

# Polar coding for optical wireless communication

***Citation for published version (APA):***

Zheng, H. (2020). *Polar coding for optical wireless communication*. [Phd Thesis 1 (Research TU/e / Graduation TU/e), Electrical Engineering]. Technische Universiteit Eindhoven.

***Document status and date:***

Published: 15/12/2020

***Document Version:***

Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

***Please check the document version of this publication:***

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

***General rights***

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

[www.tue.nl/taverne](http://www.tue.nl/taverne)

***Take down policy***

If you believe that this document breaches copyright please contact us at:

[openaccess@tue.nl](mailto:openaccess@tue.nl)

providing details and we will investigate your claim.

# Polar Coding for Optical Wireless Communication

PROEFSCHRIFT

ter verkrijging van de graad van doctor aan de  
Technische Universiteit Eindhoven, op gezag van de  
rector magnificus, prof.dr.ir. F.P.T. Baaijens, voor een  
commissie aangewezen door het College voor  
Promoties in het openbaar te verdedigen  
op donderdag 15 december 2020 om 16.00 uur

door

Haotian Zheng

geboren te Hubei, China

Dit proefschrift is goedgekeurd door de promotoren en de samenstelling van de promotiecommissie is als volgt:

Voorzitter: prof.dr.ing. A.J.M. Pemen  
Promotor: prof.ir. A.M.J. Koonen  
Co-promotor: dr.ir. Z. Cao  
dr.ir. A. Balatsoukas Stimming  
Leden: prof.dr. M. Karlsson (Chalmers University of Technology)  
prof.dr.ir. F.M.J. Willems  
dr. C.M. Okonkwo  
dr. S.A. Hashemi (Stanford University)

Het onderzoek of ontwerp dat in dit proefschrift wordt beschreven is uitgevoerd in overeenstemming met de TU/e Gedragscode Wetenschapsbeoefening.

A catalogue record is available from the Eindhoven University of Technology Library.

ISBN: 978-90-386-5185-9

NUR: 959

Title: Polar Coding for Optical Wireless Communication

Author: Haotian Zheng

Eindhoven University of Technology, 2020.

Keywords: Optical wireless communication / Polar Codes / Complexity-adjustable decoder / Inter-frame polar coding / Fast successive cancellation decoder / Hardware implementation / Infrared light communication

Copyright © 2020 Haotian Zheng

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means without the prior written consent of the author.

Typeset using L<sup>A</sup>T<sub>E</sub>X, printed in The Netherlands.



*Dedicated to my parents and teachers*



---

## SUMMARY

---

The demand for wireless communications is growing rapidly. It is expected that before 2030 not only almost all inhabitants of the earth will carry at least one personal wireless device, but that also 50 billion autonomous devices will have a wireless connection to the Internet of Things (IoT). The rapid growth of wireless traffic has introduced immense pressure on the limited radio frequency spectrum and the opening-up of new radio spectrum alone cannot bear this heavy burden. Optical wireless communication (OWC), using a carrier in the optical spectrum, promises to become a key alternative as a complementary solution to mitigate the pressure on the scarce radio spectrum and to realize high-speed wireless communication. Compared to radio communications, OWC using optical beams offers a number of unique advantages: license-free huge bandwidth, physical security, improved privacy, electromagnetic interference immunity, and high energy efficiency because each light beam can be targeted to the intended user individually at the time he needs it. An OWC system is mainly deployed in indoor scenarios as a supplement to radio techniques to help dealing with the tremendous indoor data traffic volume. High reliability, high speed and low power are three main requirements of an OWC system and also great challenges for its signal processing modules. Amongst the baseband signal processing units, the channel coding units, especially the decoder, are key components to achieve the above requirements. Proposed by Arikan in 2008, polar codes have received a great deal of attention from both academia and industry in the past few years to the extent that they have been selected to be included in the channel codes adopted by the 5th Generation Wireless Communications Standard (5G). They have the advantages of good error-correction performance without error floor, no random-like construction or iterative decoding, and flexible code rate. However, being the most commonly used decoder of polar codes in practice, the successive cancellation list (SCL) decoder suffers from high complexity and high latency induced by its inherent principle. What's more, it has a bottleneck in error-correction performance at the short code length which is preferred by a latency-stringent communication system like OWC.

This Ph.D. research is aiming to design polar coding schemes with high error-correction performance, low latency and low computational complexity, especially advanced polar decoders, as a means to help realize high-reliability, high-speed and low-power OWC systems. The research results are multi-faceted, including theoretical innovations, hardware implementations and experimental demonstrations.

To combat the high computational complexity, we first identify the defects in the principle of an SCL decoder and also investigate another decoder with different principle and complementary performance, called successive cancellation stack (SCS) decoder. Then we combine the advantages of both decoders to create the list-aided successive cancellation stack (LSCS) decoder. It is a multi-mode decoder



which can make a flexible trade-off between the computational and time complexity, corresponding to energy and latency costs respectively, by adjusting its working mode, while maintaining a constant error-correction performance. Taking advantage of this property, we can choose different modes of the LSCS algorithm to meet different application requirements at a low computational complexity. Besides, an enhanced version of LSCS is also proposed to improve resource utilization and further reduce time complexity.

In order to address the bottleneck in error-correction performance at short code length, we jumped out of the conventional research idea focusing on improving the decoding performance of each separate frame. A novel inter-frame related polar coding is proposed which creates associations between two adjacent frames in a clever way by sharing some mutual information. Based on the proposed encoding scheme, a simple and efficient decoder is designed which makes use of this mutual information to execute re-decoding for decoding failed frames, thus increasing the average probability of successful decoding. Compared to the conventional polar coding, the performance improvement is significant even at short and moderate code lengths, while the increment in memory requirements and average computational complexity is negligible.

In order to overcome the high latency incurred by sequential bit-by-bit decoding nature of the successive cancellation decoder, many efforts have gone into the research of fast simplified successive cancellation decoders employing different special node types corresponding to codes with special frozen bit patterns. A fast multi-bit processor is designed for each node type considering their specific code structure, thus reducing the total latency. The current research trend is to discover more and more special node types and equip the fast SC decoder with their dedicated processor. However, we try to find the common point of those special nodes and come up with a generalized node type, called sequence repetition (SR) node, and almost all existing nodes are special cases of this one. Furthermore, we prove an important property of an SR node which enables the design of a fast decoder with the highest parallelism and lowest decoding latency ever, using only one processor. In addition, for general nodes outside the class of SR nodes, a threshold-based hard-decision-aided (TA) scheme is introduced to speed up their decoding process, especially under good channel conditions. Unlike existing TA schemes whose effect on performance is unpredictable, we derive the threshold value that guarantees a given error-correction performance in the proposed scheme theoretically.

In consideration for practical implementation, a hardware architecture that exploits the proposed SR node has been designed and specified for the 5G polar codes. At the end of this thesis, the first polar-coded modulation based infrared light communication system is experimentally demonstrated.

---

# CONTENTS

---

<b>Summary</b>	<b>vii</b>
<b>List of Abbreviations</b>	<b>xiii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Optical wireless communication . . . . .	1
1.2 Channel coding in OWC . . . . .	4
1.2.1 Role of channel coding in OWC . . . . .	4
1.2.2 Channel codes classification . . . . .	5
1.2.3 Reported work on channel codes in OWC . . . . .	6
1.3 Polar Codes: challenges and motivation . . . . .	8
1.3.1 Brief introduction of polar codes . . . . .	8
1.3.2 Challenges in polar codes . . . . .	10
1.4 Outline and contributions of the thesis . . . . .	12
<b>2 A Review of Polar Codes</b>	<b>15</b>
2.1 The theory of channel coding . . . . .	15
2.1.1 Channel model and capacity . . . . .	15
2.1.2 Channel coding . . . . .	17
2.2 Polar Codes . . . . .	18
2.2.1 Channel polarization . . . . .	19
2.2.2 Polar codes construction . . . . .	23
2.2.3 Polar encoding and decoding . . . . .	25
2.3 Summary . . . . .	35
<b>3 Channel Coding Requirements for Optical Wireless Communications</b>	<b>37</b>
3.1 Typical optical wireless communication system . . . . .	38
3.2 Channel properties . . . . .	39
3.3 Requirements on channel coding . . . . .	40
<b>4 Complexity-Adjustable SC Decoding of Polar Codes</b>	<b>43</b>
4.1 Introduction . . . . .	43
4.2 SCL and SCS decoding algorithms . . . . .	45
4.3 LLR-threshold based path extension scheme . . . . .	47
4.4 List-aided successive cancellation stack decoding algorithm . . . . .	51
4.5 Enhanced list-aided successive cancellation stack decoding algorithm	54
4.6 Numerical results . . . . .	57
4.7 Conclusion . . . . .	62

