the availability of extrinsic information is given by d_{t-1} and d_{t+1}. Note that this conditional error probability depends on not only d_{t-1} and d_{t+1}, but also subchannel capacities, the information indices, the coupling indices, and the bit-mapper. The error probability of CB_t at instance 3 is

$$\begin{aligned} P_{e,t} = \sum_{d_{t-1}\in\{0,1\}} \sum_{d_{t+1}\in\{0,1\}} & \left(d_{t-1} + (-1)^{d_{t-1}} \cdot P^{\mathrm{F}}_{e,t-1}\right) \\ & \cdot \left(d_{t+1} + (-1)^{d_{t+1}} \cdot P^{\mathrm{B}}_{e,t+1}\right) \cdot E_{d_{t-1},d_{t+1}}. \end{aligned} \tag{9.7}$$

---

[2] We assume that all CBs have the same error probability. In fact, due to the initialization and termination of the coupling, the CBs closed to each end of the TB have slightly lower error probabilities. When L approaches infinity, the impact of initialization and termination on the CBER becomes negligible.

[3] For the simplicity of presentation, when $d_{t-1} = a$ and $d_{t+1} = b$ is given for any $a, b \in \{0, 1\}$, we use $E_{a,b}$ to represent $E_{d_{t-1}=a,d_{t+1}=b}$. For instance, $E_{0,0}$ represents $E_{d_{t-1}=0,d_{t+1}=0}$.

At instance 1, the decoding starts from CB_1. The initial error probability of CB_1 is $P_{e,1}^{\mathrm{F}} = E_{1,0}$, as it is known that $\mathbf{u}_{0,1} = \mathbf{0}$. For $1 < t' < t$, the forward error probability of $\mathrm{CB}_{t'}$ is

$$\begin{aligned} P_{e,t'}^{\mathrm{F}} =& \sum_{d_{t'-1}\in\{0,1\}} \left(d_{t'-1} + (-1)^{d_{t'-1}} \cdot P_{e,t'-1}^{\mathrm{F}}\right) \cdot E_{d_{t'-1},0} \\ =& P_{e,t'-1}^{\mathrm{F}} \cdot E_{0,0} + (1 - P_{e,t'-1}^{\mathrm{F}}) \cdot E_{1,0}. \end{aligned} \tag{9.8}$$

When t' is small, $P_{e,t'}^{\mathrm{F}}$ increases with t'. When $t' \gg 1$, $P_{e,t'}^{\mathrm{F}}$ becomes a constant. To find the steady-state error probability under forward decoding, we can drop t' from the CB error probability by simply substituting $P_{e,t'}^{\mathrm{F}} = P_{e,t'-1}^{\mathrm{F}} = P_e^{\mathrm{F}}$ into (9.8). This gives

$$P_e^{\mathrm{F}} = \frac{E_{1,0}}{1 - E_{0,0} + E_{1,0}}. \tag{9.9}$$

At instance 2, the decoding starts from CB_L with initial error probability $P_{e,L}^{\mathrm{B}} = E_{0,1}$. For $t < t' < L$, the backward error probability of decoding $\mathrm{CB}_{t'}$ is

$$\begin{aligned} P_{e,t'}^{\mathrm{B}} =& \sum_{d_{t'+1}\in\{0,1\}} \left(d_{t'+1} + (-1)^{d_{t'+1}} \cdot P_{e,t'+1}^{\mathrm{B}}\right) \cdot E_{0,d_{t'+1}} \\ =& P_{e,t'+1}^{\mathrm{B}} \cdot E_{0,0} + (1 - P_{e,t'+1}^{\mathrm{B}}) \cdot E_{0,1}. \end{aligned} \tag{9.10}$$

Similar to instance 1, $P_{e,t'}^{\mathrm{B}}$ becomes a constant when $t' \ll L$. By substituting $P_{e,t'}^{\mathrm{B}} = P_{e,t'+1}^{\mathrm{B}} = P_e^{\mathrm{B}}$ into (9.10), the steady-state backward error probability can be obtained by

$$P_e^{\mathrm{B}} = \frac{E_{0,1}}{1 - E_{0,0} + E_{0,1}}. \tag{9.11}$$

At instance 3, substituting $P^{\mathrm{F}}_{e,t-1} = P^{\mathrm{F}}_{e}$ and $P^{\mathrm{B}}_{e,t+1} = P^{\mathrm{B}}_{e}$ into (9.7) gives

$$P_{e,t} = \frac{E_{1,1}(1 - E_{0,0})^2 + E_{1,0}E_{0,1}(2 - E_{0,0})}{(1 + E_{1,0} - E_{0,0})(1 + E_{0,1} - E_{0,0})}. \tag{9.12}$$

To obtain the theoretical CBER of the PIC-BIPCM with coupling memory $m = 1$, information indices $\mathcal{A}$, coupling indices $\{\mathcal{B}_{\mathrm{H}}, \mathcal{B}_{\mathrm{T}}\}$, and bit-mapper $\mathbf{s}$ over channel $W^{(1:q)}$, we use density evolution to calculate $E_{d_{t-1},d_{t+1}}$ for $d_{t-1}, d_{t+1} \in \{0, 1\}$, where the details are given in Alg. 9.1 in the Appendix. By substituting the values of $E_{d_{t-1},d_{t+1}}$ into (9.12), $P_{e,t}$ gives the theoretical CBER of the PIC-BIPCM. The expression (9.12) is also the CBER upper bound of the PIC-BIPCM with coupling memory $m > 1$, information indices $\mathcal{A}$, coupling indices $\{\mathcal{B}_{\mathrm{H}}, \mathcal{B}_{\mathrm{T}}\}$, and bit-mapper $\mathbf{s}$.

It is also worth noting that we can use $E_{1,1}$ as the CBER lower bound of the PIC-BIPCM with coupling memory $m \geq 1$, because the best CB decoding condition is that all extrinsic information are available. For a PIC-BIPCM with $m = 1$ and a PIC-BIPCM with $m > 1$, if they have the same information indices, coupling indices, and bit-mapper, they would have the same CBER lower bound. As we will see in Sec. 9.6, the CBER closed-form expression (9.12) and the CBER lower bound can tingly bound the simulated CBER over the BEC.

Remark 9.2. The CBER analysis for the PIC-BIPCM with $m = 1$ over the AWGN channel can be obtained by applying Gaussian approximation (GA) [116–118]. To be specific, knowing the channel signal-to-noise ratio (SNR), we can calculate the conditional probabilities $E_{d_{t-1},d_{t+1}}$ for $d_{t-1}, d_{t+1} \in \{0, 1\}$ by using GA. Then, we can obtain the theoretical CBER by substituting $E_{d_{t-1},d_{t+1}}$ into (9.12). We do not present the GA based analysis results in this chapter because the approximation error of GA will increase with the modulation order. In Sec. 9.6, we will demonstrate that the proposed design criteria based on the CBER analysis over the BEC are effective for the AWGN channel. ◀

9.5 Design of PIC-BIPCM

In the previous section, we have presented the CBER analysis of PIC-BIPCM over the BEC. In this section, we will design PIC-BIPCM by employing the CBER analysis as a design guideline. To be specific, we will first investigate the design criteria of coupling indices given an arbitrary bit-mapper. Then, we investigate the design of the bit-mapper by taking into account the proposed coupling indices design criteria.

9.5.1 Design of PIC-BIPCM Coupling Indices

In this subsection, we discuss the design criteria of coupling indices $\{\mathcal{B}_{\mathrm{H}}, \mathcal{B}_{\mathrm{T}}\}$ for a given bit-mapper. We first verify that the design criterion proposed for PIC-PCs in [139] can still hold for the PIC-BIPCM. Then, we discuss the design of coupling indices w.r.t how PIC affects the BIPCM bit-mapping.

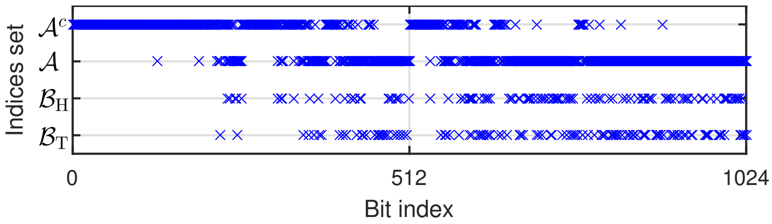
Design of Coupling Indices w.r.t Information Bit Error Probability

The first aspect we need to consider for coupling indices design is the partial polarization of finite-length polar codes. In a finite-length polar codeword, some information bits have significantly higher error probabilities than others. Hence, the error probabilities of these unreliable information bits dominate the block error probability of the polar codes. As shown in [139], for a binary-input channel, PIC-PCs provide significant CBER performance improvement by sharing the unreliable information bits between CBs. Coupling reliable information bits only brings minor performance improvement because the uncoupled unreliable information bits obtain less spatial coupling gain and their error probabilities dominate the CB error probability. With the CBER analysis in Sec. 9.4.3, it is easy to show that this criterion still holds for PIC-BIPCM. In Sec. 9.6, we will show the CBER performance with different coupling indices to confirm the effectiveness of using this criterion.

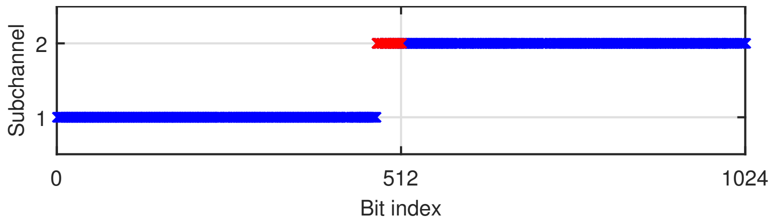
Design of Coupling Indices w.r.t Bit-mapping

The second aspect we need to consider for coupling indices design is the BIPCM bit-mapping. We note that the head coupled bits of a CB are punctured before bit-mapping. However, the information indices $\mathcal{A}$ are chosen based on the bit-mapping which assumes that head coupled bits are not punctured as the coupling indices are not known at this stage. Consequently, the actual bit-mapping becomes different from the bit-mapping used for choosing information indices. This mismatch could lead to a potential performance degradation for the PIC-BIPCM. To preserve the expected bit-mapping, we want to ensure that the information bits shared between two CBs are located in the same sub-codewords in each of the CBs. To explain it in detail, an example is given as follows.

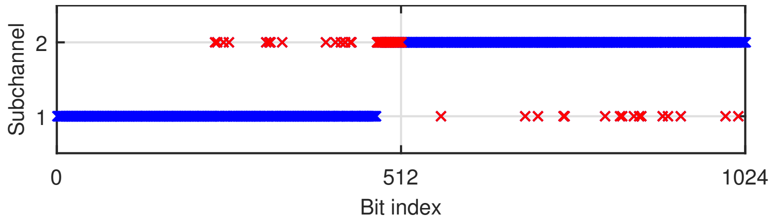
Example 9.4. Fig. 9.10 shows an example of how coupling indices affect the bit-mapping of a direct PIC-BIPCM with $N = 1024$, $R = \frac{1}{2}$, $\lambda = 0.25$, and $m = 1$. We consider that a set of $q = 2$ parallel BECs $W^{(1:2)}$ with erasure probability $\{0.2, 0.4\}$, respectively. The bit-mapper is $\mathbf{s} = [1, 2]$. At the code construction stage, we first compute the error probability of input vector $u_{[1:1024]}$ via density evolution, expecting that the two sub-codewords $x_{[1:512]}$ and $x_{[513:1024]}$ will be transmitted via $W^{(1)}$ and $W^{(2)}$, respectively. Based on the error probability of $u_{[1:1024]}$, we first choose the information indices $\mathcal{A}$, where $\|\mathcal{A}\| = 569$. Then, coupling indices $\mathcal{B}_{\mathrm{H}}$ and $\mathcal{B}_{\mathrm{T}}$ are selected as two disjoint subsets of $\mathcal{A}$, and



(a) Indices sets.



(b) Bit-mapping at transmitter.



(c) Bit-mapping at receiver.

Figure 9.10: An example of how coupling indices affects PIC-BIPCM bit-mapping when randomly selected coupling indices is used. In (b) and (c), the bits with mismatched bit-mapping is highlighted in red.

$\|\mathcal{B}_\mathrm{H}\| = \|\mathcal{B}_\mathrm{T}\| = 114$. The indices sets employed in this example are shown in Fig. 9.10a. Due to the polarization property of polar codes, the information indices are not evenly distributed among the two sub-codewords. Consequently, the coupling indices are not evenly distributed either.

Consider the PIC-BIPCM encoding at the transmitter side. During the encoding for CB_t, the head coupled sequence $x_{\mathcal{B}_\mathrm{H}}$ are punctured from $x_{[1:1024]}$. The codeword of CB_t (i.e., $x_{[1:1024]\setminus\mathcal{B}_\mathrm{H}}$) is segmented into two sub-CBs of equal length, and each sub-CB is mapped to one of the subchannels. As shown in Fig. 9.10a, the number of head coupled bits in $x_{[1:512]}$ is significantly less than that in $x_{[513:1024]}$. As a consequence, in the example shown in Fig. 9.10b, the last 30 bits in $x_{[1:512]\backslash\mathcal{B}_{\mathrm{H}}}$ are mapped to $W^{(2)}$, even though they are expected to be mapped to $W^{(1)}$ when designing the coupling indices.

At the receiver, the decoder can obtain the channel LLR of the head coupled bits of CB_t from CB_{t-1}. As the coupled indices are randomly selected, the head coupled bits of CB_t could be mapped to any subchannel when they are transmitted as the tail coupled bits of CB_{t-1}. As a result, the number of coded bits whose actual bit-mapping does not match with their expected bit-mapping ($x_{[1:512]}$ and $x_{[513:1024]}$ are transmitted via $W^{(1)}$ and $W^{(2)}$, respectively) is non-negligible. Such an example is shown in Fig. 9.10c, where 52 coded bits in $x_{[1:512]}$ are mapped to $W^{(2)}$, while 16 coded bits in $x_{[513:1024]}$ are mapped to $W^{(1)}$. ■

For a set of parallel channel $W^{(1:q)}$, let $K_{\mathrm{H},i}$ represent the number of head coupled bits located in the i-th sub-codeword $x_{\left[(i-1)\cdot\frac{N}{q}+1:i\cdot\frac{N}{q}\right]}$. To avoid the mismatch of bit-mapping, ideally, the coupling indices should satisfy the following criteria:

1. For any $1 \le i \le q$, the head coupled bits in the i-th sub-codeword of CB_t should come from the i-th sub-codeword of $\mathrm{CB}_{t-m}, \ldots, \mathrm{CB}_{t-1}$.
2. For any $1 \le i, j \le q$ and $i \neq j$, the head coupled bits in the i-th sub-codeword and the j-th sub-codeword should have the same length. i.e., $K_{\mathrm{H},i} = K_{\mathrm{H},j}$.

However, as the information bits are not evenly distributed among all the sub-codewords, so the length of head coupled sequences in each sub-codewords might not be the same. With the first criterion, we minimize the number of bits with mismatched bit-mapping.

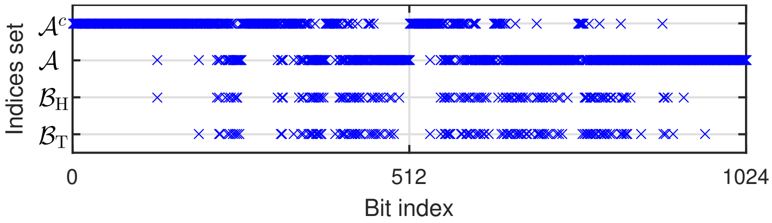
By jointly considering the two criteria discussed above, we propose the following method of choosing coupling indices:

1. Given $W^{(1:q)}$, compute the bit-channel error probability $p_N^{(i)}$ for $1 \leq i \leq N$ with density evolution.
2. Choose information indices $\mathcal{A}$ of length K satisfying $p_N^{(i)} < p_N^{(j)}$ for any $i \in \mathcal{A}$ and $j \in \mathcal{A}^c$.
3. Choose $\mathcal{B}$ of length $2K_c$ as a subset of $\mathcal{A}$ such that $p_N^{(i)} > p_N^{(j)}$ for any $i \in \mathcal{B}$ and $j \in \mathcal{A} \setminus \mathcal{B}$.
4. Sort the indices in $\mathcal{B}$ in ascending order, resulting in the sorted indices $[b_1, b_2, \ldots, b_{2K_c}]$.
5. Choose the coupling indices of the PIC-BIPCM as $\mathcal{B}_{\mathrm{H}} = [b_1, b_3, \ldots, b_{2K_c-1}]$, and $\mathcal{B}_{\mathrm{T}} = [b_2, b_4, \ldots, b_{2K_c}]$, respectively.

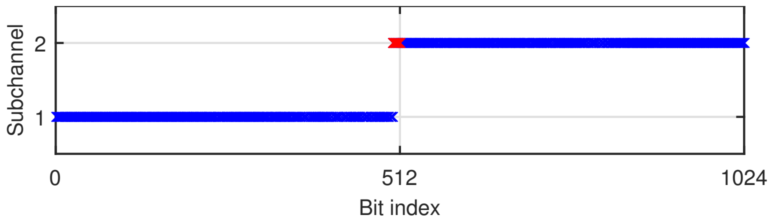
Consider Example 4 again. We show the indices sets and bit-mapping when the proposed coupling indices design criteria are adopted in Fig. 9.11. It is clear that the number of bits with mismatched bit-mapping is greatly reduced for "U-U, sorted" in Fig. 9.11c compared to "rand" in Fig. 9.10c. In the following subsection, we design bit-mappers by taking the proposed coupling indices design criteria into account.

9.5.2 Design of PIC-BIPCM Bit-mapper

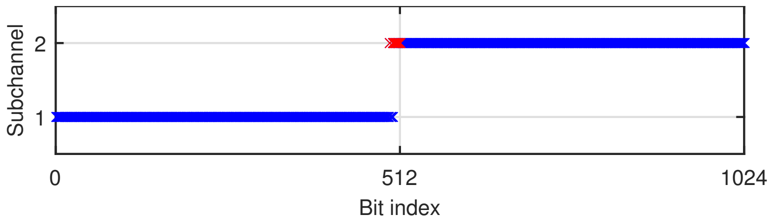
In this subsection, we propose a method of designing a proper bit-mapper for PIC-BIPCM while incorporating the design criteria of coupling indices. We consider PIC-BIPCM constructed with direct BIPCM, punctured BIPCM, and multi-kernel BIPCM, respectively. To find a bit-mapper that can provide good CBER performance of PIC-BIPCM, we first search for a candidate set consisting of bit-mappers leading to good error performance for uncoupled BIPCM. For each



(a) Indices sets.



(b) Bit-mapping at transmitter.



(c) Bit-mapping at receiver.

Figure 9.11: An example of how coupling indices affects PIC-BIPCM bit-mapping the coupling indices selected with the proposed design criteria. In (b), and (c), the bits with mismatched bit-mapping is highlighted in red.

bit-mapper in the candidate set, we choose the information indices and coupling indices, respectively. Then, we use the CBER closed-form expression of PIC-BIPCM derived in Sec. 9.4 to evaluate the performance of each bit-mapper and search a bit-mapper that results in the lowest CBER. All designed bit-mappers for various modulation orders are reported in Table 9.2 and the constellation labels are given in Fig. 9.20 in the Appendix. The detailed design methods are as follows.

In the following, we give an example of designing the bit-mapper for Gray-labeled 256-QAM over the AWGN channel.

Table 9.2: Designed bit-mappers for PIC-BIPCM with $R = \frac{1}{2}$.

Modulation	BIPCM scheme	N	Bit-mapper
32-QAM	punctured BIPCM	1024	$[6, 7, 8, 1, 2, 3, 5, 4]$
	multi-kernel BIPCM	640	$[1, 2, 3, 5, 4]$
64-QAM	punctured BIPCM	1024	$[4, 3, 1, 2]$
	multi-kernel BIPCM	768	$[3, 2, 1]$
256-QAM	direct BIPCM	1024	$[4, 3, 1, 2]$
4-BEC	direct BIPCM	1024	$[4, 3, 1, 2]$

Bit-mapper Design for Direct PIC-BIPCM

For $W^{(1:q)}$, there are $q!$ possible bit-mappers. For any given bit-mapper, we can find a group of $\prod_{i=1}^{n} 2^i$ equivalent bit-mappers which lead to the same information indices as well as the same error performance [131, 158] for uncoupled direct BIPCM. Given the code design parameters, we design the bit-mapper for direct PIC-BIPCM with the following steps.

1. Search the bit-mapper $\mathbf{s}^*$ that gives the best error performance for the uncoupled BIPCM with the given codeword length, code rate, and subchannel capacities.
2. Find the equivalent bit-mappers set $\mathcal{S}^*$ such that all the bit-mappers in $\mathcal{S}^*$ are equivalent to $\mathbf{s}^*$ for the uncoupled BIPCM.
3. With bit-mapper $\mathbf{s}^*$, choose the coupling indices $\{\mathcal{B}_{\mathrm{H}}, \mathcal{B}_{\mathrm{T}}\}$ with the design criteria proposed in Sec. 9.5.1.
4. For each bit-mapper in $\mathcal{S}^*$, compute the CBER for the PIC-BIPCM with the coupling indices chosen in Step 3.
5. Find the bit-mapper in $\mathcal{S}^*$ which gives the lowest CBER among all other bit-mappers.

Example 9.5. Consider a set of 4 parallel BECs (4-BEC) $W^{(1:4)}$ whose subchannel capacities equal to the subchannel capacities of the AWGN channel with Gray-labeled 256-QAM. The erasure rate of $W^{(i)}$ is denoted by $\epsilon^{(i)}$, and $\epsilon^{(1)} < \epsilon^{(2)} < \epsilon^{(3)} < \epsilon^{(4)}$.

For the uncoupled direct BIPCM with $q = 4$, there are 24 possible bit-mappers. Theses bit-mappers can be divided into 3 equivalent sets, where each set contains 8 equivalent bit-mappers. To find the bit-mapper set that gives the best error performance for the uncoupled BIPCM, we only need to pick one bit-mapper from each of the equivalent sets and compare their error performance. Consider $N = 1024$ and $R = \frac{1}{2}$, the bit-mapper set with the best error performance is

$$\begin{aligned}\mathcal{S}^* = \{&[1,2,4,3],[1,3,4,2],[2,1,3,4],[2,4,3,1],\\ &[3,1,2,4],[3,4,2,1],[4,2,1,3],[4,3,1,2]\}.\end{aligned}$$

The theoretical CBER of the PIC-BIPCM with $N = 1024$, $R = \frac{1}{2}$, $\lambda = 0.25$, $m = 1$, and the above equivalent bit-mappers over the BEC are plotted in Fig. 9.12, respectively, where the theoretical CBER are calculated with the closed-form expression (9.12) derived in Sec. 9.4.3. For the ease of presentation, instead of labeling all the subchannel capacities of $W^{(1:4)}$ in the figure, we only label the average capacity $\bar{I}(W^{(1:4)}) = \sum_{i=1}^{q} \frac{1-\epsilon^{(i)}}{q}$. The PIC-BIPCM with bit-mapper $[4,3,1,2]$ has the lowest CBER when $\bar{I}(W^{(1:4)})$ is greater than 0.63, which means that $[4,3,1,2]$ should outperform all other bit-mappers over the AWGN channel with 256-QAM at a high SNR. In Sec. 9.6, we will present the simulation results to show that $[4,3,1,2]$ has good CBER performance over the AWGN channel. ■

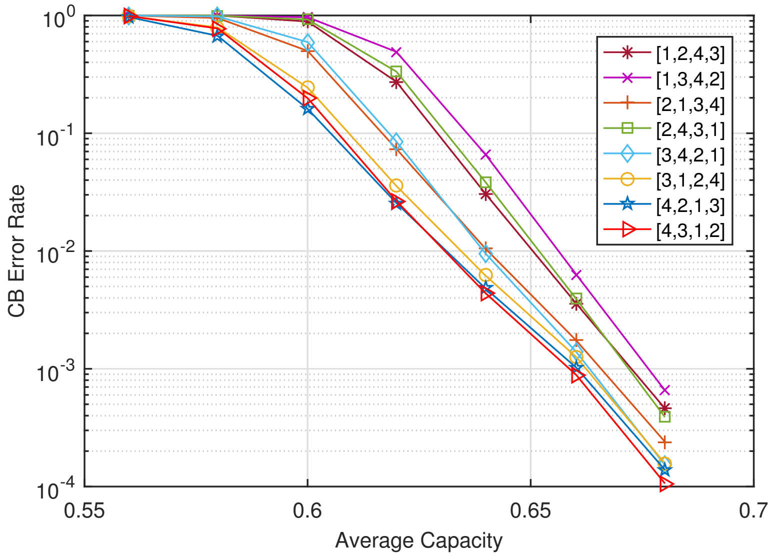


Figure 9.12: Theoretical CBER of $(1024, \frac{1}{2}, 0.25, 1)$ PIC-BIPCM with different bit-mappers over 4-BEC.