<b>5</b>	<b>High Error-Correction Performance Decoder of Polar Codes</b>	<b>63</b>
5.1	Introduction . . . . .	63
5.2	Inter-frame polar coding . . . . .	64
5.2.1	Inter-frame correlated encoding scheme . . . . .	64
5.2.2	Inter-frame assisted decoding scheme . . . . .	65
5.3	Complexity analysis . . . . .	69
5.4	Simulation results . . . . .	69
5.5	Conclusion . . . . .	72
<b>6</b>	<b>Fast Successive-Cancellation Decoder of Polar Codes</b>	<b>73</b>
6.1	Introduction . . . . .	73
6.2	Binary tree representation and fast SC decoding . . . . .	75
6.3	Fast SC decoding with sequence repetition nodes . . . . .	77
6.3.1	Sequence repetition (SR) node . . . . .	77
6.3.2	Source node . . . . .	78
6.3.3	Repetition sequence . . . . .	80
6.3.4	Decoding of SR nodes . . . . .	81
6.4	Hard-decision-aided fast SC decoding with sequence repetition nodes	82
6.4.1	Proposed threshold-based hard-decision-aided scheme . . . . .	83
6.4.2	Multi-stage decoding . . . . .	86
6.5	Decoding latency . . . . .	87
6.5.1	No resource limitation . . . . .	87
6.5.2	With hardware resource constraints . . . . .	89
6.6	Results and comparison . . . . .	89
6.7	Conclusion . . . . .	95
<b>7</b>	<b>Hardware Implementation of Decoder for Polar Codes</b>	<b>97</b>
7.1	Introduction . . . . .	97
7.2	Architecture of SRFSC decoder . . . . .	98
7.2.1	Memory, processing, and PSN modules . . . . .	99
7.2.2	Controller module . . . . .	99
7.2.3	SR module . . . . .	101
7.3	Implementation results . . . . .	103
7.4	Conclusion . . . . .	104
<b>8</b>	<b>Optical Wireless Communication System Validation of Polar Codes</b>	<b>107</b>
8.1	Introduction . . . . .	107
8.2	Principle . . . . .	108
8.2.1	Bit-interleaved polar-coded modulation . . . . .	108
8.3	Experimental setup and results . . . . .	110
8.4	Conclusion . . . . .	114
<b>9</b>	<b>Summary and Future Work</b>	<b>115</b>
9.1	Summary of contributions . . . . .	115

---

9.2	Future work . . . . .	117
9.2.1	Complexity . . . . .	117
9.2.2	Error-correction performance . . . . .	117
9.2.3	Latency . . . . .	118
9.2.4	Experiment . . . . .	118
9.2.5	Polar codes using MIMO and NOMA techniques . . . . .	118
9.2.6	Other challenges . . . . .	119
	<b>References</b>	<b>121</b>
	<b>List of publications</b>	<b>137</b>



---

## LIST OF ABBREVIATIONS

---

5G	Fifth Generation
1D	One Dimensional
2D	Two Dimensional
3D	Three Dimensional
ALU	Arithmetic Logic Unit
AWGN	Additive White Gaussian Noise
AWGR	Arrayed Waveguide Grating Router
APD	Avalanche Photodiode
BEC	Binary Erasure Channel
BER	Bit Error Rate
BCH	Bose-Chaudhary-Hocquenhem
BIPCM	Bit-Interleaved Polar-Coded Modulation
B-DMC	Binary Discrete Memoryless Channel
BP	Belief Propagation
BPSK	Binary Phase Shift Keying
BLER	Block Error Rate
CLK	Clock Cycle
CAPEX	Capital Expenditures
CC	Convolutional Code
CS	Compensation Symbol
CWC	Constant Weight Code
CDMA	Code Division Multiple Access
CRC	Cyclic Redundancy Check
CS	Compare-and-Select
CMOS	Complementary Metal Oxide Semiconductor
DE	Density Evolution
DMT	Discrete Multitone
DMC	Discrete Memoryless Channel
DC	Direct Current
DPO	Digital Phosphor Oscilloscope
ECC	Error-Correction Code
eMBB	Enhanced Mobile Broadband
EDFA	Erbium-Doped Fiber Amplifier
FEC	Forward Error Correction
FSC	Fast Successive Cancellation
FoV	Field-of-View
FPGA	Field Programmable Gate Array
FS	Free-Space

GA	Gaussian Approximation
GFSC	Generalized Fast Successive Cancellation
ILC	Infrared Light Communication
BS-ILC	Beam-Steered Infrared Light Communication
IrDA	Infrared Data Association
ISI	Intersymbol Interference
IM/DD	Intensity Modulation with Direct Detection
IFA	Inter-Frame Assisted
IFFT	Inverse Fast Fourier Transform
LLR	Log-Likelihood Ratio
LED	Light Emitting Diode
LOS	Line-of-Sight
LD	Laser Diode
LDPC	Low Density Parity Check
LR	Likelihood Ratio
LSCS	List-aided Successive Cancellation Stack
LTPE	LLR-Threshold based Path Extension
LUT	Look Up Table
ML	Maximum Likelihood
MAP	Maximum A Posteriori
MIMO	Multiple Input Multiple Output
MCU	Metric Computation Unit
MD	Mobile Device
MZM	Mach Zehnder Modulator
MMF	Multimode Fiber
MRFB	Most Reliable Frozen Bit
MUUB	Most Unreliable Unfrozen Bit
NRZ	Non-Return-to-Zero
NOMA	Non-Orthogonal Multiple Access
NSPC	Non-Systematic Polar Codes
OWC	Optical Wireless Communications
OFDM	Orthogonal Frequency Division Multiplexing
OOK	On-Off Keying
OCT	Orthogonal Circulant Matrix Transform
OXC	Optical Crossconnect
CCC	Central Communication Controller
PIN	P-type/intrinsic/n-type Diode
PAM	Pulse-Amplitude Modulation
PPM	Pulse-Position-Modulation
PDF	Probability Density Function
PM	Path Metric
PE	Processing Element
PSN	Partial Sum Network

---

PRA	Pencil-Radiating Antenna
PRBS	Pseudo-Random Binary Sequence
PSN	Partial Sum Network
P/S	Parallel to Serial
QC	Quasi-Cyclic
QAM	Quadrature-Amplitude Modulation
RPM	Relative Path Metric
RGB	Red+Green+Blue
RF	Radio Frequency
RLL	Run Length Limited
RM	Reed-Muller
REP	Repetition
RAM	Random Access Memory
RCPC	Rate-compatible Punctured Convolutional
ROP	Received Optical Power
SC	Successive Cancellation
SCL	Successive Cancellation List
SCS	Successive Cancellation Stack
SCAN	Soft Cancellation
SR	Sequence Repetition
SNR	Signal-to-Noise Ratio
SMPC	Systematic Polar Codes
SPC	Single Parity Check
SM	Sign and Magnitude
SMF	Single-Mode Fiber
SRFSC	SR Node-Based Fast SC
S/P	Serial to Parallel
TA	Threshold-Based Hard-Decision-Aided
TIA	Transimpedance Amplifier
TB	Transport Block
TLS	Tunable Laser Source
TS	Training Sequence
UVC	Ultraviolet Communication
URLLC	Ultra-Reliable Low-Latency Communication
VLC	Visible Light Communication
VOA	Variable Optical Attenuator
WDM	Wavelength Division Multiplexing





INTRODUCTION

1.1 Optical wireless communication

Communication via light has been a convenient and low-cost way of exchanging information without the need of a guided channel since the ancient years. The historical forms include beacon fires, smoke, ship flags and semaphores. In 1880, Alexander Graham Bell invented the photophone, the world’s first telephone system transmitting voice over a visible light wave [1], which is a prototype of modern optical wireless communication. However, the concept of modern optical wireless communications (OWC) contains carriers covering the whole optical band, which includes not only the most commonly used visible light but also infrared and ultraviolet [2] as shown in Figure 1.1.

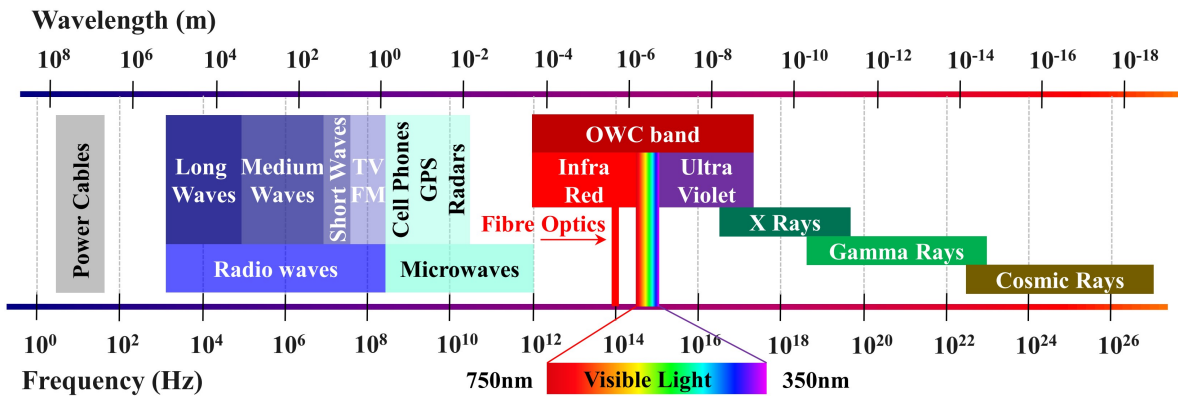


Figure 1.1 OWC in electromagnetic spectrum from [3].

The success of radio communications made the optical communication systems almost forgotten in the first half of the twentieth century. Things turned around after the middle of the twentieth century, when radio communication systems came into saturation in capacity mainly due to the depletion of radio spectrum resources. A surge of interest was drawn to the data transmission adopting carriers in the optical spectrum, and the invention of laser [4] and Light Emitting Diodes (LEDs) [5, 6] have accelerated this trend. In 1962, MIT Lincoln Labs built an experimental OWC link using GaAs LED and was able to transmit TV signals over a distance of 48 km. The first laser link to handle commercial traffic was built in Japan by Nippon Electric Company around 1970, with a distance of 14 km [7]. Although many trials of long distance OWC were conducted using different types of lasers and modulation schemes [8], the results were in general disappointing due to large

Table 1.1 Comparison between OWC and RF communication systems

Property	OWC	RF Communications
Transmitter	Mainly LED, Laser	RF antenna
Receiver	Mainly PIN, APD	RF antenna
Bandwidth regulated	No	Yes
Data rate	Low-High	Low-Medium
Scenario	Mainly Indoor	Indoor and outdoor
Penetration	Weak	Strong
Mobility	Limited	High
Power consumption	Low	Medium
Dominant noise	Background light	All electrical devices
Multipath fading	No	Yes
Security	High	Low
Health safety	Potential eye hazard	No obvious hazard

(PIN: p-type/intrinsic/n-type diode; APD: avalanche photodiode.)

divergence of laser beams and the inability to cope with atmospheric effects, such as air turbulence, rain, fog. With the development of low-loss fiber optics in the 1970s, they became the obvious choice for long distance optical transmission and shifted the focus away from OWC systems. Nevertheless, OWC is still a strong competitor to the “last mile solution” both for indoor and outdoor communications. The latest transmission rate record is 8.9 Tbit/s offered by 2D steerable infrared beams with a reach beyond 2.5 m shown in [9]. OWC offers a number of unique advantages over its Radio Frequency (RF) counterpart, such as a much wider unregulated bandwidth for high data rate, secure connectivity, absence of electromagnetic interference, low latency. A general comparison between OWC technologies and RF communications is listed in Table 1.1.