Bit-mapper Design for Punctured PIC-BIPCM

For punctured BIPCM, we treat $W^{(1:q)}$ as a set of q' parallel subchannels $W^{(1:q')}$, where $W^{(q:q')}$ are the zero-capacity virtual subchannels. Let $\mathbf{s} = [s_1, \ldots, s_{q'}]$ represent the bit-mapper for punctured BIPCM. For a codeword $x_{[1:N]}$, if $s_i \leq q$, $x_{\left[\frac{(i-1)\cdot N}{q'}+1:\frac{i\cdot N}{q'}\right]}$ is transmitted via $W^{(s_i)}$. Otherwise, if $s_i > q$, $x_{\left[\frac{(i-1)\cdot N}{q'}+1:\frac{i\cdot N}{q'}\right]}$ is punctured. Let $\mathcal{P}$ represent the indices set of punctured coded bits, and recall that $\mathcal{A}$ is the indices set of information bits. To promise a good CBER performance of punctured PIC-BIPCM, we want to ensure that $\mathcal{A}$ does not intersect $\mathcal{P}$. The reasons are as follows.

For the encoding of punctured PIC-BIPCM, the codeword of CB_t is given by $\mathbf{x}_t = x_{[1:N]\setminus\mathcal{B}_\mathrm{H}}$, and then $\widetilde{\mathbf{x}}_t = x_{[1:N]\setminus\{\mathcal{B}_\mathrm{H}\cup\mathcal{P}\}}$ is transmitted. To ensure that $\|\widetilde{\mathbf{x}}_t\|$ is a constant, $\mathcal{B}_\mathrm{H}$ should not intersect with $\mathcal{P}$, i.e., $\mathcal{B}_\mathrm{H} \cap \mathcal{P} = \emptyset$. When $\mathcal{A}$ intersects with $\mathcal{P}$ while $\mathcal{B}_\mathrm{H} \cap \mathcal{P} = \emptyset$, some unreliable information bits are punctured and thus are not coupled. This leads to a potential performance loss because we want to couple the unreliable information bits to improve their error probabilities as they dominate the CB error probability. Therefore, $\mathcal{A}$ should not intersect with $\mathcal{P}$, i.e., $\mathcal{A} \cap \mathcal{P} = \emptyset$.

According to [129, 130], for a length-N polar code, puncturing $q' - q$ sub-codewords from $x_{[1:N]}$ results in $q' - q$ un-decodable sets, where each set consists of $\frac{N}{q'}$ un-decodable bits whose error probability is always 1. Let $\mathcal{U}$ represent the indices set of un-decodable bits. As $\mathcal{A} \cap \mathcal{U} = \emptyset$, by adopting a bit-mapper which satisfies $\mathcal{P} = \mathcal{U}$, we can ensure that $\mathcal{A} \cap \mathcal{P} = \emptyset$. In summary, the bit-mapper of the punctured PIC-BIPCM is designed as follows:

1. Search for a set of bit-mappers $\mathcal{S}$ which satisfy $\mathcal{P} = \mathcal{U}$.
2. Find a set of bit-mappers $\mathcal{S}^* \subset \mathcal{S}$ which gives the best error performance for uncoupled BIPCM among all other bit-mapper sets.
3. For each bit-mapper in $\mathcal{S}^*$, choose the coupling indices according to the design criteria proposed in Sec. 9.5.1 and compute the CBER for the corresponding PIC-BIPCM.
4. Find the bit-mapper in $\mathcal{S}^*$ which has the lowest CBER among all other bit-mappers.

In the following, we give the design examples for Gray-labeled 64-QAM and quasi-Gray-labeled 32-QAM, considering $N = 1024$ and $R = \frac{1}{2}$.

Example 9.6. Consider 64-QAM, where $q = 3$ and $q' = 4$. Due to the structural property of polarization kernel $\mathbf{F}$, when an arbitrary sub-codeword of $x_{[1:N]}$ is punctured, $u_{[1:\frac{N}{4}]}$ is always un-decodable. This means that the bit-mappers which map $x_{\left[1:\frac{N}{4}\right]}$ to $W^{(4)}$ can promise $\mathcal{P} = \mathcal{U}$, i.e., $\mathcal{S} = \{\mathbf{s} | s_1 = 4\}$. Then, we have $\mathcal{S}^* = \{[4, 3, 1, 2], [4, 2, 1, 3]\}$. By comparing the CBER of the corresponding PIC-BIPCM, we find that $[4, 3, 1, 2]$ gives the best performance.

Consider 32-QAM, where $q = 5$ and $q' = 8$. Three sub-codewords of $x_{[1:N]}$ are punctured. The bit-mappers promising $\mathcal{P} = \mathcal{U}$ are:
$\mathcal{S} = \{\mathbf{s}|[s_1, s_2, s_3] = [6, 7, 8]\} \cup \{\mathbf{s}|[s_1, s_2, s_5] = [6, 7, 8]\} \cup \{\mathbf{s}|[s_1, s_3, s_5] = [6, 7, 8]\}$. After finding $\mathcal{S}^*$ and computing the corresponding CBER for the PIC-BIPCM, it is observed that bit-mapper $[6, 7, 8, 1, 2, 3, 5, 4]$ gives the best performance. ■

Bit-mapper Design for Multi-kernel PIC-BIPCM

For multi-kernel PIC-BIPCM, the kernel $\mathbf{F}_q$ needs to have a large polarization exponent [135] to promise a good error performance. The kernel matrices for 64-QAM ($q = 3$) and 32-QAM ($q = 5$) can be found in Table 9.3 and Table 9.4, where $\mathbf{F}_3$ is employed in [156,157], and $\mathbf{F}_5$ is given by [136]. The density evolution equations of $\mathbf{F}_3$ and $\mathbf{F}_5$ over BEC are presented in the same table. With the destiny evolution equations of $\mathbf{F}_q$, we can compute the block error probability of multi-kernel polar codes, and then compute the CBER of the corresponding PIC-BIPCM.

To find a bit-mapper that gives the best error performance for multi-kernel PIC-BIPCM, we first choose the coupling indices according to the design criteria proposed in Sec. 9.5.1 for each possible bit-mapper, and compute the CBER for the corresponding PIC-BIPCM. Then, we choose a bit-mapper which gives the lowest CBER. For 64-QAM, the designed bit-mapper is $[3, 2, 1]$. For 32-QAM, the designed bit-mapper is $[1, 2, 3, 5, 4]$.

9.6 Numerical Results

In this section, we investigate the CBER performance of PIC-BIPCM via Monte-Carlo simulation. We consider transmitting CBs continuously until enough CB decoding errors are observed so that the rate loss due to termination becomes negligible. Windowed decoding scheme is use for decoding PIC-BIPCM, where window sizes are $L_W = 10$ and $L_W = 20$ for $m = 1$ and $m = 4$, respectively. We use (N, R, λ, m) to represent the PIC-BIPCM parameters. We employ SC decoding unless specified otherwise. When CASCL decoding is employed, an 8-bit CRC is used and the decoding list size is 8.

To demonstrate the performance improvement of PIC-BIPCM over existing BIPCM. We use the BIPCM with the same codeword length and code rate as the benchmark. For the AWGN channel with 256-QAM, we consider the direct BIPCM as the benchmark. The bit-mapper of the direct BIPCM is obtained according to [158]. For the AWGN channel with 64-QAM and 32-QAM, we consider the punctured BIPCM and the multi-kernel BIPCM as the benchmarks. The bit-mapper of the punctured BIPCM is obtained according to [158]. The multi-kernel BIPCM for 64-QAM is from [156,157]. The multi-kernel BIPCM for 32-QAM is constructed with the kernel from [136].

9.6.1 Performance over the Parallel BECs

In this subsection, we consider the 4-BEC whose subchannel capacities are equivalent to the subchannel capacities of the AWGN channel with Gray-labeled 256-QAM. Codeword length $N = 1024$ is employed. We first compare the theoretical CBER derived in Sec. 9.4.3 with the simulated CBER to demonstrate the effectiveness of the CBER analysis. We then show how coupling indices design affects the CBER performance of PIC-BIPCM.

The theoretical and simulated CBER performance of PIC-BIPCM are shown in Fig. 9.13. In this figure, "sim." represents the simulated CBER, "approx., $m = 1$" represents the theoretical CBER of the PIC-BIPCM with $m = 1$ given

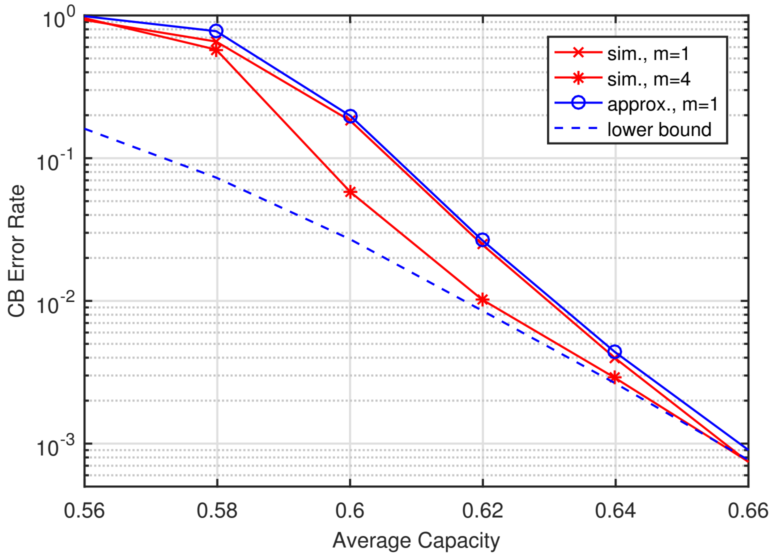


Figure 9.13: CBER of $(1024, \frac{1}{2}, 0.25, m)$ PIC-BIPCM with $m \in \{1, 4\}$ over 4-BEC.

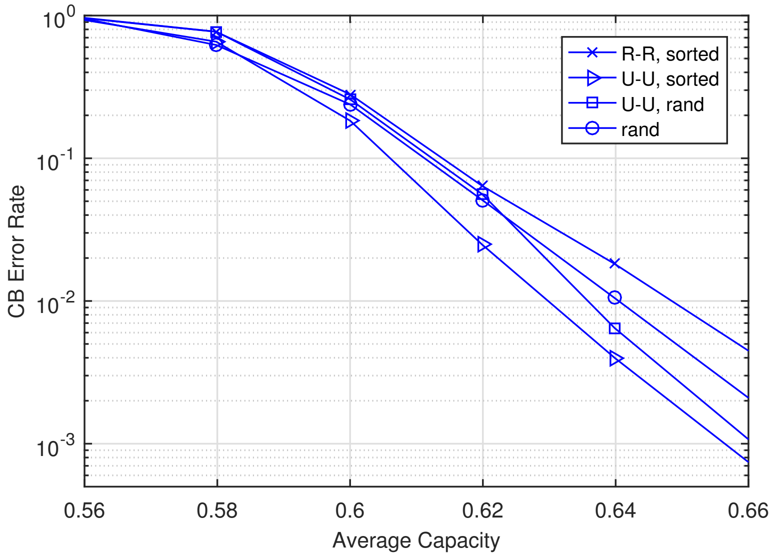


Figure 9.14: CBER of $(1024, \frac{1}{2}, 0.25, 1)$ PIC-BIPCM with different coupling indices designs over 4-BEC.

by the closed-form expression (9.12), which is also the CBER upper bound of the PIC-BIPCM with $m = 4$, and “lower bound” is the CBER lower bound of the PIC-BIPCM with $m = 1$ and $m = 4$. At a low average capacity, the PIC-BIPCM with $m = 4$ outperforms the PIC-BIPCM with $m = 1$. When average capacity increases, the performance gap between them reduces. When the average capacity is greater than 0.66, the performance gap becomes negligible, because both of them approach the CBER lower bound. In addition, it can also be observed that the theoretical CBER of the PIC-BIPCM with $m = 1$ is close to the stimulated CBER. Although the performance improvement of increasing coupling memory does not seem to be large as shown in Fig. 9.13, we will show later that when CASCL decoding is used, a large coupling memory could bring significant performance improvement.

Fig. 9.14 shows the CBER performance of the PIC-BIPCM with various coupling indices designs. In the legend, “R-R, sorted” represents that the reliable information bits are shared between CBs and coupling indices are sorted in ascending order, “U-U, sorted” represents that unreliable information bits are shared between CBs and coupling indices are sorted in ascending order. (i.e., the proposed coupling indices in Sec. 9.5), “U-U, rand” represents that unreliable information bits are shared between CBs and coupling indices are randomly interleaved, and “rand” represents that the shared information bits are randomly chosen and interleaved. It is observed that both “U-U, sorted” and “U-U, rand” outperform other coupling indices designs, suggesting that coupling unreliable information bits can benefit PIC-BIPCM the most. Meanwhile, “U-U, sorted” outperforms “U-U, rand”, which confirms the effectiveness of the design criteria presented in Sec. 9.5.1. For the rest of the PIC-BIPCM simulation results, the coupling indices are selected based on the proposed design criteria.

9.6.2 Performance over the AWGN Channel with 256-QAM

In this subsection, we consider the AWGN channel with Gray-labeled 256-QAM and show how different bit-mappers, coupling ratios, and coupling memories affect

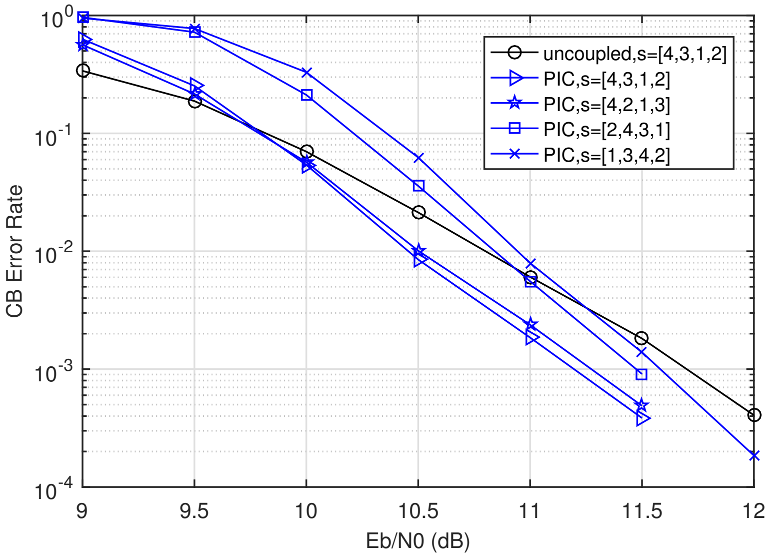


Figure 9.15: CBER of $(1024, \frac{1}{2}, 0.25, 1)$ PIC-BIPCM with different bit-mappers with 256-QAM.

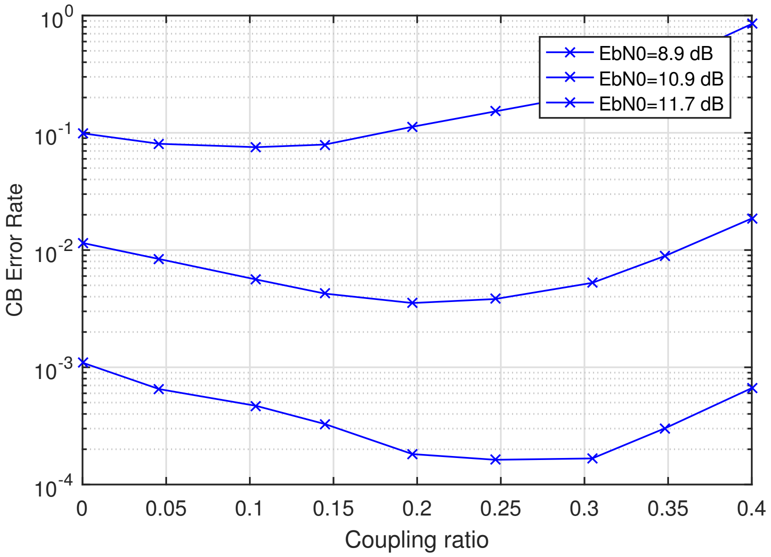


Figure 9.16: CBER of $(1024, \frac{1}{2}, \lambda, 1)$ PIC-BIPCM against λ with 256-QAM.

the CBER performance of direct PIC-BIPCM. Codeword length $N = 1024$ is employed.

We first show the CBER performance of the PIC-BIPCM with different bit-mappers. Recall that in Sec. 9.5.2, we use the CBER analysis over the BEC as a guideline to design the bit-mapper for PIC-BIPCM. Specifically, Fig. 9.12

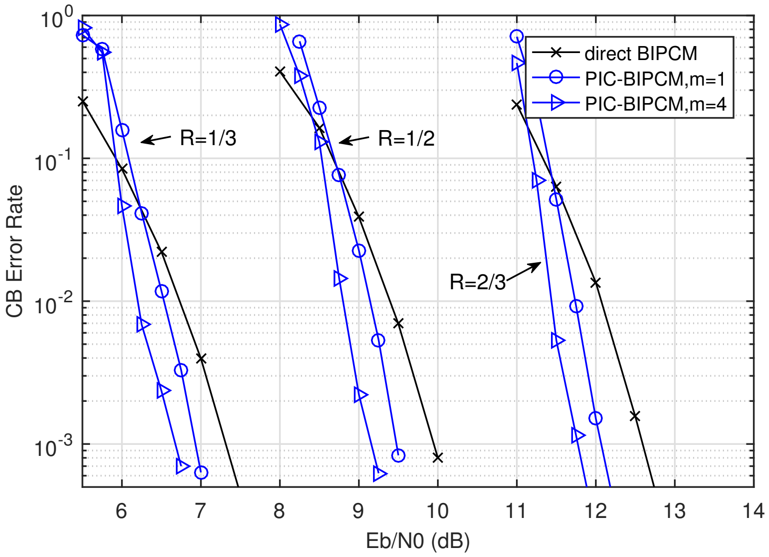


Figure 9.17: CBER of $(1024, R, 0.25, m)$ PIC-BIPCM with CASCL decoding with 256-QAM.

compares the theoretical CBER of the PIC-BIPCM with different bit-mappers, which suggests that bit-mapper $[4, 3, 1, 2]$ would have better CBER performance among all other bit-mappers. To show that this design method is effective for the PIC-BIPCM over the AWGN channel, we choose four bit-mappers shown in Fig. 9.12, and plot the simulated CBER of the PIC-BIPCM with these bit-mappers in Fig. 9.15, respectively. In both Fig. 9.12 and Fig. 9.15, the ranking of the bit-mappers based on the CBER at a high average capacity/SNR follows $[4, 3, 1, 2]$, $[4, 2, 1, 3]$, $[2, 4, 3, 1]$, and $[1, 3, 4, 2]$. This shows that the bit-mapper design based on the CBER analysis over the BEC performs well over the AWGN channel. For the rest of the simulation results for 256-QAM, bit-mapper $[4, 3, 1, 2]$ is used.

To investigate the impact of coupling ratio on the CBER performance, we show the CBER performance of PIC-BIPCM against various coupling ratio λ in Fig. 9.16. We consider SNR of 8.9 dB, 10.9 dB, and 11.7 dB, corresponding to the CBER of direct BIPCM (which is the same as the PIC-BIPCM with $\lambda = 0$) reaching 10^{-1}, 10^{-2}, and 10^{-3}, respectively. It is observed that the coupling ratio $\lambda \in [0.25, 0.3]$ brings a good CBER performance. In all other PIC-BIPCM simulations, $\lambda = 0.25$ is used. When $\lambda > 0.3$, the performance of PIC-BIPCM degrades with the increasing λ. On one hand, some reliable information bits are coupled when the coupling ratio is large, but coupling reliable information bits only brings minor coupling gain. On the other hand, we have to increase the code rate of the underlying component polar codes to compensate for the rate reduction due to coupling. Consequently, the performance of the PIC-BIPCM with a large coupling ratio, e.g., $\lambda = 0.4$ only has a minor gain over the direct BIPCM.

In Fig. 9.17, we show the CBER performance of the PIC-BIPCM with a CASCL decoder with coupling memory $m \in \{1, 4\}$ and code rate $R \in \{\frac{1}{3}, \frac{1}{2}, \frac{2}{3}\}$, respectively. At a CBER of 10^{-3}, the PIC-BIPCM with $m = 1$ has about 0.2 – 0.3 dB coding gain over the direct BIPCM, and the PIC-BIPCM with $m = 4$ has about 0.6 – 0.8 dB coding gain over the direct BIPCM. This means that increasing the coupling memory can effectively improve the CBER performance of PIC-BIPCM.

9.6.3 Performance over the AWGN Channel with 32-QAM and 64-QAM

In this subsection, we compare the performance of punctured PIC-BIPCM and multi-kernel PIC-BIPCM over AWGN channel with quasi-Gray-labeled 32-QAM and Gray-labeled 64-QAM, respectively. We consider $\widetilde{N} = 640$ for 32-QAM, and $\widetilde{N} = 768$ for 64-QAM, where the corresponding bit-mappers are listed in Table 9.2. Both SC decoding and CASCL decoding are considered.

The simulated CBER performances are shown in Fig. 9.18 and Fig. 9.19 respectively. When SC decoding is employed, both the punctured PIC-BIPCM

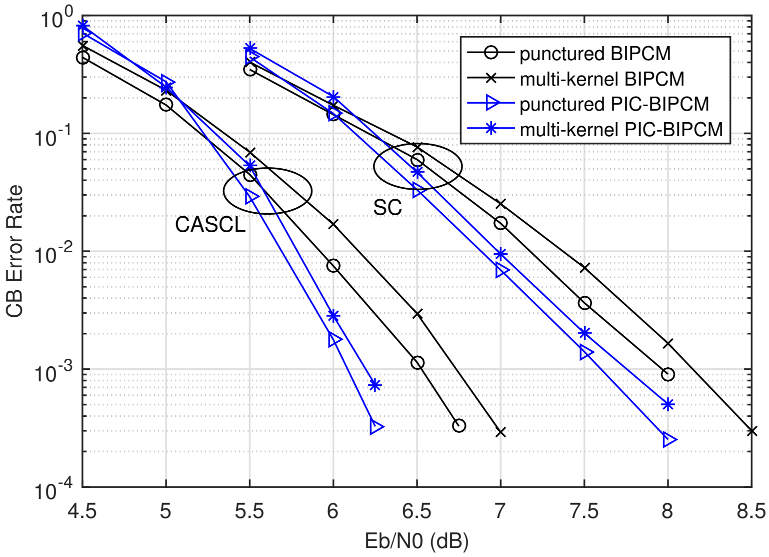


Figure 9.18: CBER of $(1024, \frac{1}{2}, 0.25, 1)$ punctured PIC-BIPCM and $(640, \frac{1}{2}, 0.25, 1)$ multi-kernel PIC-BIPCM with 32-QAM.

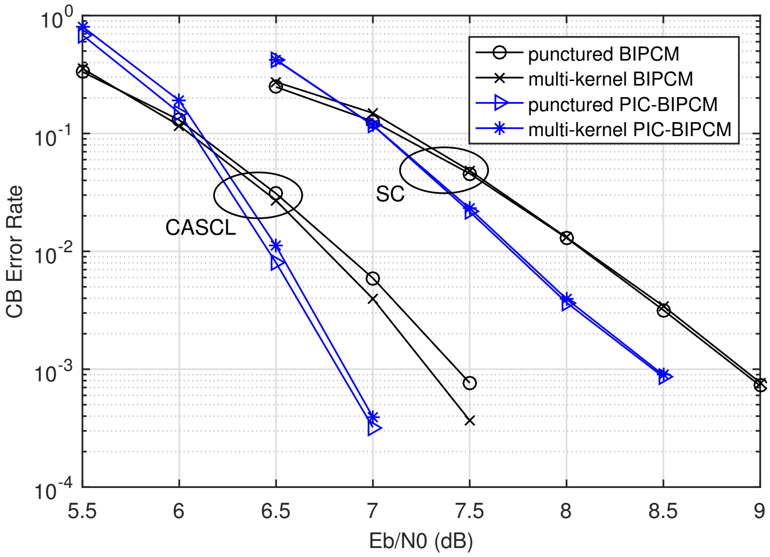


Figure 9.19: CBER of $(1024, \frac{1}{2}, 0.25, 1)$ punctured PIC-BIPCM and $(768, \frac{1}{2}, 0.25, 1)$ multi-kernel PIC-BIPCM with 64-QAM.

and the multi-kernel PIC-BIPCM can achieve about 0.3 – 0.35 dB coding gain over the punctured BIPCM and the multi-kernel BIPCM at CBER of 10^{-3}, respectively. When CASCL decoding is employed, the performance gain is about 0.4 – 0.6 dB. This demonstrates the effectiveness of our PIC-BIPCM design. It is also interesting to note that the punctured PIC-BIPCM slightly outperforms the multi-kernel PIC-BIPCM.