Amongst the three OWC technologies: visible light communication (VLC), infrared light communication (ILC) and ultraviolet communication (UVC), VLC and ILC are most attractive as they can provide high-capacity wireless communication solution for the massive indoor data traffic volume. VLC has a spectrum range between 400 nm (750THz) and 700 nm (430THz), and may operate on the existing lighting infrastructure. Since it uses LEDs as the transmission source typically, VLC was driven by the progress of LED for solid state lighting. Visible light from an LED traffic signal head was modulated for audio broadcast in an early experiment by Pang et al in 1999 [10]. Tanaka and Komine proposed VLC using white LED in 2001 and 2004, respectively [11, 12]. The use of Red+Blue+Green (RGB) LEDs in VLC system enabled the introduction of Wavelength Division Multiplexing (WDM) technology, which demonstrated a 3.22 Gbps transmission over more than 25 cm distance in 2013 [13]. Compared to prementioned LEDs with a bandwidth on the order of 10 MHz, the current  $\mu$ LED and Resonant-Cavity (RC)-LED have a band-

width of approximately 100 MHz. In [14], they were employed to achieve a data rate of 11.2 Gbps over 1.5 m reach. In addition, advanced modulation techniques like Orthogonal Frequency Division Multiplexing (OFDM) [15] and Discrete Multitone (DMT) modulation format [16] were investigated to apply to VLC systems to boost the transmission rate.

OWC operating in the infrared (IR) spectrum commonly centers at 850 nm (353 THz), 1310 nm (229 THz) and 1550 nm (193 THz). Indoor infrared communication was first proposed by F.R. Gfeller and U. Bapst in 1979, operating at 950 nm and achieving 1 Mbps [17]. The Infrared Data Association (IrDA) standard was formed in 1993, thereafter the standard was widely adopted for short range communication. In 2010, the European Community project OMEGA demonstrated a bi-directional OWC one-dimensional (1D) system operated at 1.25 Gbps at Non-Return-to-Zero (NRZ) On-Off-Keying (OOK) format [18]. More recently, OWC systems based on narrow infrared beam steering was reported in the Beam-steered Reconfigurable Optical-Wireless System for Energy-efficient communication (BROWSE) Advanced Grant project [19], led by Professor Ton Koonen from Eindhoven University of Technology. In these systems, a passive diffractive module is used to steer each beam two-dimensionally by just changing its wavelength remotely in the associated transmitter. The 2D beam steering with two crossed gratings was first proposed in [19–21], which established a point-to-point high capacity link. Its directivity enables a high SNR, which leads to high data rates and power efficiency. A speed of 42.8 Gb/s can be achieved, 100 times faster than current wireless networks generally achieve [22, 23]. However, the realization of beam steering using two crossed gratings requires highly-efficient gratings with low polarization dependency and careful highly stable mechanical adjustment of these gratings. As an alternative approach, 2D beam steering with high port-count arrayed waveguide grating router (AWGR) modules which are readily available commercially was proposed in [9]. This setup offers similar performance (up to 35 Gbit/s NRZ-OOK and 112 Gbit/s PAM-4 per beam), but is easier to assemble and requires less alignment effort. Building on this high-capacity IR beam-steered OWC system, a novel device localization concept was further introduced [24]. It only requires a simple fully passive function at the user device by means of a passive retroreflector based on an optical corner cube (CC) since a CC reflects light rays in the same direction as they came from. Compared to VLC, infrared light communication (ILC) using beams with wavelengths  $\lambda > 1400$  nm can have much higher emission powers, up to 10 dBm allowed by eye safety standards. Moreover, the use of IR laser instead of LED leads to a narrow beam, thus high bandwidth transmission can be achieved. BS-ILC can access up to 20.9 THz bandwidth in the 1460–1625 nm range by employing the well-established S+C+L band fiber-optic communication technologies. As concluded in [25], VLC systems may be preferred regarding capital expenditures (CAPEX) due to their lower infrastructure costs, whereas BS-ILC systems may be more beneficial regarding operating expenditures (OPEX) due to their lower energy consumption, as well as due to their higher energy efficiency, capacity density and higher privacy.

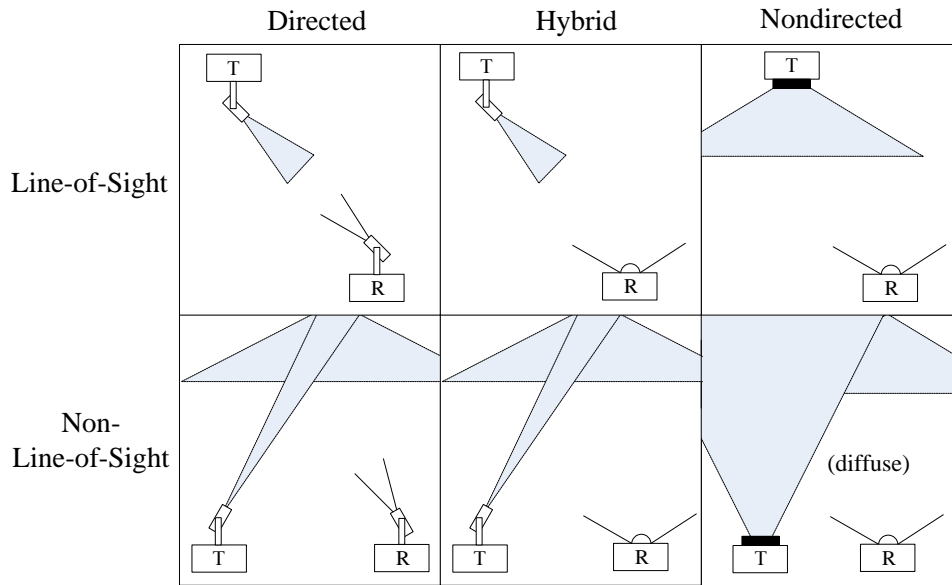


Figure 1.2 Classification of OWC links according to the degree of directionality of the transmitter and receiver and whether there exists a LoS path between them [7].

In terms of practical system implementation, an overview of different link configurations for indoor optical wireless systems is shown in Figure 1.2. Links are classified by two criteria. First, the degree of directionality of the transmitter and receiver. Directed links employ directional transmitters and receivers, while nondirected links employ wide-angle transmitters and receivers. In comparison, a directed link has a higher power efficiency and a nondirected link is more convenient for mobile devices. A hybrid link is a combination of the two forms. The second criterion is whether there exists an uninterrupted line-of-sight (LoS) path between the transmitter and receiver. LoS link is susceptible to blockage and shadowing but provides a higher link power budget. Non-LoS links mainly rely upon reflection of the light from diffusely reflecting surface, thus has a better link robustness and ease of use. In BS-ILC systems, advanced localization technology helps to track the position of device and maintain the connection of mobile users. And the severe impact of blockage is solved by handover between different transmitters.

## 1.2 Channel coding in OWC

### 1.2.1 Role of channel coding in OWC

Basically there are two types of coding. 1, Source coding: to code the information of a source as efficiently as possible (e.g. data compression). 2, Channel coding: to code the transmitted data such that it is as robust as possible against transmission errors. This thesis is about channel coding. For an OWC system with noise caused by ambient light, channel coding with forward error correction (FEC) are vital

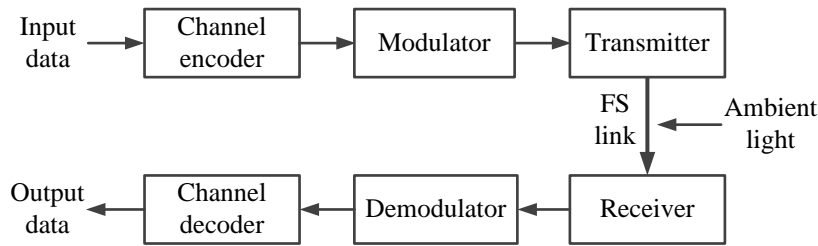


Figure 1.3 Schematic diagram of OWC system.

components to improve communication reliability, which can be described by a quantifiable and intuitive metric, called the bit error rate (BER). The most simple channel coding scheme is the repetition code. The encoder makes multiple copies of blocks of the data. For example, for a system whose uncoded BER is  $P_e$ , a single bit  $a$  is copied three times and sent as  $aaa$ . The decoder will choose the bit value that happened most frequently as the estimate of  $a$ . Hence, the estimate is correct if either no copies of  $a$  are in error or if one copy of  $a$  is in error and the probability of correct estimate is

$$P_{c,\text{coded}} = (1 - P_e)^3 + 3(1 - P_e)^2 P_e = 1 - 3P_e^2 + 2P_e^3. \quad (1.1)$$

It can be verified that  $P_{e,\text{coded}} = 1 - P_{c,\text{coded}} < P_e$  for any  $P_e \in [0, 0.5)$ , which means the repetition code improves the BER of the system. The performance improvement is obtained at the cost of extra bits. As a result of this data redundancy, the ratio between the number of the information bits and the transmitted bits, called code rate, is reduced. Generally, a lower code rate can bring better performance. What is of the highest concern is the maximum code rate for a reliable communication. This question is answered by Shannon in his seminal work [26]. He proved that reliable transmission is possible at any rate that is strictly smaller than the capacity of the channel, which is detailed in Chapter 2. However, random codes used by Shannon in the proof are not practical.