9.7 Conclusion

In this chapter, we proposed PIC-BIPCM and investigated its error performance over the AWGN channel with various high-order modulations, respectively. We analyzed the finite-length performance of PIC-BIPCM under SC decoding and derived a closed-form expression for the CBER of the PIC-BIPCM with $m = 1$. With the CBER analysis, we jointly designed the coupling indices and the bit-mapper to improve the CBER performance of PIC-BIPCM. Simulation results confirmed our analysis and demonstrated the improvement on the CBER performance of PIC-BIPCM over the three classes of existing BIPCM schemes.

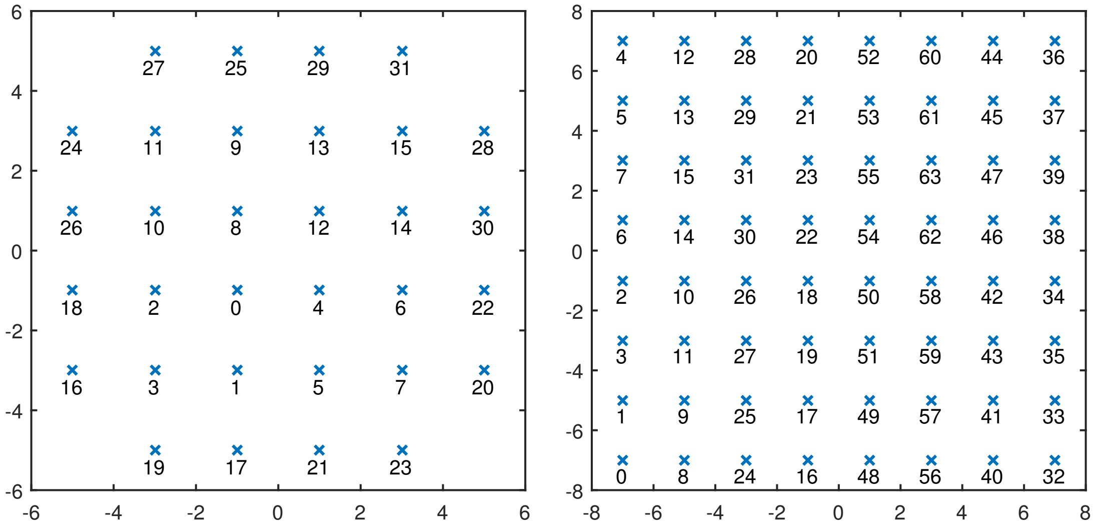
Appendix

9.A Computing CB Error Probability

Algorithm 9.1 $E_{d_{t-1},d_{t+1}} = F(W^{(1:q)}, d_{t-1}, d_{t+1}, \mathcal{A}, \mathcal{B}_\mathrm{H}, \mathcal{B}_\mathrm{T}, \mathbf{s})$

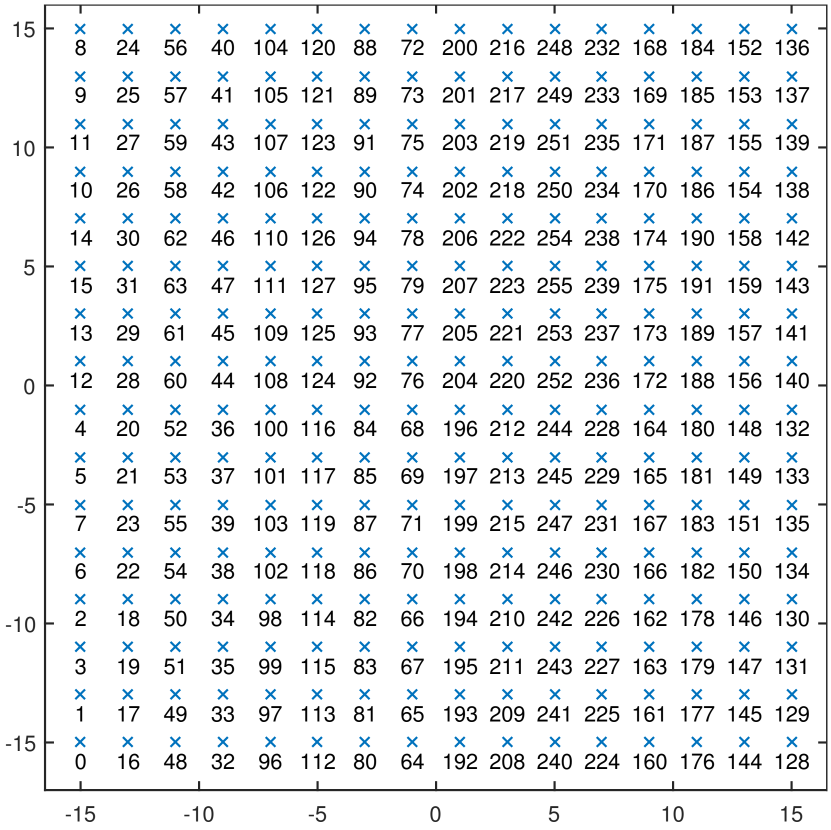
```
Input:
  $W^{(1:q)}$: a set of $q$ parallel BECs with erasure rate $\epsilon^{(1)}, \epsilon^{(2)}, \ldots, \epsilon^{(q)}$, respectively.
  $d_{t-1}, d_{t+1}$: extrinsic information availability.
  $\mathcal{A}$: information indices.
  $\mathcal{B}_\mathrm{H}, \mathcal{B}_\mathrm{T}$: coupling indices.
  $\mathbf{s} = [s_1, \ldots, s_q]$: bit-mapper.
Output:
  $E_{d_{t-1},d_{t+1}}$: the conditional CB error probability.
1) Initialize $\epsilon_1^{(1:N)}$ with channel erasure probability.
  for $i \in [1:N] \setminus \mathcal{B}_\mathrm{H}$ and $j \in [1:q]$ do
    if $i \in j$-th sub-CB then
      $\epsilon_1^{(i)} = \epsilon^{(s_i)}$
    end if
  end for
2) Update $\epsilon_1^{(1:N)}$ with extrinsic erasure probability.
  if $d_{t-1} = 1$ then
    $\epsilon_1^{(\mathcal{B}_\mathrm{H})} = 0$
  else if $d_{t-1} = 0$ then
    $\epsilon_1^{(\mathcal{B}_\mathrm{H})} = \epsilon_1^{(\mathcal{B}_\mathrm{T})}$
  end if
  if $d_{t+1} = 1$ then
    $\epsilon_1^{(\mathcal{B}_\mathrm{T})} = 0$
  end if
3) Apply density evolution equations (9.6) to $\epsilon_1^{(1:N)}$ and obtain $\epsilon_N^{(1:N)}$.
4) Compute the theoretical CB error probability: $E_{d_{t-1},d_{t+1}} \approx 1 - \prod_{i \in \mathcal{A}} (1 - \frac{\epsilon_N^{(i)}}{2})$.
```

9.B M-QAM Constellations



(a) quasi-Gray-labeled 32-QAM.

(b) Gray-labeled 64-QAM



(c) Gray-labeled 256-QAM.

Figure 9.20: Constellation for M-QAM.

9.C $\mathbf{F}_3$ and $\mathbf{F}_5$ Kernel Matrices

Table 9.3: 3 × 3 kernel matrices and density evolution equation.

Kernel matrix	Density evolution equation
$\mathbf{F}_3 = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}$	$\begin{cases} \epsilon_3^{(1)} = 1 - (1 - \epsilon_1^{(1)})(1 - \epsilon_1^{(2)})(1 - \epsilon_1^{(3)}) \\ \epsilon_3^{(2)} = \epsilon_1^{(1)}(1 - (1 - \epsilon_1^{(2)})(1 - \epsilon_1^{(3)})) \\ \epsilon_3^{(3)} = \epsilon_1^{(2)}\epsilon_1^{(3)} \end{cases}$

Table 9.4: 5 × 5 kernel matrices and density evolution equation.

Kernel matrix	Density evolution equation
$\mathbf{F}_5 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 \end{bmatrix}$	$\begin{cases} \epsilon_5^{(1)} = 1 - (1 - \epsilon_1^{(1)})(1 - \epsilon_1^{(2)})(1 - \epsilon_1^{(3)})(1 - \epsilon_1^{(4)})(1 - \epsilon_1^{(5)}) \\ \epsilon_5^{(2)} = \left(1 - (1 - \epsilon_1^{(1)})(1 - \epsilon_1^{(3)})(1 - \epsilon_1^{(4)})\right) \\ \qquad\qquad \cdot \left(1 - (1 - \epsilon_1^{(2)})(1 - \epsilon_1^{(5)})\right) \\ \epsilon_5^{(3)} = 1 - \left(1 - \epsilon_1^{(2)}\epsilon_1^{(5)}\right)\left(1 - \epsilon_1^{(3)}(1 - \epsilon^{(1)})(1 - \epsilon_1^{(4)})\right) \\ \epsilon_5^{(4)} = \epsilon_1^{(4)}\left(1 - (1 - \epsilon_1^{(1)})(1 - \epsilon_1^{(2)}\epsilon_1^{(3)}\epsilon_1^{(5)})\right) \\ \epsilon_5^{(5)} = \epsilon_1^{(1)}\epsilon_1^{(2)}\epsilon_1^{(3)}\epsilon_1^{(5)} \end{cases}$

Chapter 10

Thesis Conclusions

In this thesis, spatially coupled codes for point-to-point communication systems with TB-based HARQ have been studied. We proposed the partial coupling scheme to improve the TB error performance and consequently reduce the waste of transmit power and spectrum efficiency due to TB retransmission. Specifically, we constructed the partially coupled turbo-likes codes and partially coupled polar codes. Both theoretical performance analysis and simulation results demonstrated that our partial coupling scheme can significantly improve the decoding performance over the underlying constituent code. We conclude the thesis in the following by summarizing our main contributions. Several related future research topics are presented as well.

10.1 Contributions

In Chapter 1, we presented an overview of the mobile communication networks followed by our research motivations of this thesis. Previous works related to coding techniques for point-to-point communication were reviewed and discussed. The thesis organization and the main contributions of each work conducted in this thesis were provided.

In Chapter 2, we provided the fundamental background on channel coding that is necessary for understanding the works in the later chapters. This includes the system model for digital point-to-point communication, various channel models, important definitions of channel coding, and state-of-the-art coding techniques.

In Chapter 3, we summarized the fundamental background on turbo codes from the literature to provide the necessary background for understanding the work throughout Chapters 4 – 6. To be specific, we reviewed the encoding and decoding of convolutional codes and turbo codes, performance analysis tools, and some existing spatially coupled turbo-like codes.

In Chapter 4, we introduced the partial coupling technique, where a fraction of the information/parity sequence of a code block (CB) is used as a part of the information sequence of the succeeding CBs. We constructed two types of partially coupled turbo codes: partially information coupled turbo codes (PIC-TCs) and partially parity coupled turbo codes (PPC-TCs). The decoding schemes of PIC-TCs and PPC-TCs were presented, respectively. The iterative decoding thresholds of the proposed codes over binary erasure channels (BECs) were obtained via density evolution analysis. Simulation results demonstrated the significant error performance improvement of the partially coupled turbo codes over turbo codes and some existing spatially coupled turbo codes.

In Chapter 5, the partially couple duo-binary turbo codes (dTCs) were presented. The proposed codes were constructed by using one input of the dTC to carry the information sequence of the current CB and using another input to carry the coupled information/parity subsequences from the previous CBs. We constructed four different schemes within the class of partially coupled dTCs by investigating the impact of partial coupling on the turbo code level and convolutional code level. With density evolution analysis and simulation, we showed that each construction within this class has its unique characteristics that can be exploited to satisfy different requirements.