## 1.2.2 Channel codes classification

There has been a great deal of effort in the search for optimal yet practical (i.e., easily implementable) channel codes [27]. Classical algebraic block codes encode messages into blocks, such as Hamming codes, Reed-Muller codes, Reed-Solomon codes and BCH (Bose-Chaudhary-Hocquenhem) codes, whose aim is to find specific codes that maximize the minimum Hamming distance [28]. More recently, convolutional codes have been invented that get very close to Shannon's capacity limit [29]. The convolution codes do not encode in blocks, instead they read and transmit bits continuously, where transmitted bits are a linear combination of previous source bits. Similar codes including Turbo codes [30, 31], Low Density Parity Check (LDPC)

codes [32, 33], etc., are called probabilistic codes since they were more directly inspired by Shannon's probabilistic approach to coding. Unlike algebraic coding, probabilistic coding is more concerned with finding classes of codes that optimize average performance. In general, algebraic codes have lower encoding/decoding complexity than probabilistic codes, while, probabilistic codes like Turbo and LDPC have much better error-correction performance, even capacity-approaching in some specific cases [34]. LDPC codes are selected to be included as a coding scheme for the 5th Generation of Wireless Communications Standard (5G) [35]. The latest breakthrough in channel coding came with Arıkan's polar codes [36], which are provably capacity-achieving over a very wide range of transmission channels. Polar codes have advantages over the probabilistic codes in the sense of having no random-like construction, low-complexity encoding, good error-correction performance without error floor and flexible code rate. Currently, polar codes are selected as the coding scheme for the control link in Enhanced Mobile Broadband (eMBB) channel of the 5G standard which requires codes of short lengths.

### 1.2.3 Reported work on channel codes in OWC

The IEEE 802.15.7 standard for VLC [37, 38] introduces two FEC coding schemes, Reed-Solomon (RS) codes and convolutional codes (CC). An adaptive RS code based orthogonal frequency division multiplexing (OFDM) scheme has been proposed to optimize the tradeoff between BER and data rate in VLC system [39–42]. LDPC-coded VLC systems were first demonstrated in [43], supporting a data rate over 1Gbps. VLC systems to mitigate the influence of interference in LED arrays using LDPC were studied in [44]. Recently, a class of protograph based low-density parity-check (P-LDPC) codes was employed in RGB-LED-based VLC system to improve system performance [45]. Since a VLC systems need to take illumination demand into consideration, dimming control and flicker is a challenging problem for a VLC system that needs variable controlled dimming. The number and distribution of zeros and ones in the transmitted bits must be appropriately arranged for providing target dimming levels. In the IEEE VLC standard, the dimming function is provided via run length limited (RLL) codes [46–48] and compensation symbols (CS) for OOK modulation. The standard does not consider modification of FEC codes according to dimming support. Different error-correction schemes with dimming control have been proposed for VLC systems to achieve reliable data transmission while also providing the flicker-free operation and good dimming control. FEC schemes based on modified Reed-Muller (RM) codes were designed for dimming support in VLC systems in [49, 50]. Another coding scheme based on rate-compatible punctured convolutional (RCPC) code was introduced for providing simple dimming control solution [51]. The punctured bits in the codeword are replaced with compensation symbols generated by dimming controller to achieve the target dimming level. In [52], the proposed turbo coded system employs puncturing and scrambling techniques to match the Hamming weight of codewords with targeted dimming rates. Based on Quasi-cyclic (QC) LDPC codes, an adaptive FEC scheme was proposed

in [53] to efficiently adjust dimming values in VLC systems. The previous works focused on the modification of FECs. Joint FEC-RLL coding solutions were pioneered in [54], where a concatenation of an outer RS code and an inner RLL code was proposed. The serial concatenation of convolutional codes and Miller codes was explored in [55]. A new coding scheme for dimmable VLC systems based on serial concatenation of columnscaled (CS) low-density parity-check (LDPC) codes and constant weight codes (CWCs) was proposed in [56], whose coding rates are not constrained by the dimming range as error control and dimming control are decoupled.

The design of error-correction codes for ILC does not need to consider dimming control since infrared light is invisible. The Advanced Infrared (AIr) standard by the Infrared Data Association (IrDA) proposed a rate-adaptive transmission scheme based on variable-rate repetition coding, providing robust links at data rates between 250 kb/s and 4 Mb/s by using pulse-position modulation (PPM). Additionally, the HHH(1, 13) modulation scheme, based on run-length limited (RLL) coding, has been also proposed by the IrDA in the Very Fast Infrared (VFIr) standard for data transfer of 16 Mb/s, but operating over line-of-sight links of up to 1 m. Rate-compatible punctured convolutional codes (RCPC) and adaptive pulse position modulation (PPM) were applied to indoor infrared wireless communication systems to achieve high bit rate and realise communications even under bad channel conditions with limited transmitter power [57–59]. The concatenation of an outer punctured convolutional code and an inner repetition code was proposed in [58, 60] for indoor infrared wireless communication systems. Its code rate varies adaptively depending on channel conditions to achieve the required BER at the expense of bit rate. A RCPC coding scheme with a modified puncturing matrix was studied in [61], which achieves better bit-error rate results than conventional RCPC and convolutional coding schemes. Turbo code was first adopted in infrared wireless communication in [62], where iterative maximum a posteriori probability (MAP) decoding of Turbo coded OOK and Turbo coded binary PPM (BPPM) were presented. It was confirmed in [63] that turbo coding is very effective to improve the throughput-delay performance of an infrared Code Division Multiple Access (CDMA) network. A power-variable rate-adaptive low-density parity-check (LDPC)-coded OFDM scheme was used to deal with bandwidth limitations of indoor infrared links, supporting a high-speed optical signal (40 Gb/s and beyond) to an end-user [64].

Polar codes have only been applied to OWC systems in recent years. Polar codes with list+cyclic redundancy check (CRC) decoding were shown to outperform state-of-the-art LDPC codes in short block lengths, and shorter FEC codes are preferred for latency-stringent systems such as short-reach OWC [65]. In [66], based on polar codes, an efficient and flicker-free FEC coding scheme for dimmable VLC was proposed to increase the transmission efficiency and simplify the coding



structure. The coding gain<sup>1</sup> of the proposed scheme is about 4.6 dB and 1.4 dB higher than that of the RS codes-based scheme and the LDPC codes-based scheme, respectively. Wang and Kim have studied the joint design of polar codes and RLL codes in detail [67–69]. They also proposed a modified likelihood ratio (LR) for decoding of polar codes for mitigating intersymbol interference (ISI), which is a major impairment in visible light communication [70]. A recent study proposed an efficient construction of flicker-free polar codes to tackle the issue of flickering in VLC, which outperforms most state-of-the-art schemes in terms of error correction performance and implementation complexity [71]. The bit-interleaved polar-coded modulation (BIPCM) scheme based on orthogonal circulant matrix transform (OCT) precoding was proposed and experimentally demonstrated for a 256-QAM OFDM-VLC system, achieving a net data rate of 343 Mb/s over 80-cm free-space transmission with a BER below  $10^{-3}$  [72]. The applications of polar codes into Multiple Input Multiple Output (MIMO)-OFDM and non-orthogonal multiple access (NOMA)-enabled VLC systems were further studied in [73, 74]. In [75], a secure coding scheme based on polar codes was proposed to simultaneously achieve physical-layer security and transmission reliability for indoor VLC systems under Wyner’s wiretap model. So far, the application of polar codes in ILC systems has not been reported.

## 1.3 Polar Codes: challenges and motivation

### 1.3.1 Brief introduction of polar codes

Polar codes are proposed based on the method called channel polarization. Channel polarization refers to the fact that it is possible to synthesize, out of  $N$  independent copies of a given binary discrete memoryless channel (B-DMC), a second set of  $N$  binary-input channels such that, as  $N$  tends to infinity the capacity of these synthetic virtual channels polarizes to either 0 or 1. One can only send information through those virtual channels with capacity near 1. The remaining virtual channels transmit fixed values known to both the sender and receiver in advance. Codes constructed on the basis of this idea are called polar codes. We will give a detailed review on the theory of polar codes in Chapter 2. As the first provably capacity-achieving codes with low complexity, polar codes have received extensive attention since their inception. Unlike other codes, polar codes are constructed given a specific channel type or channel condition. For the example of a given additive white Gaussian noise (AWGN) channel, polar codes should be optimized point-by-point in the given SNR range in order to achieve the optimal performance at each SNR. Despite the fact that code construction is channel dependent, the encoder structure is universal, following the recursive structure of channel transformation shown in Chapter 2. As for the decoding of polar codes, there are many types of decoders with different principles

---

<sup>1</sup>In coding theory and related engineering problems, coding gain is the measure in the difference between the signal-to-noise ratio (SNR) levels between the uncoded system and coded system required to reach the same bit error rate (BER) levels when used with the error correction code.

and they directly determine the error-correction performance given a specific code construction. Thus, decoders are the research focus of polar codes.

**Construction and Encoding of Polar Codes:** In his seminal paper [36] Arıkan proposed a recursive calculation algorithm based on Bhattacharyya parameters to do channel-reliability evaluation and code construction for binary erasure channel (BEC). For more general channels, Arıkan proposed a Monte Carlo based approach which is universal but time-consuming. Mori and Tanaka proposed the use of density evolution (DE) tools for tracing the probability density function (PDF) of log-likelihood ratios (LLRs) used in the decoding of polar codes [76, 77]. However, its computational complexity is high in practical applications. Different approximation schemes have been taken to simplify the calculations [78–80]. The most practical way is called Gaussian approximation (GA), which enables one to compute only the expected value of LLRs using a recursive formula in the AWGN channel, thus drastically reducing the complexity [79]. However, when the code length is long ( $2^{14}$  and above code length), the conventional approximate GA which uses a two-segment approximation function is not accurate, resulting in a catastrophic performance loss. New principles to design the GA approximation functions for polar codes were proposed in [81]. After code construction, the encoding can be executed using a recursive structure to obtain the codeword. Polar codes in their standard form are non-systematic codes, in other words, the information bits do not appear as part of the codeword transparently. Systematic polar encoding whose codeword contains information bits was proposed in [82], which offers significant advantages in terms of bit error rate performance with respect to conventional non-systematic counterpart. Efficient algorithms for systematic polar encoding were introduced in [83].

**Decoding of Polar Codes:** It has been proved in [36] that polar codes with successive cancellation (SC) decoding can achieve the capacity of B-DMCs. Nevertheless, the performance of SC decoding for finite code length is not ideal. As an enhanced version of SC, successive cancellation list (SCL) decoding approaches the performance of maximum likelihood (ML) decoding when the list size is large [84]. With the help of cyclic redundancy check (CRC), CRC-aided SCL (CA-SCL) [85] decoding provides a performance beyond the ML performance for SCL without CRC and can outperform LDPC codes at short and moderate code length [65]. Thus, CA-SCL decoding has become the standard benchmark in the research of polar decoders. Aimed at the high complexity problem of SCL decoding, successive cancellation stack (SCS) [86] and successive cancellation flip (SCF) [87] decoding were proposed, which can effectively reduce the computational complexity. But as a tradeoff for complexity reduction, SCS decoding requires a longer decoding latency and larger memory, and SCF decoding has a worse error-correction performance. To overcome the sequential bit-by-bit decoding problem of SC and SCL decoding, simplified fast SC/SCL decoding is proposed, which speeds up the decoding process by making multi-bit decision at some special nodes corresponding to constituent codes with special frozen bit patterns [88–93]. Other decoding algorithms include belief propa-

gation (BP) decoding [94, 95], sphere decoding [96, 97], ordered statistic decoding [98] and soft cancellation (SCAN) decoding [99].

### 1.3.2 Challenges in polar codes

Despite the effort and interest for polar codes both in academic and industrial communities, there are still some challenges that need to be addressed to enable its use in high-reliability, high-speed and low-power OWC system, which are shown in Figure 1.4. The three challenges of polar codes correspond to the three requirements of OWC system. Note that in addition to low latency, high error-correction performance is also important to achieve the high-speed demand in the sense that polar codes with higher error-correction performance enable the use of a higher code rate or higher order modulation format given a target BER, thus boosting the transmission speed. The three challenges are analyzed in detail as follows:

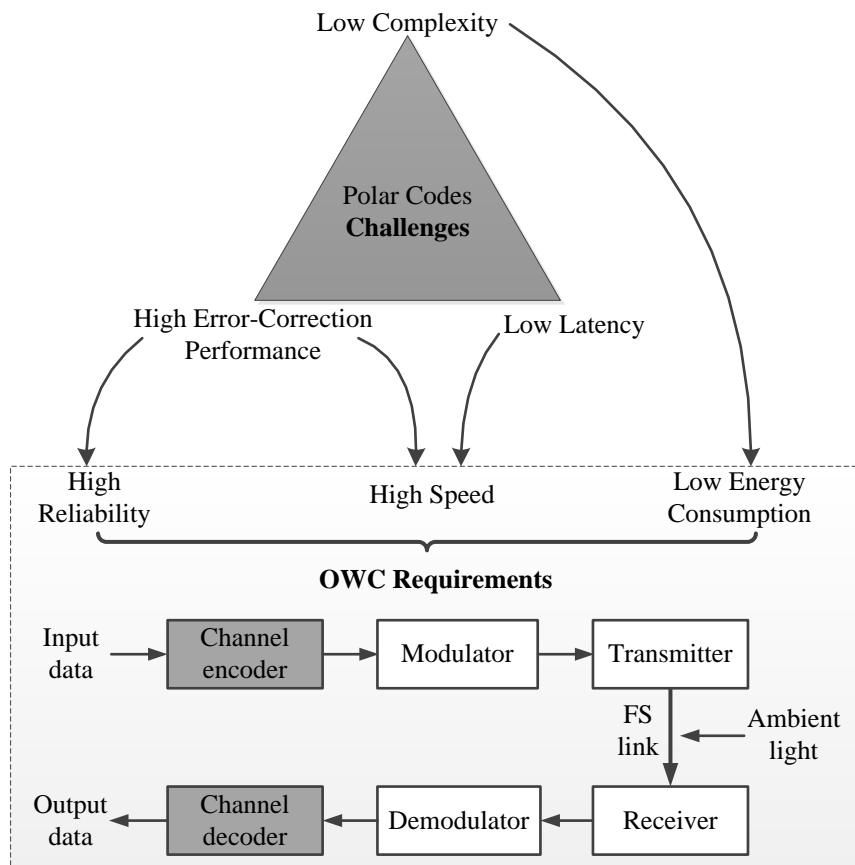


Figure 1.4 Challenges in Polar Codes.

**Challenge 1: Complexity**

CA-SCL decoder is currently the most commonly used decoder with a very good error-correction performance. This is mainly due to a parallel search of list size  $L$  decoding paths, which also causes a large computational complexity. Moreover, the complexity remains constant, even if the noise is low enough for the use of a simpler SC decoder. The high computational complexity leads to a high power consumption, which is not friendly for a mobile device. As one of the few decoders with similar error-correction performance to CA-SCL decoder, a CA-SCS decoder has a much lower complexity than CA-SCL decoder. Moreover, its complexity is channel adaptive, close to that of SC decoder when the channel condition is good. However, the vital disadvantage of high decoding latency hinders its use in latency-sensitive applications. We can observe that it is difficult for a decoder to have the best performance in all metrics such as error-correction, latency and complexity. Hence, the question is how to deal with the tradeoff between metrics and achieve the requirement of a diverse set of applications with as low complexity as possible.

**Challenge 2: Error-correction performance**

Polar codes take advantage of the polarization phenomenon by only transmitting information bits through the polarized subchannels with the highest capacity. However, for a practical finite codeword length, especially at short code length, there is a non-negligible fraction of subchannels whose capacity is not fully polarized. The information transmitted through such subchannels is unreliable. This problem limits the error-correction performance of finite length polar codes. The concatenation of polar codes with other codes is considered as an effective way to deal with the partial polarization problem [100–102]. CA-SCL decoding is an example which concatenates polar codes with CRC [85, 84]. The shortcomings of this scheme are a reduction of code rate and longer code length due to concatenation. Another way to improve the error-correction performance is to exploit the benefits of long codes through spatial coupling [103–105]. Theoretically, an infinite long codeword chain can be constructed by coupling an infinite number of finite length code blocks in a certain structure. Spatially coupled codes can achieve considerable coding gains over uncoupled counterparts while maintaining the decoding complexity in a reasonably low level by using windowed decoders [106, 107]. In this scheme, coupled information is shared by several consecutive code blocks in the same codeword. Hence, the effective code rate of the information coupled polar codes is reduced [108, 109]. Existing solutions all lead to a long code length and reduced code rate. However, for a latency-constrained system, such as indoor short-reach OWC system, short code length is required to maintain a low decoding latency. So a challenge is to improve the error-correction performance of polar code with short code length without sacrificing the code rate.

### Challenge 3: Latency

Conventional SC/SCL decoding implies sequential bit-by-bit decoding, which leads to a high decoding latency. Belief propagation (BP) decoding can be used instead of sequential decoding to reduce latency by parallelizing the decoding process using iterative receivers. However, the overall decoding complexity is high since it is proportional to the number of iterations. What is more, there is a gap between the error-correction performance of BP decoder and CA-SCL decoder. Another simple but efficient way is to make multi-bit parallel decoding at some special node types corresponding to constituent codes with special frozen bit patterns, which is called fast simplified SC/SCL decoding [88–93]. It exploits the characteristic of each special node type and designs their dedicated fast decoders. Although the nature of sequential decoding is not changed, fast SC/SCL decoding can provide a significant latency reduction with negligible performance loss compared to conventional SC/SCL decoding. However, an important problem is that there exist too many special node types, and a separate decoder for each of them will put a heavy burden on the hardware implementation. In addition, we are still on the way of finding a new special node with higher parallelism that can be exploited in decoding, thus further reducing the latency.

## 1.4 Outline and contributions of the thesis

The main aim of the dissertation is to address the three key challenges mentioned before and design polar coding with high error-correction performance, low latency and low computational complexity for high-reliability, high-speed and low-power wireless communication systems, especially indoor OWC systems. This thesis first provides treatment of these challenges from algorithm level and theoretical perspective. Hardware implementation and experiment demonstration are then presented to further expand the research and build a bridge between theory and practice. The research work carried out in this thesis has led to several contributions in this field. They are categorized in the coming chapters and sections, according to the challenges each contribution addresses, as follows:

**Chapter 2** presents an overview of polar codes. We start from the theory of channel coding, followed by the channel polarization theory, which explains why polar codes can achieve channel capacity when the block length tends to infinity. Then we move on to practical design of polar codes, including code construction, encoding and decoding schemes of polar codes in finite block length regime.

**Chapter 3** introduces the optical wireless communication system, to be more specific, the one using pencil beams, employed in the BROWSE project to implement an indoor communication network [19]. Its channel properties and requirements on channel coding are analyzed and discussed in detail.