In Chapter 6, we introduced the generalized SC-PCCs (GSC-PCCs) by combining and extending the idea of PIC-TCs and SC-PCCs [38]. Specifically, GSC-PCCs were constructed by applying spatial coupling on PCCs with a fraction λ of information bits repeated q times. SC-PCCs can be seen as the GSC-PCCs with $q = 1$ (or $\lambda = 0$). PIC-TCs can be seen as special cases of GSC-PCCs with $q = 2$, where spatial coupling only applies to the repeated bits. The density evolution analysis showed that GSC-PCCs can surpass SC-PCCs and PIC-TCs of the same rates and can closely approach the BEC capacity.

In Chapter 7, we reviewed the fundamental background on polar codes summarized from the literature to provide the necessary background for understanding the work throughout Chapters 8 – 9. This includes the channel polarization phenomenon, the encoding of polar codes, various decoding algorithms, and performance analysis tools.

In Chapter 8, PIC-polar codes were constructed by applying PIC to systematic polar codes. The coupling indices were designed to utilize the polarization property of polar codes. We also proposed a decoding scheme that can mitigate the error propagation problem between polar CBs. A closed-form expression for TBER of the PIC-polar codes was derived. The numerical results showed that PIC-polar codes can outperform the uncoupled polar codes with a slightly increased decoding complexity.

In Chapter 9, we investigated the design of PIC-polar coding schemes for 2^Q-ary modulations. To be specific, we considered PIC-BIPCM schemes constructed with three BIPCM schemes: direct BIPCM, punctured BIPCM, and multi-kernel

BIPCM, respectively. The joint design of the coupling indices and the modulation bit-mapper were discussed. A closed-form expression for the CBER of the PIC-BIPCM with $m = 1$ was derived. Both analysis and simulation results showed that spatial coupling gain and coded modulation can be jointly attained by our PIC-BIPCM.

10.2 Future Research Directions

In this section, we present some future research topics related to our works and present some potential research topics.

Partially Coupled LDPC codes

In [140], authors construct PIC-LDPC codes by using the WiMAX standard LDPC codes [23] as component code. In [160], authors couple the LDPC CBs using PIC and superimpose parity bits of one CB onto the following CBs using the partial superposition (PS) technique, resulting in a PIC-PS-LDPC code. The simulation results demonstrate that the PIC-PS technique is universal in the sense that various LDPC codes, including NR LDPC code [13], can be used to construct the PIC-PS-LDPC codes.

A potential research direction is to design irregular finite-length LDPC code for partial coupling schemes by optimizing the node degrees. In particular, it needs to take into account the coupled information/parity bits as partial coupling increases the degree of the coupled variable nodes.

Zipper Codes

Zipper code [161] is a framework for describing spatially coupled codes such as staircase codes [162] and braided block codes [163]. By introducing this framework to describe our partially coupled codes, we may further discover efficient low-complexity decoding schemes for our codes.

Performance Analysis Over the AWGN channel

So far, the analysis for finite-length scaling [100, 101] of spatially coupled turbo-like codes is still an open problem. At the current stage, we observe that the finite-length scaling may be affected by coupling structure and coupling ratio. The coupling ratio that maximizes iterative decoding threshold may not promise optimal performance for short information length (e.g., around 1000). A promising research direction could be looking into the structural properties of our codes and analyzing their finite-length performance.

Design of Coupling Indices for Partially Coupled Turbo-like Codes

At the current stage, the design of our partially coupled turbo-like codes mainly focuses on improving the performance in the waterfall regime by optimizing the iterative decoding threshold. To further improve the performance in the error floor regime, we need to consider optimizing the distance properties of the coupled codes. In particular, we need to improve the design of coupling indices while taking into account how turbo interleaver and puncturing affect the distance properties of the component turbo-like codes.

Bibliography

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

[2] M. Mansour and N. Shanbhag, "High-throughput LDPC decoders," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 11, no. 6, pp. 976–996, 2003.

[3] Z. Wang and Z. Cui, "Low-complexity high-speed decoder design for quasi-cyclic LDPC codes," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 15, no. 1, pp. 104–114, 2007.

[4] X. Wang, J. Tong, H. Zhang, S. Dai, R. Li, and J. Wang, "Toward terabits-per-second communications: Low-complexity parallel decoding of GN-coset codes," in *2021 IEEE Wireless Communications and Networking Conference (WCNC)*, 2021, pp. 1–5.

[5] A. Sural, E. G. Sezer, Y. Ertugrul, O. Arikan, and E. Arikan, "Terabits-per-second throughput for polar codes," in *2019 IEEE 30th International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC Workshops)*, 2019, pp. 1–7.

[6] H. Wymeersch, H. Steendam, and M. Moeneclaey, "Log-domain decoding of LDPC codes over GF(q)," in *2004 IEEE International Conference on Communications (IEEE Cat. No.04CH37577)*, vol. 2, 2004, pp. 772–776 Vol.2.

[7] A. Voicila, D. Declercq, F. Verdier, M. Fossorier, and P. Urard, "Low-complexity decoding for non-binary LDPC codes in high order fields," *IEEE Trans. Commun.*, vol. 58, no. 5, pp. 1365–1375, 2010.

[8] G. Sarkis, P. Giard, A. Vardy, C. Thibeault, and W. J. Gross, "Fast polar decoders: Algorithm and implementation," *IEEE J. Sel. Areas Commun.*, vol. 32, no. 5, pp. 946–957, 2014.

[9] 3GPP TS 45.003, "Technical specification group GERAN; channel coding," Release 5, 2001.

[10] 3GPP TS 25.212, “Universal mobile telecommunications system (UMTS); multiplexing and channel coding (FDD),” Release 4, 2001.

[11] 3GPP TS 25.222, “Universal mobile telecommunications system (UMTS); multiplexing and channel coding (TDD),” Release 4, 2001.

[12] 3GPP TS 36.212, “LTE; evolved universal terrestrial radio access (E-UTRA); multiplexing and channel coding,” Release 13, 2016.

[13] 3GPP TS 38.212, “5G; NR; multiplexing and channel coding,” Release 16, 2020.

[14] M. Shafi, A. F. Molisch, P. J. Smith, T. Haustein, P. Zhu, P. De Silva, F. Tufvesson, A. Benjebbour, and G. Wunder, “5G: A tutorial overview of standards, trials, challenges, deployment, and practice,” *IEEE J. Sel. Areas Commun.*, vol. 35, no. 6, pp. 1201–1221, 2017.

[15] W. Saad, M. Bennis, and M. Chen, “A vision of 6G wireless systems: Applications, trends, technologies, and open research problems,” *IEEE Network*, vol. 34, no. 3, pp. 134–142, 2020.

[16] T. S. Rappaport, Y. Xing, O. Kanhere, S. Ju, A. Madanayake, S. Mandal, A. Alkhateeb, and G. C. Trichopoulos, “Wireless communications and applications above 100 GHz: Opportunities and challenges for 6G and beyond,” *IEEE Access*, vol. 7, pp. 78 729–78 757, 2019.

[17] Z. Zhang, Y. Xiao, Z. Ma, M. Xiao, Z. Ding, X. Lei, G. K. Karagiannidis, and P. Fan, “6G wireless networks: Vision, requirements, architecture, and key technologies,” *IEEE Vehicular Technology Magazine*, vol. 14, no. 3, pp. 28–41, 2019.

[18] Y. Polyanskiy, H. V. Poor, and S. Verdu, “Channel coding rate in the finite blocklength regime,” *IEEE Trans. Inf. Theory*, vol. 56, no. 5, pp. 2307–2359, 2010.

[19] R. Gallager, “Low-density parity-check codes,” *IRE Trans. Inf. Theory*, vol. 8, no. 1, pp. 21–28, January 1962.

[20] D. MacKay, “Good error-correcting codes based on very sparse matrices,” *IEEE Trans. Inf. Theory*, vol. 45, no. 2, pp. 399–431, 1999.

[21] R. McEliece, D. MacKay, and J.-F. Cheng, “Turbo decoding as an instance of Pearl’s "belief propagation" algorithm,” *IEEE J. Sel. Areas Commun.*, vol. 16, no. 2, pp. 140–152, 1998.

[22] ETSI EN 302 307, “Digital Video Broadcasting (DVB); second generation framing structure, channel coding and modulation systems for broadcasting,

interactive services, news gathering and other broadband satellite applications (DVB-S2)," V1.4.1, 2014.

[23] IEEE Std 802.16e™-2005, "Part 16: Air interface for fixed and mobile broadband wireless access systems," Feb. 2006.

[24] IEEE Std 802.11n™-2012, "Part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications," Mar. 2012.

[25] 3GPP TSG RAN WG1 Meeting 85, Ericsson, "Performance evaluation of turbo codes and LDPC codes at higher code rates," May. 2016.

[26] T. Richardson and S. Kudekar, "Design of low-density parity check codes for 5G new radio," *IEEE Communications Magazine*, vol. 56, no. 3, pp. 28–34, 2018.

[27] E. Arikan, "Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels," *IEEE Trans. Inf. Theory*, vol. 55, no. 7, pp. 3051–3073, 2009.

[28] I. Tal and A. Vardy, "List decoding of polar codes," *IEEE Trans. Inf. Theory*, vol. 61, no. 5, pp. 2213–2226, 2015.

[29] D. Hui, S. Sandberg, Y. Blankenship, M. Andersson, and L. Grosjean, "Channel coding in 5G new radio: A tutorial overview and performance comparison with 4G LTE," *IEEE Vehicular Technology Magazine*, vol. 13, no. 4, pp. 60–69, 2018.

[30] M. Sybis, K. Wesolowski, K. Jayasinghe, V. Venkatasubramanian, and V. Vukadinovic, "Channel coding for ultra-reliable low-latency communication in 5G systems," in *2016 IEEE 84th Vehicular Technology Conference (VTC-Fall)*, 2016, pp. 1–5.

[31] H.-T. Pai, Y. S. Han, and Y.-J. Chu, "New HARQ scheme based on decoding of tail-biting convolutional codes in ieee 802.16e," *IEEE Trans. Veh. Technol.*, vol. 60, no. 3, pp. 912–918, 2011.

[32] 3GPP TSG RAN WG1 Meeting 86, LG Electronics, "Discussion on outer coding on eMBB data," Aug. 2016.

[33] 3GPP TSG RAN WG1 Meeting 86, Qualcomm, "Erasure coding evaluation methodology," Aug. 2016.

[34] 3GPP TR 36.881, "Technical specification group radio access network; evolved universal terrestrial radio access (e-utra); study on latency reduction techniques for LTE," Release 14, 2016.

[35] A. Jimenez Felstrom and K. S. Zigangirov, “Time-varying periodic convolutional codes with low-density parity-check matrix,” vol. 45, no. 6, Sep. 1999, pp. 2181–2191.

[36] 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, 2013.

[37] 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, July 2018.

[38] S. Moloudi, M. Lentmaier, and A. Graell i Amat, “Spatially coupled turbo-like codes,” *IEEE Trans. Inf. Theory*, vol. 63, no. 10, pp. 6199–6215, Oct 2017.

[39] 3GPP TS 36.213, “LTE; evolved universal terrestrial radio access (E-UTRA); physical layer procedures,” Release 13, 2016.

[40] 3GPP TS 38.213, “5G; NR; physical layer procedures for control,” Release 16, 2020.

[41] R. W. Hamming, “Error detecting and error correcting codes,” *The Bell System Technical Journal*, vol. 29, no. 2, pp. 147–160, 1950.

[42] A. HOCQUENGHEM, “Codes correcteurs d’erreurs,” *Chiffers*, vol. 2, pp. 147–156, 1959. [Online]. Available: https://ci.nii.ac.jp/naid/10016152997/en/

[43] R. Bose and D. Ray-Chaudhuri, “On a class of error correcting binary group codes,” *Information and Control*, vol. 3, no. 1, pp. 68–79, 1960. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0019995860902874

[44] W. W. Peterson and D. T. Brown, “Cyclic codes for error detection,” *Proceedings of the IRE*, vol. 49, no. 1, pp. 228–235, 1961.

[45] P.Elias, “Coding for noisy channels,” *IRE Conv. Rec.*, pp. 4:37–47, 1955.

[46] A. Viterbi, “Error bounds for convolutional codes and an asymptotically optimum decoding algorithm,” *IEEE Trans. Inf. Theory*, vol. 13, no. 2, pp. 260–269, 1967.

[47] L. Bahl, J. Cocke, F. Jelinek, and J. Raviv, “Optimal decoding of linear codes for minimizing symbol error rate (corresp.),” *IEEE Trans. Inf. Theory*, vol. 20, no. 2, pp. 284–287, March 1974.

[48] C. Berrou, A. Glavieux, and P. Thitimajshima, “Near shannon limit error-correcting coding and decoding: Turbo-codes. 1,” in *Proceedings of ICC ’93 - IEEE International Conference on Communications*, vol. 2, 1993, pp. 1064–1070 vol.2.

[49] G. Ungerboeck, “Channel coding with multilevel/phase signals,” *IEEE Trans. Inf. Theory*, vol. 28, no. 1, pp. 55–67, 1982.

[50] ——, “Trellis-coded modulation with redundant signal sets part i: Introduction,” *IEEE Communications Magazine*, vol. 25, no. 2, pp. 5–11, 1987.

[51] U. Wachsmann, R. F. H. Fischer, and J. B. Huber, “Multilevel codes: theoretical concepts and practical design rules,” *IEEE Trans. Inf. Theory*, vol. 45, no. 5, pp. 1361–1391, 1999.

[52] G. Caire, G. Taricco, and E. Biglieri, “Bit-interleaved coded modulation,” *IEEE Trans. Inf. Theory*, vol. 44, no. 3, pp. 927–946, 1998.

[53] H. Ma, W. K. Leung, X. Yan, K. Law, and M. Fossorier, “Delayed bit interleaved coded modulation,” in *2016 9th International Symposium on Turbo Codes and Iterative Information Processing (ISTC)*, 2016, pp. 86–90.

[54] R. Tanner, “A recursive approach to low complexity codes,” *IEEE Trans. Inf. Theory*, vol. 27, no. 5, pp. 533–547, 1981.

[55] F. Kschischang, B. Frey, and H.-A. Loeliger, “Factor graphs and the sum-product algorithm,” *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 498–519, 2001.

[56] J. Thorpe, “Low density parity check (LDPC) codes constructed from protographs,” *Jet Propulsion Laboratory INP Progress Report*, pp. 42–154, Aug. 2003.

[57] J. Pearl, “Fusion, propagation, and structuring in belief networks,” *Articial Intelligence*, vol. 29, no. 3, pp. 241–288, 1986.

[58] R. J. McEliece and L. Swanson, “Reed–solomon codes and the exploration of the solar system,” *Jet Propulsion Laboratory*, 1993.

[59] K. S. Andrews, D. Divsalar, S. Dolinar, J. Hamkins, C. R. Jones, and F. Pollara, “The development of turbo and LDPC codes for deep-space applications,” *Proceedings of the IEEE*, vol. 95, no. 11, pp. 2142–2156, 2007.

[60] 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.

[61] 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.

[62] Y. Xie, L. Yang, P. Kang, and J. Yuan, “Euclidean geometry-based spatially coupled LDPC codes for storage,” *IEEE J. Sel. Areas Commun.*, vol. 34, no. 9, pp. 2498–2509, Sep. 2016.

[63] A. Huebner, K. S. Zigangirov, and D. J. Costello, “Laminated turbo codes: A new class of block-convolutional codes,” *IEEE Trans. Inf. Theory*, vol. 54, no. 7, pp. 3024–3034, July 2008.

[64] W. Zhang, M. Lentmaier, K. S. Zigangirov, and D. J. Costello, “Braided convolutional codes: A new class of turbo-like codes,” *IEEE Trans. Inf. Theory*, vol. 56, no. 1, pp. 316–331, Jan 2010.

[65] S. Moloudi, M. Lentmaier, and A. Graell i Amat, “Spatially coupled hybrid concatenated codes,” in *SCC 2017; 11th International ITG Conference on Systems, Communications and Coding*, Feb 2017, pp. 1–6.

[66] A. Elkelesh, M. Ebada, S. Cammerer, and S. ten Brink, “Flexible length polar codes through graph based augmentation,” in *SCC 2017; 11th International ITG Conference on Systems, Communications and Coding*, 2017, pp. 1–6.

[67] K. Wang, W. Hou, S. Lu, P. Wu, Y. Ueng, and J. Cheng, “Improving polar codes by spatial coupling,” in *2018 International Symposium on Information Theory and Its Applications (ISITA)*, 2018, pp. 432–436.

[68] B. Feng, J. Jiao, L. Zhou, S. Wu, B. Cao, and Q. Zhang, “A novel high-rate polar-staircase coding scheme,” in *2018 IEEE 88th Vehicular Technology Conference (VTC-Fall)*, 2018, pp. 1–5.

[69] C. Condo, V. Bioglio, and I. Land, “Staircase construction with non-systematic polar codes,” in *2020 Optical Fiber Communications Conference and Exhibition (OFC)*, 2020, pp. 1–3.

[70] CCSDS 131.0-B-3, “CCSDS recommendation for space data system standards; TM synchronization and channel coding,” September 2017.

[71] J. Hagenauer and P. Hoeher, “A viterbi algorithm with soft-decision outputs and its applications,” in *1989 IEEE Global Telecommunications Conference and Exhibition 'Communications Technology for the 1990s and Beyond'*, 1989, pp. 1680–1686 vol.3.

[72] P. Robertson, E. Villebrun, and P. Hoeher, “A comparison of optimal and sub-optimal map decoding algorithms operating in the log domain,” in

Proceedings IEEE International Conference on Communications ICC '95, vol. 2, 1995, pp. 1009–1013 vol.2.

[73] M. Valenti and J. Sun, "The umts turbo code and an efficient decoder implementation suitable for software-defined radios," *International Journal of Wireless Information Networks*, vol. 8, pp. 203–215, 10 2001.

[74] C. Berrou and M. Jezequel, "Non-binary convolutional codes for turbo coding," *Electronics Letters*, vol. 35, no. 1, pp. 39–40, Jan 1999.

[75] C. Berrou, M. Jezequel, C. Douillard, and S. Kerouedan, "The advantages of non-binary turbo codes," in *Proceedings 2001 IEEE Information Theory Workshop (Cat. No.01EX494)*, 2001, pp. 61–63.

[76] C. Douillard and C. Berrou, "Turbo codes with rate-m/(m+1) constituent convolutional codes," *IEEE Trans. Commun.*, vol. 53, no. 10, pp. 1630–1638, Oct 2005.

[77] S. Benedetto, D. Divsalar, G. Montorsi, and F. Pollara, "Serial concatenation of interleaved codes: performance analysis, design, and iterative decoding," *IEEE Trans. Inf. Theory*, vol. 44, no. 3, pp. 909–926, 1998.

[78] C. Koller, A. Graell i Amat, J. Kliewer, F. Vatta, and D. J. Costello, "Hybrid concatenated codes with asymptotically good distance growth," in *2008 5th International Symposium on Turbo Codes and Related Topics*, 2008, pp. 19–24.

[79] M. U. Farooq, S. Moloudi, and M. Lentmaier, "Thresholds of braided convolutional codes on the AWGN channel," in *2018 IEEE International Symposium on Information Theory (ISIT)*, 2018, pp. 1375–1379.

[80] H. El Gamal and A. Hammons, "Analyzing the turbo decoder using the gaussian approximation," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 671–686, 2001.

[81] S. Moloudi, M. Lentmaier, and A. Graell i Amat, "Spatially coupled turbo-like codes: A new trade-off between waterfall and error floor," *IEEE Trans. Commun.*, vol. 67, no. 5, pp. 3114–3123, 2019.

[82] B. M. Kurkoski, P. H. Siegel, and J. K. Wolf, "Exact probability of erasure and a decoding algorithm for convolutional codes on the binary erasure channel," in *IEEE Global Telecommunications Conference*, vol. 3, Dec 2003, pp. 1741–1745 vol.3.

[83] L. Perez, J. Seghers, and D. Costello, "A distance spectrum interpretation of turbo codes," *IEEE Trans. Inf. Theory*, vol. 42, no. 6, pp. 1698–1709, 1996.

[84] R. Garello, F. Chiaraluce, P. Pierleoni, M. Scaloni, and S. Benedetto, “On error floor and free distance of turbo codes,” in *ICC 2001. IEEE International Conference on Communications. Conference Record (Cat. No.01CH37240)*, vol. 1, 2001, pp. 45–49 vol.1.

[85] S. Benedetto and G. Montorsi, “Design of parallel concatenated convolutional codes,” *IEEE Trans. Commun.*, vol. 44, no. 5, pp. 591–600, 1996.

[86] T. Duman and M. Salehi, “New performance bounds for turbo codes,” *IEEE Transactions on Communications*, vol. 46, no. 6, pp. 717–723, 1998.

[87] S. Benedetto and G. Montorsi, “Unveiling turbo codes: some results on parallel concatenated coding schemes,” *IEEE Trans. Inf. Theory*, vol. 42, no. 2, pp. 409–428, 1996.

[88] B. Vucetic and J. Yuan, *Turbo Codes Principles and Applications.* Norwell, MA: Kluwer, 2000.

[89] D. Divsalar and R. J. McEliece, “Effective free distance of turbo codes,” *Electronics Letters*, vol. 32, no. 5, p. 445, 1996.

[90] P. Elias, “Error-free coding,” *IRE Trans. Inf. Theory*, vol. 4, no. 4, pp. 29–37, 1954.

[91] M. Zhu, D. G. M. Mitchell, M. Lentmaier, D. J. Costello, and B. Bai, “Window decoding of braided convolutional codes,” in *2015 IEEE Information Theory Workshop - Fall (ITW)*, 2015, pp. 143–147.

[92] M. Mahdavi, M. Umar Farooq, L. Liu, O. Edfors, V. Öwall, and M. Lentmaier, “The effect of coupling memory and block length on spatially coupled serially concatenated codes,” in *2021 IEEE 93rd Vehicular Technology Conference (VTC2021-Spring)*, 2021, pp. 1–7.

[93] L. Yang, Y. Xie, X. Wu, J. Yuan, X. Cheng, and L. Wan, “Partially information-coupled turbo codes for LTE systems,” *IEEE Trans. Commun.*, vol. 66, no. 10, pp. 4381–4392, 2018.

[94] M. Qiu, X. Wu, A. G. i. Amat, and J. Yuan, “Analysis and design of partially information- and partially parity-coupled turbo codes,” *IEEE Trans. Commun.*, vol. 69, no. 4, pp. 2107–2122, 2021.

[95] C. Measson, R. Urbanke, A. Montanari, and T. Richardson, “Maximum a posteriori decoding and turbo codes for general memoryless channels,” in *Proceedings. International Symposium on Information Theory, 2005. ISIT 2005.*, 2005, pp. 1241–1245.

[96] X. Wu, M. Qiu, and J. Yuan, "Partially information coupled duo-binary turbo codes," in *2020 IEEE International Symposium on Information Theory (ISIT)*, 2020, pp. 461–466.

[97] M. Qiu, X. Wu, Y. Xie, and J. Yuan, "Density evolution analysis of partially information coupled turbo codes on the erasure channel," in *2019 IEEE Information Theory Workshop (ITW)*, 2019, pp. 1–5.

[98] S. Moloudi, M. Lentmaier, and A. G. Amat, "Finite length weight enumerator analysis of braided convolutional codes," in *2016 International Symposium on Information Theory and Its Applications (ISITA)*, 2016, pp. 488–492.

[99] A. Amraoui, A. Montanari, T. Richardson, and R. Urbanke, "Finite-length scaling for iteratively decoded LDPC ensembles," *IEEE Trans. Inf. Theory*, vol. 55, no. 2, pp. 473–498, 2009.

[100] I. Andriyanova, "Finite-length scaling of turbo-like code ensembles on the binary erasure channel," *IEEE J. Sel. Areas Commun.*, vol. 27, no. 6, pp. 918–927, 2009.

[101] J. W. Lee, R. Urbanke, and R. E. Blahut, "On the performance of turbo codes over the binary erasure channel," *IEEE Commun. Lett.*, vol. 11, no. 1, pp. 67–69, 2007.

[102] F. Vatta, A. Graell i Amat, A. Banerjee, and D. J. Costello, "Nonsystematic turbo codes: Design and bounds on effective free distance," in *2008 International Symposium on Information Theory and Its Applications*, 2008, pp. 1–6.

[103] D. Costello, H. Cabral, and O. Takeshita, *Some Thoughts on the Equivalence of Systematic and Nonsystematic Convolutional Encoders.* Kluwer Academic Publishers, 2002, pp. 57–75.