**Chapter 4** proposes a complexity-adjustable successive cancellation decoder called enhanced list-aided successive cancellation stack (LSCS) decoder to meet

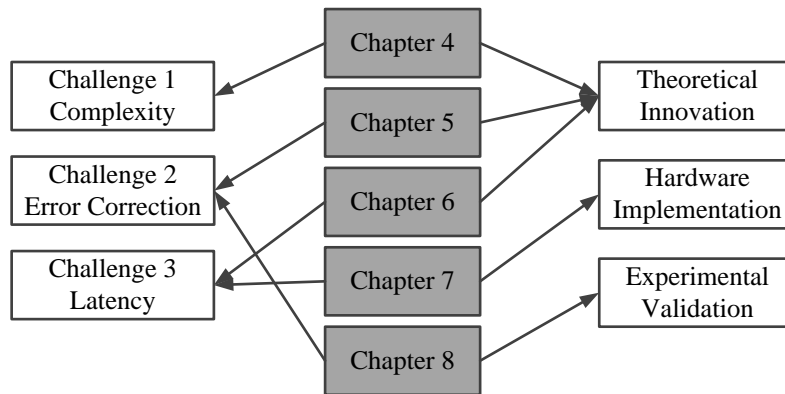


Figure 1.5 Challenges and their relations addressed in this thesis.

different application requirements at a low computational complexity. It firstly studies the LLR characteristics of the correct path in the decoding process. Exploiting the characteristics, an LLR-threshold based path extension scheme is designed to reduce the memory consumption of stack decoding. By employing both the ideas of SCL and SCS decoding, a novel LSCS decoding is introduced, which can provide a flexible tradeoff between time complexity and computational complexity. Moreover, LSCS decoding is improved to obtain an enhanced version to further decrease the time complexity.

**Chapter 5** presents a new inter-frame correlated polar coding scheme to improve the error-correction performance of polar codes. Two consecutive frames are correlated-encoded in the sense that the frozen bits of the second frame partially depend on the unfrozen bits of the first frame. Using this new encoding scheme, a novel decoding scheme is investigated, where consecutive frames can assist each other by performing a re-decoding on a decoding failed frame with the help of the shared information. Simulation results show that the proposed polar codes can yield a significant performance improvement compared to classical polar codes with negligible extra memory and complexity.

**Chapter 6** proposes a fast decoding algorithm based on a new class of sequence repetition (SR) node to reduce the decoding latency. The concept of sequence repetition node and the accompanying concepts of the repetition sequence and the source node are firstly presented. Most existing special node types are special cases of the proposed sequence repetition node. Moreover, an important property of the SR node is proven that enables the design of an efficient fast decoder of the SR node. In addition, for general nodes outside the class of SR, a threshold-based hard-decision-aided scheme is introduced to speed up their decoding process. The threshold value that guarantees a given error-correction performance in the proposed scheme is derived theoretically. Both numerical simulation and hardware implementation results in terms of decoding latency are provided.

**Chapter 7** presents the first FPGA implementation of the SR-node-based fast SC decoder. A dedicated architecture for the SR node processor is designed and an instruction structure is given. Moreover, the proposed architecture is specified for a 5G polar code with code length 1024 and three code rates  $1/4$ ,  $1/2$ ,  $3/4$ , and 64 processing elements. The FPGA implementation results including both resource consumption and decoding latency are provided and compared with previous works.

**Chapter 8** experimentally demonstrates the first polar coded modulation based infrared light communication system. A Monte Carlo method is proposed to jointly design an inter-frame related polar code with 16-ary quadrature-amplitude modulation (16-QAM) and orthogonal frequency-division multiplexing (OFDM). The indoor transmission of 9.6 Gbit/s 16-QAM OFDM signal is experimentally achieved over a 3.2 km single-mode fiber and 0.8 m free space with no errors over  $10^7$  bits.

Finally, **Chapter 9** summarizes the main contributions of this thesis. The possible areas for improvements and the potentially interesting open problems for future research are also discussed.

## A REVIEW OF POLAR CODES

---

### 2.1 The theory of channel coding

In real-world communication systems, the sent data will be corrupted by the noisy and dispersive channel. To recover the original data, an efficient way is to add redundant bits in the data before transmission, which is called channel coding [110]. The fraction of information bits to all transmitted bits is defined as the code rate of channel coding. A good channel coding scheme has a high code rate (as close as possible to 1) and low probability of error. Two important questions that follow are: First, the maximum code rate at which information bits can be transmitted reliably over a noisy channel. Second, a practical channel coding scheme which can achieve the maximum code rate. The remaining part of this chapter will address these two questions.

#### 2.1.1 Channel model and capacity

Consider a channel  $W$  with input alphabet  $\mathcal{X}$  and output alphabet  $\mathcal{Y}$ , a transition probability function  $W(y|x)$ ,  $x \in \mathcal{X}, y \in \mathcal{Y}$  is used to model channel  $W$  mathematically.  $W(y|x)$  is defined as the probability of observing  $y$  when  $x$  is sent. Throughout the thesis, we will focus on discrete memoryless channels (DMCs), defined as follows.

**Definition 2.1.1.** *In a discrete memoryless channel (DMC), denoted by  $W : \mathcal{X} \rightarrow \mathcal{Y}$ ,  $\mathcal{X}$  and  $\mathcal{Y}$  are two finite sets, i.e.  $|\mathcal{X}|, |\mathcal{Y}| < \infty$ . The output  $y_k \in \mathcal{Y}$  at time  $k$  only depends on the input  $x_k \in \mathcal{X}$  at time  $k$  and not on the behaviour of the channel in the past time slots, i.e. let  $x_1^k$  denote  $\{x_1, \dots, x_k\}$ , the probability of observing  $y_k$  when  $x_1^k$  is sent and  $y_1^{k-1}$  is observed is*

$$W(y_k|x_1^k, y_1^{k-1}) = W(y_k|x_k), \quad \forall k \in \{1, 2, \dots\}. \quad (2.1)$$

Note that if a DMC is used without feedback, there is

$$W(y_1^k|x_1^k) = \prod_{i=1}^k W(y_i|x_i), \quad \forall k \in \{1, 2, \dots\}. \quad (2.2)$$

If the input alphabet  $\mathcal{X}$  is always  $\{0, 1\}$ , the DMC is denoted as a binary-input discrete memoryless channel (B-DMC). A B-DMC is said to be symmetric if there exists a permutation  $\pi$  of the output alphabet  $\mathcal{Y}$  such that 1)  $\pi^{-1} = \pi$ . and 2)



$p(y|1) = p(\pi(y)|0)$ . Throughout this thesis, symmetric B-DMCs are used without feedback.

Before introducing the channel capacity, we first explain three fundamental concepts: entropy, conditional entropy and mutual information.

a) Entropy: The entropy of a random variable  $X \sim p(x)$  is defined as

$$H(X) = - \sum_{x \in \mathcal{X}} p(x) \log_2 p(x), \quad (2.3)$$

where  $p(x)$ ,  $x \in \mathcal{X}$  is the probability distribution function of  $X$ . This is a measure of the uncertainty in  $X$ .

b) Conditional entropy: The conditional of a random variable  $X$  given another random variable  $Y$ , where  $(X, Y) \sim p(x, y)$ ,  $x \in \mathcal{X}$ ,  $y \in \mathcal{Y}$  is defined as

$$H(X|Y) = \sum_{y \in \mathcal{Y}} p(y) \log_2 H(X|Y=y) = - \sum_{y \in \mathcal{Y}} p(y) \sum_{x \in \mathcal{X}} p(x|y) \log_2 p(x|y). \quad (2.4)$$

This is a measure of the uncertainty in  $X$  conditioned on knowing  $Y$ .

c) Mutual information: The mutual information between two random variables  $(X, Y) \sim p(x, y)$ ,  $x \in \mathcal{X}$ ,  $y \in \mathcal{Y}$  is defined as

$$I(X; Y) = H(X) - H(X|Y) = \sum_{x \in \mathcal{X}, y \in \mathcal{Y}} p(x, y) \log_2 \frac{p(x, y)}{p(x)p(y)}. \quad (2.5)$$

This is a measure of the amount of information about  $X$  revealed by the knowledge of  $Y$ .

**Definition 2.1.2.** The mutual information  $I(X; Y)$  of a DMC  $W : \mathcal{X} \rightarrow \mathcal{Y}$  and input distribution  $p(x)$  is defined as

$$I(X; Y) = \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p(x) W(y|x) \log_2 \frac{W(y|x)}{\sum_{x' \in \mathcal{X}} W(y|x') p(x')}. \quad (2.6)$$

Based on the above concepts, the channel capacity is defined as follows.

**Definition 2.1.3.** The channel capacity of a DMC  $W : \mathcal{X} \rightarrow \mathcal{Y}$  is defined as

$$C = \max_{p(x)} I(X; Y), \quad (2.7)$$

where the maximum is over the set of all input probability distribution  $p(x)$ .

As we will see below, the channel capacity plays a fundamental role in channel coding.

### 2.1.2 Channel coding

Unless the channel has no noise and dispersion, the data transmitted over the channel may be corrupted. To cope with the possible error incurred during transmission, channel coding is an effective solution. We first give the definition related to channel codes.

**Definition 2.1.4.** An  $(N, M)$  code for a discrete memoryless channel with input alphabet  $\mathcal{X}$  and output alphabet  $\mathcal{Y}$  is defined by an encoding function

$$f : \{1, 2, \dots, M\} \rightarrow \mathcal{X}^N, \quad (2.8)$$

and a decoding function

$$g : \mathcal{Y}^N \rightarrow \{1, 2, \dots, M\}, \quad (2.9)$$

where  $N$  is the block length of the codeword.  $\mathcal{M} = \{1, 2, \dots, M\}$  is the message set.  $f(1), f(2), \dots, f(M)$  are the codewords. The set of all codewords forms the codebook.

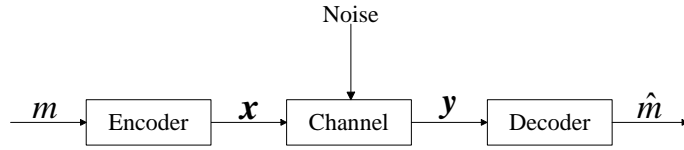


Figure 2.1 The communication system with channel coding.

In Figure 2.1, message  $m$  is randomly chosen from the message set  $\mathcal{M}$ . Thus,  $x_1^N = f(m)$  and the estimate  $\hat{m} = g(y_1^N)$ .