[104] M. Qiu, X. Wu, J. Yuan, and A. G. i. Amat, "Generalized spatially coupled parallel concatenated convolutional codes with partial repetition," in *2021 IEEE International Symposium on Information Theory (ISIT)*, 2021, pp. 581–586.

[105] E. Arikan, "A performance comparison of polar codes and Reed-Muller codes," *IEEE Commun. Lett.*, vol. 12, no. 6, pp. 447–449, 2008.

[106] O. Afisiadis, A. Balatsoukas-Stimming, and A. Burg, "A low-complexity improved successive cancellation decoder for polar codes," in *2014 48th Asilomar Conference on Signals, Systems and Computers*, 2014, pp. 2116–2120.

[107] F. Ercan, C. Condo, and W. J. Gross, “Improved bit-flipping algorithm for successive cancellation decoding of polar codes,” *IEEE Trans. Commun.*, vol. 67, no. 1, pp. 61–72, 2019.

[108] K. Niu and K. Chen, “CRC-aided decoding of polar codes,” *IEEE Commun. Lett.*, vol. 16, no. 10, pp. 1668–1671, 2012.

[109] E. Arikan, “From sequential decoding to channel polarization and back again,” 2019.

[110] ——, “Polar codes: A pipelined implementation,” *Proc. Int. Symp. Broad. Commun. (ISBC)*, pp. 11–14, 2010.

[111] U. U. Fayyaz and J. R. Barry, “Low-complexity soft-output decoding of polar codes,” *IEEE J. Sel. Areas Commun.*, vol. 32, no. 5, pp. 958–966, 2014.

[112] A. Elkelesh, M. Ebada, S. Cammerer, and S. ten Brink, “Belief propagation list decoding of polar codes,” *IEEE Commun. Lett.*, vol. 22, no. 8, pp. 1536–1539, 2018.

[113] M. Geiselhart, A. Elkelesh, M. Ebada, S. Cammerer, and S. ten Brink, “CRC-aided belief propagation list decoding of polar codes,” in *2020 IEEE International Symposium on Information Theory (ISIT)*, 2020, pp. 395–400.

[114] R. Mori and T. Tanaka, “Performance of polar codes with the construction using density evolution,” *IEEE Commun. Lett.*, vol. 13, no. 7, pp. 519–521, 2009.

[115] I. Tal and A. Vardy, “How to construct polar codes,” *IEEE Trans. Inf. Theory*, vol. 59, no. 10, pp. 6562–6582, 2013.

[116] P. Trifonov, “Efficient design and decoding of polar codes,” *IEEE Trans. Commun.*, vol. 60, no. 11, pp. 3221–3227, Nov. 2012.

[117] D. Wu, Y. Li, and Y. Sun, “Construction and block error rate analysis of polar codes over AWGN channel based on Gaussian approximation,” *IEEE Commun. Lett.*, vol. 18, no. 7, pp. 1099–1102, July 2014.

[118] H. Li and J. Yuan, “A practical construction method for polar codes in AWGN channels,” in *IEEE 2013 Tencon - Spring*, Apr. 2013, pp. 223–226.

[119] S.-Y. Chung, T. J. Richardson, and R. L. Urbanke, “Analysis of sum-product decoding of low-density parity-check codes using a Gaussian approximation,” *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 657–670, Feb. 2001.

[120] C. Schürch, "A partial order for the synthesized channels of a polar code," in *2016 IEEE International Symposium on Information Theory (ISIT)*, 2016, pp. 220–224.

[121] G. He, J.-C. Belfiore, I. Land, G. Yang, X. Liu, Y. Chen, R. Li, J. Wang, Y. Ge, R. Zhang, and W. Tong, "Beta-expansion: A theoretical framework for fast and recursive construction of polar codes," in *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*, 2017, pp. 1–6.

[122] W. Wu and P. H. Siegel, "Generalized partial orders for polar code bit-channels," *IEEE Trans. Inf. Theory*, vol. 65, no. 11, pp. 7114–7130, 2019.

[123] T. Tonnellier and W. J. Gross, "On systematic polarization-adjusted convolutional (PAC) codes," *IEEE Commun. Lett.*, vol. 25, no. 7, pp. 2128–2132, 2021.

[124] B. Li, H. Shen, and D. Tse, "A RM-polar codes," 2014.

[125] M. Mondelli, S. H. Hassani, and R. L. Urbanke, "From polar to Reed-Muller codes: A technique to improve the finite-length performance," *IEEE Trans. Commun.*, vol. 62, no. 9, pp. 3084–3091, 2014.

[126] E. Abbe and M. Ye, "Reed-Muller codes polarize," *IEEE Trans. Inf. Theory*, vol. 66, no. 12, pp. 7311–7332, 2020.

[127] E. Arikan, "Systematic polar coding," *IEEE Commun. Lett.*, vol. 15, no. 8, pp. 860–862, Aug. 2011.

[128] G. Sarkis, I. Tal, P. Giard, A. Vardy, C. Thibeault, and W. J. Gross, "Flexible and low-complexity encoding and decoding of systematic polar codes," *IEEE Trans. Commun.*, vol. 64, no. 7, pp. 2732–2745, July 2016.

[129] D. Shin, S. Lim, and K. Yang, "Design of length-compatible polar codes based on the reduction of polarizing matrices," *IEEE Trans. Commun.*, vol. 61, no. 7, pp. 2593–2599, 2013.

[130] V. Bioglio, F. Gabry, and I. Land, "Low-complexity puncturing and shortening of polar codes," in *2017 IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*, 2017, pp. 1–6.

[131] V. Miloslavskaya, "Shortened polar codes," *IEEE Trans. Inf. Theory*, vol. 61, no. 9, pp. 4852–4865, Sep. 2015.

[132] N. Presman, O. Shapira, and S. Litsyn, "Mixed-kernels constructions of polar codes," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 2, pp. 239–253, 2016.

[133] F. Gabry, V. Bioglio, I. Land, and J. Belfiore, “Multi-kernel construction of polar codes,” in *2017 IEEE International Conference on Communications Workshops (ICC Workshops)*, 2017, pp. 761–765.

[134] V. Bioglio, F. Gabry, I. Land, and J. C. Belfiore, “Multi-kernel polar codes: Concept and design principles,” *IEEE Trans. Commun.*, vol. 68, no. 9, pp. 5350–5362, 2020.

[135] S. B. Korada, E. Sasoglu, and R. Urbanke, “Polar codes: Characterization of exponent, bounds, and constructions,” *IEEE Trans. Inf. Theory*, vol. 56, no. 12, pp. 6253–6264, 2010.

[136] H. Lin, S. Lin, and K. A. S. Abdel-Ghaffar, “Linear and nonlinear binary kernels of polar codes of small dimensions with maximum exponents,” *IEEE Trans. Inf. Theory*, vol. 61, no. 10, pp. 5253–5270, 2015.

[137] V. Bioglio, C. Condo, and I. Land, “Design of polar codes in 5G new radio,” *IEEE Commun. Surveys Tuts.*, vol. 23, no. 1, pp. 29–40, 2021.

[138] G. Durisi, T. Koch, and P. Popovski, “Toward massive, ultrareliable, and low-latency wireless communication with short packets,” *Proc. IEEE*, vol. 104, no. 9, pp. 1711–1726, Sept 2016.

[139] X. Wu, L. Yang, Y. Xie, and J. Yuan, “Partially information coupled polar codes,” *IEEE Access*, vol. 6, pp. 63 689–63 702, 2018.

[140] L. Yang, Y. Xie, J. Yuan, X. Cheng, and L. Wan, “Chained LDPC codes for future communication systems,” *IEEE Commun. Lett.*, vol. 22, no. 5, pp. 898–901, May 2018.

[141] M. Mondelli, S. H. Hassani, and R. L. Urbanke, “Unified scaling of polar codes: Error exponent, scaling exponent, moderate deviations, and error floors,” *IEEE Trans. Inf. Theory*, vol. 62, no. 12, pp. 6698–6712, 2016.

[142] E. Arikan and E. Telatar, “On the rate of channel polarization,” in *2009 IEEE International Symposium on Information Theory*, June 2009, pp. 1493–1495.

[143] S. H. Hassani, R. Mori, T. Tanaka, and R. L. Urbanke, “Rate-dependent analysis of the asymptotic behavior of channel polarization,” *IEEE Trans. Inf. Theory*, vol. 59, no. 4, pp. 2267–2276, April 2013.

[144] S. H. Hassani, K. Alishahi, and R. L. Urbanke, “Finite-length scaling for polar codes,” *IEEE Trans. Inf. Theory*, vol. 60, no. 10, pp. 5875–5898, Oct 2014.

[145] 3GPP TS 36.211, “LTE; evolved universal terrestrial radio access (E-UTRA); physical channels and modulation,” Release 13, 2016.

[146] 3GPP TS 38.211, "5G; NR; physical channels and modulation," Release 16, 2020.

[147] A. Cavatassi, T. Tonnellier, and W. J. Gross, "Asymmetric construction of low-latency and length-flexible polar codes," in *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*, 2019, pp. 1–6.

[148] P. Trifonov, "Star polar subcodes," in *2017 IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*, 2017, pp. 1–6.

[149] ——, "Chained polar subcodes," in *SCC 2017; 11th International ITG Conference on Systems, Communications and Coding*, 2017, pp. 1–6.

[150] ——, "Randomized chained polar subcodes," in *2018 IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*, 2018, pp. 25–30.

[151] H. Zheng, S. A. Hashemi, B. Chen, Z. Cao, and A. M. J. Koonen, "Inter-frame polar coding with dynamic frozen bits," *IEEE Communications Letters*, vol. 23, no. 9, pp. 1462–1465, 2019.

[152] X. Wu and J. Yuan, "Partially information coupled bit-interleaved polar coded modulation for 16-QAM," in *2019 IEEE Information Theory Workshop (ITW)*, 2019, pp. 1–5.

[153] P. Chen and B. Bai, "Design and performance of the polar coded modulation for high mobility communications," in *2018 IEEE 87th Vehicular Technology Conference (VTC Spring)*, June 2018, pp. 1–5.

[154] Y. Yajima and H. Ochiai, "On design of multilevel coded modulation based on CRC-concatenated polar codes," in *2020 IEEE 17th Annual Consumer Communications Networking Conference (CCNC)*, 2020, pp. 1–6.

[155] K. Chen, K. Niu, and J. Lin, "Polar coded modulation with optimal constellation labeling," in *National Doctoral Academic Forum on Information and Communications Technology 2013*, 2013, pp. 1–5.

[156] H. Mahdavifar, M. El-Khamy, J. Lee, and I. Kang, "Polar coding for bit-interleaved coded modulation," *IEEE Trans. Veh. Technol.*, vol. 65, no. 5, pp. 3115–3127, May 2016.

[157] P. Chen, M. Xu, B. Bai, and X. Ma, "Design of polar coded 64-QAM," in *2016 9th International Symposium on Turbo Codes and Iterative Information Processing (ISTC)*, Sep. 2016, pp. 251–255.

[158] K. Chen, K. Niu, and J. Lin, "An efficient design of bit-interleaved polar coded modulation," in *2013 IEEE 24th Annual International Symposium*

on Personal, Indoor, and Mobile Radio Communications (PIMRC), Sep. 2013, pp. 693–697.

[159] M. Seidl, A. Schenk, C. Stierstorfer, and J. B. Huber, "Polar-coded modulation," *IEEE Trans. Commun*, vol. 61, no. 10, pp. 4108–4119, Oct. 2013.

[160] Q. Wang, L. Chen, and X. Ma, "Chained LDPC codes via partial information coupling and partial parity superposition," in *2021 IEEE Global Communications Conference (GLOBECOM)*, 2021.

[161] A. Y. Sukmadji, U. Martínez-Peñas, and F. R. Kschischang, "Zipper codes: Spatially-coupled product-like codes with iterative algebraic decoding," in *2019 16th Canadian Workshop on Information Theory (CWIT)*, 2019, pp. 1–6.

[162] B. P. Smith, A. Farhood, A. Hunt, F. R. Kschischang, and J. Lodge, "Staircase codes: FEC for 100 Gb/s OTN," *J. Lightw. Technol.*, vol. 30, no. 1, pp. 110–117, Jan 2012.

[163] A. J. Feltstrom, D. Truhachev, M. Lentmaier, and K. S. Zigangirov, "Braided block codes," *IEEE Trans. Inf. Theory*, vol. 55, no. 6, pp. 2640–2658, 2009.