**Definition 2.1.5.** For all  $1 \leq m \leq M$ , let

$$\lambda_m = W \left\{ g(y_1^N) \neq m \mid x_1^N = f(m) \right\} \quad (2.10)$$

be the conditional probability of error given that the message is  $m$ .

**Definition 2.1.6.** The maximal probability of error of an  $(N, M)$  code is defined as

$$\lambda_{max} = \max_m \lambda_m. \quad (2.11)$$

**Definition 2.1.7.** The average probability of error of an  $(N, M)$  code is defined as

$$P_e = \frac{1}{M} \sum_{m \in \mathcal{M}} W \left( g(y_1^N) \neq m \mid x_1^N = f(m) \right) = \frac{1}{M} \sum_{m=1}^M \lambda_m. \quad (2.12)$$

Therefore,

$$P_e \leq \max_m \lambda_m = \lambda_{\max}. \quad (2.13)$$

**Definition 2.1.8.** *The rate of an  $(N, M)$  channel code is defined as*

$$R = \frac{\log_2 M}{N} \text{ bits per transmission}, \quad (2.14)$$

$R$  is the ratio between how many bits of message are transmitted and how many bits are used for encoding.

**Definition 2.1.9.** *A code rate  $R$  is achievable for a discrete memoryless channel if, for all  $n$ , there exists an  $(N, M)$  code with  $M = 2^{NR}$  such that  $\lambda_{\max} \xrightarrow{N \rightarrow \infty} 0$ .*

The operational capacity of a channel is the supremum over all achievable rates for the channel. In the landmark work [26], Shannon proved the noisy channel coding theorem which shows that the operational capacity is equal to the channel capacity for any discrete memoryless channel.

**Theorem 2.1.1.** *For any DMC, if  $R \leq C$ , then  $R$  is achievable. Conversely, if  $R > C$ , it is not achievable.*

The proof of achievability in the first part is in two steps. It first shows the existence of a sequence of codes that can transmit  $M = 2^{NR+1}$  messages (rate  $R + \frac{1}{N}$ ) where the average probability of error  $P_e$  goes to zero. Then, by removing the  $\frac{M}{2}$  messages with the worst probabilities of error, it constructs a new sequence of codes that transmit  $\frac{M}{2} = 2^{NR}$  messages (rate  $R$ ) where the maximal probability of error  $\lambda_{\max}$  goes to zero. For the converse part, the proof shows that, for an arbitrary sequence of codes with rate  $R > C$ ,  $P_e$  does not converge to 0. Since  $\lambda_{\max} > P_e$ , it also does not converge to 0. Therefore, the rate  $R$  is not achievable in this case.

The channel coding theorem proves the existence of codes that enables the information transmission at rates below capacity with an arbitrary small probability of error if the code length is large enough. However, the random codes used by Shannon are not practical. Hence, from the day this theorem was published, the search for the optimal and practical codes has been ongoing.

## 2.2 Polar Codes

Polar codes are channel codes proposed by Arıkan in 2008 [36]. They are designed on the basis of the channel polarization phenomenon and are the first provable

capacity-achieving channel codes with affordable encoding/decoding complexity. This section will illustrate the theory of channel polarization first, followed by the construction of polar codes and finally the encoding and decoding schemes of polar codes.

### 2.2.1 Channel polarization

We write  $W : \mathcal{X} \rightarrow \mathcal{Y}$  to denote a B-DMC with transition probabilities  $W(y|x)$ ,  $x \in \mathcal{X}$ ,  $y \in \mathcal{Y}$  and  $\mathcal{X} \in \{0,1\}$ . When using the input letters in  $\mathcal{X}$  with equal probability, the mutual information expression in (2.6) can be reshaped as follows

$$I(W) = \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} \frac{1}{2} W(y|x) \log_2 \frac{W(y|x)}{\frac{1}{2}W(y|0) + \frac{1}{2}W(y|1)}, \quad (2.15)$$

$I(W)$  is defined as the symmetric capacity. When  $W$  is a symmetric channel,  $I(W)$  is the highest rate at which reliable communication is possible across  $W$ , that is, it equals the Shannon capacity.

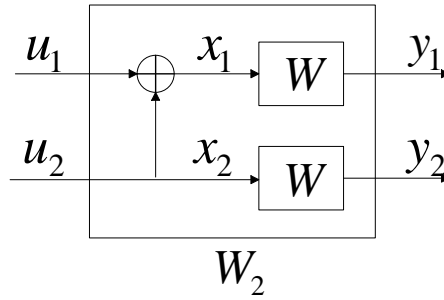


Figure 2.2 The channel  $W_2$ .

The operation of channel polarization is called channel transformation consisting of a channel combining phase and a channel splitting phase. Figure 2.2 shows the first level of the channel transformation. Two independent copies of  $W$  are combined and channel  $W_2 : \mathcal{X}^2 \rightarrow \mathcal{Y}^2$  is obtained with the transition probabilities.

$$W_2(y_1, y_2 | u_1, u_2) = W(y_1 | u_1 \oplus u_2) W(y_2 | u_2). \quad (2.16)$$

$W_2$  is then split into two synthesized subchannels  $W_2^{(1)}$  and  $W_2^{(2)}$  as below.

$$W_2^{(1)}(y_1^2 | u_1) = \sum_{u_2 \in \mathcal{X}} \frac{1}{2} W_2(y_1^2 | u_1^2) = \sum_{u_2 \in \mathcal{X}} \frac{1}{2} W(y_1 | u_1 \oplus u_2) W(y_2 | u_2). \quad (2.17)$$

$$W_2^{(2)}(y_1^2, u_1 | u_2) = \frac{1}{2} W_2(y_1^2 | u_1^2) = \frac{1}{2} W(y_1 | u_1 \oplus u_2) W(y_2 | u_2). \quad (2.18)$$

$W_2^{(1)}$  can be viewed as a synthesized channel with input  $u_1$  and output  $y_1^2$ .  $W_2^{(2)}$  can be viewed as a synthesized channel with input  $u_2$  and output  $y_1^2, u_1$ . The symmetric capacities of the channels before and after channel transformation have the following relationships.

**Theorem 2.2.1.** Suppose  $(W, W) \mapsto (W_2^{(1)}, W_2^{(2)})$ , then

$$I(W_2^{(1)}) + I(W_2^{(2)}) = 2I(W), \quad (2.19)$$

$$I(W_2^{(1)}) \leq I(W) \leq I(W_2^{(2)}). \quad (2.20)$$

This theorem [36] shows: after a single step of channel transformation,  $W$  is transformed into  $W_2^{(1)}$  and  $W_2^{(2)}$ , which are still B-DMCs. The symmetric capacity of  $W_2^{(1)}$  and  $W_2^{(2)}$  decreases and increases with respect to  $I(W)$ , respectively. But their sum remains unchanged, that is,  $2I(W)$ .

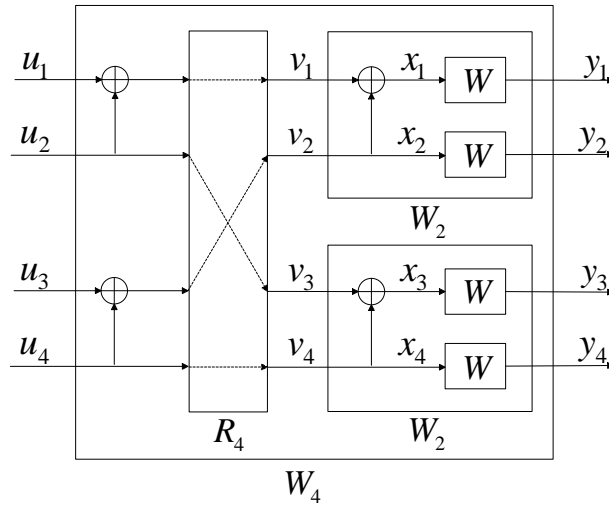


Figure 2.3 The channel  $W_4$  and its relation to  $W_2$  and  $W$ .

Since  $W_2^{(1)}$  and  $W_2^{(2)}$  are still B-DMCs, another step of channel transformation can be performed on  $W_2^{(1)}$  and  $W_2^{(2)}$ , respectively. Figure 2.3 shows the two-level channel transformation of four channel  $W$ , which executes  $(W_2^{(1)}, W_2^{(1)}) \mapsto (W_4^{(1)}, W_4^{(2)})$  and  $(W_2^{(2)}, W_2^{(2)}) \mapsto (W_4^{(3)}, W_4^{(4)})$ . Like this, if the one step channel transformation



$$W_N \left( y_1^N | u_1^N \right) = \prod_{i=1}^N W \left( y_i | x_i \right). \quad (2.23)$$

These transition probabilities can be calculated recursively according to the recursive structure in Figure 2.4 as below.

$$\begin{aligned} W_{2N}^{(2i-1)} \left( y_1^{2N}, u_1^{2i-2} | u_{2i-1} \right) \\ = \sum_{u_{2i}} \frac{1}{2} W_N^{(i)} \left( y_1^N, u_{1,o}^{2i-2} \oplus u_{1,e}^{2i-2} | u_{2i-1} \oplus u_{2i} \right) \cdot W_N^{(i)} \left( y_{N+1}^{2N}, u_{1,e}^{2i-2} | u_{2i} \right), \end{aligned} \quad (2.24)$$

$$\begin{aligned} W_{2N}^{(2i)} \left( y_1^{2N}, u_1^{2i-1} | u_{2i} \right) \\ = \frac{1}{2} W_N^{(i)} \left( y_1^N, u_{1,o}^{2i-2} \oplus u_{1,e}^{2i-2} | u_{2i-1} \oplus u_{2i} \right) \cdot W_N^{(i)} \left( y_{N+1}^{2N}, u_{1,e}^{2i-2} | u_{2i} \right), \end{aligned} \quad (2.25)$$

where  $a_{1,e}^j$  ( $a_{1,o}^j$ ) denote the subvector with even (odd) indices ( $a_k : 1 \leq k \leq j; k$  even (odd)). After getting the transition probability of each synthetic channel, the symmetric capacity  $I \left( W_N^{(i)} \right)$  can be computed according to (2.15). The channel polarization theorem is then given in [36] as follows:

**Theorem 2.2.2.** *For any B-DMC  $W$ , the channels  $\{W_N^{(i)}\}$  polarize in the sense that, for any fixed  $\delta \in (0, 1)$ , as  $N$  goes to infinity through powers of two, the fraction of indices  $i \in \{1, \dots, N\}$  for which  $I \left( W_N^{(i)} \right) \in (1 - \delta, 1]$  goes to  $I(W)$  and the fraction for which  $I \left( W_N^{(i)} \right) \in [0, \delta)$  goes to  $1 - I(W)$ .*

An example of the polarization effect for the case  $W$  is a binary erasure channel (BEC) with erasure probability  $\epsilon = 0.5$  is shown in Figure 2.5. A B-DMC  $W$  is called a BEC if for each  $y \in \mathcal{Y}$ , either  $W(y|0)W(y|1) = 0$  or  $W(y|0) = W(y|1)$ . In the latter case,  $y$  is said to be an erasure symbol. The sum of  $W(y|0)$  over all erasure symbols  $y$  is called the erasure probability  $\epsilon$  of the BEC. Note that  $I \left( W_N^{(i)} \right)$  can be calculated using the recursive relations which only valid for BECs.

$$\begin{aligned} I \left( W_1^{(1)} \right) &= 1 - \epsilon, \\ I \left( W_N^{(2i-1)} \right) &= I \left( W_{N/2}^{(i)} \right)^2, \\ I \left( W_N^{(2i)} \right) &= 2I \left( W_{N/2}^{(i)} \right) - I \left( W_{N/2}^{(i)} \right)^2. \end{aligned} \quad (2.26)$$

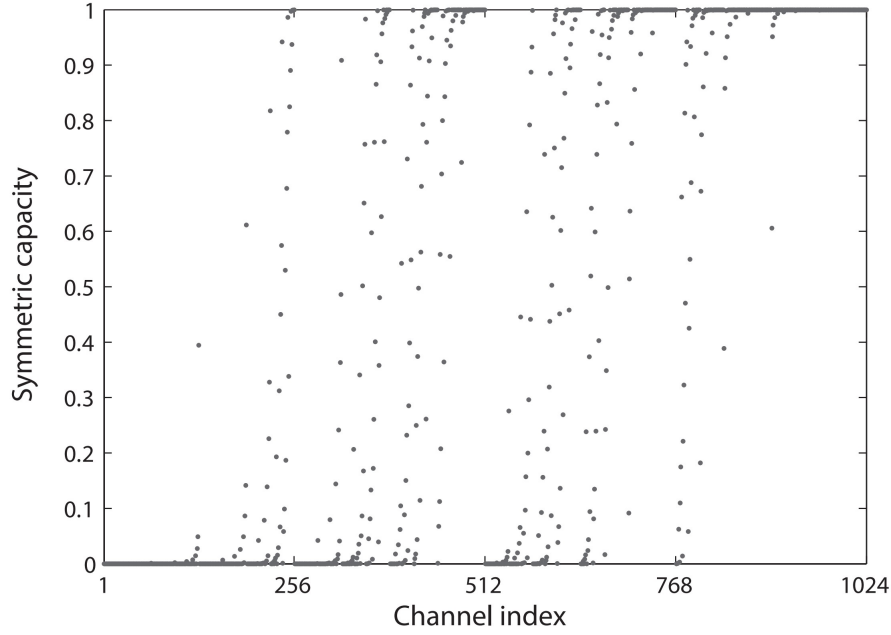


Figure 2.5 Plot of  $I(W_N^{(i)})$  versus  $i = 1, \dots, N = 2^{10}$  for a BEC with  $\epsilon = 0.5$ .

In Figure 2.5,  $I(W_N^{(i)})$  is close to 0 for small  $i$  and close to 1 for large  $i$ . When  $N \rightarrow \infty$ , the fraction of indices  $i$  with  $I(W_N^{(i)}) = 0$  goes to  $1 - I(W) = \epsilon = 0.5$  and the fraction of indices  $i$  with  $I(W_N^{(i)}) = 1$  goes to  $I(W) = 1 - \epsilon = 0.5$ . We can intuitively think of using synthetic channel  $i$  with  $I(W_N^{(i)})$  near 1 to transmit information data. This properly can be exploited to achieve reliable transmission. However, we should first find out how to determine the subset of indices  $i$  corresponding to synthetic channels with high reliability, which is an important computational problem whose solution will be given in the following subsection.

## 2.2.2 Polar codes construction

In [36], the Bhattacharyya parameter is used to estimate the reliability of polarization channel. Given a B-DMC  $W : \mathcal{X} \rightarrow \mathcal{Y}$ , the Bhattacharyya parameter is defined as

$$Z(W) = \sum_{y \in \mathcal{Y}} \sqrt{W(y|0)W(y|1)}. \quad (2.27)$$

It is an upper bound on the probability of maximum-likelihood (ML) decision error when  $W$  is used only once to transmit a 0 or 1. The relationship between  $Z(W)$  and  $I(W)$  is shown in the proposition below.



**Proposition 2.2.1.** *For any B-DMC  $W$ , we have*

$$\begin{aligned} I(W) &\geq \log \frac{2}{1+Z(W)}, \\ I(W) &\leq \sqrt{1-Z(W)^2}. \end{aligned} \quad (2.28)$$

It is easy to see that  $I(W) = 1$  iff  $Z(W) = 0$ , and  $I(W) = 0$  iff  $Z(W) = 1$ . The Bhattacharyya parameter of the synthetic channels satisfy

$$Z(W_{2N}^{(2i-1)}) \leq 2Z(W_N^{(i)}) - Z(W_N^{(i)})^2, \quad (2.29)$$

$$Z(W_{2N}^{(2i)}) = Z(W_N^{(i)})^2, \quad (2.30)$$

with equality in (2.29) iff  $W$  is a BEC. With the help of (2.29) and (2.30), we can derive the below theorem.

**Theorem 2.2.3.** *For any B-DMC  $W$  with  $I(W) > 0$ , and any fixed  $R < I(W)$ , there exists a sequence of sets  $\mathbb{A}_N \subset \{1, \dots, N\}$ ,  $N \in \{1, 2, \dots, 2^n, \dots\}$ , such that  $|\mathbb{A}_N| \geq NR$  and  $Z(W_N^{(i)}) \leq O(N^{-5/4})$  for all  $i \in \mathbb{A}_N$ .*

Theorem 4 inspires us to choose these synthetic channel with lower  $Z(W_N^{(i)})$  to convey information. This idea leads to the definition of polar codes.

**Polar Codes:** Given a B-DMC  $W$ ,  $u_1^N$  is the source block and it is encoded in the manner

$$x_1^N = u_1^N G_N, \quad (2.31)$$

where  $G_N$  is the generator matrix of order  $N$ , defined above.  $u_i, i \in \mathbb{A}$  carry information bits and the rest are set to any fixed values. The information set  $\mathbb{A}$  is chosen as a  $K$ -element subset of  $\{1, \dots, N\}$  such that  $Z(W_N^{(i)}) \leq Z(W_N^{(j)})$  for all  $i \in \mathbb{A}$ ,  $j \in \mathbb{A}^c$ . Then we can obtain a mapping from source blocks  $u_{\mathbb{A}}$  to codeword blocks  $x_1^N$ . This mapping is called a polar code for  $W$ .

The following theorem shows that the polar codes achieve the symmetric capacity of any given B-DMC  $W$  under successive cancellation decoding. The details of successive cancellation decoding will be introduced in the next subsection.

**Theorem 2.2.4.** *For any given B-DMC  $W$  and fixed  $R < I(W)$ , block error probability for polar coding under successive cancellation decoding satisfies*

$$P_e(N, R) = O\left(N^{-\frac{1}{4}}\right), \quad (2.32)$$

*this also means that polar codes are Shannon capacity achieving if  $W$  is symmetric.*

To get the information set  $\mathbb{A}$ , if B-DMC  $W$  is a BEC, the Bhattacharyya parameters of the polarized channels can be tracked by using the recursive calculation

in (2.29) and (2.30) with  $Z(W_1^{(1)}) = \epsilon$ . However, for channels other than BEC, the computational complexity grows exponentially with the code length and input alphabet size. To construct a polar code over an arbitrary symmetric B-DMC, Mori and Tanaka proposed the use of DE methods [77]. It is widely used in LDPC codes for tracing the PDF of LLRs at the variable and check nodes in the decoding graph, and is equally applicable to the polar codes design. Based on the calculated LLR PDFs, the error probabilities of all the polarized channels can be obtained. In practical implementation, the LLR PDFs should be quantized into  $q$  levels to keep an acceptable computational complexity. However, a typical value of  $q$  is  $10^5$ , thus causing a huge computational burden.

Tal and Vardy proposed an effective method to reduce computational complexity through appropriate approximation [78]. Two approximation methods, called upgrading and degrading quantization are introduced to transform the relevant channel into a new one with a smaller output alphabet. They give lower and upper bounds on  $Z(W_N^{(i)})$  of polarized channel. The two bounds are found to be very close through analysis and numerical simulation, thus can be used as approximations of  $Z(W_N^{(i)})$ . In this way, the construction complexity can be reduced dramatically compared to DE.

For binary input AWGN channels, an alternative method called GA can be applied in the construction of polar codes [79]. The GA has lower complexity than Tal and Vardy’s method but yields almost the same precision when applied to binary input AWGN channels. It is a more attractive choice than other methods since AWGN channel is typically considered by coding theorists.

### 2.2.3 Polar encoding and decoding

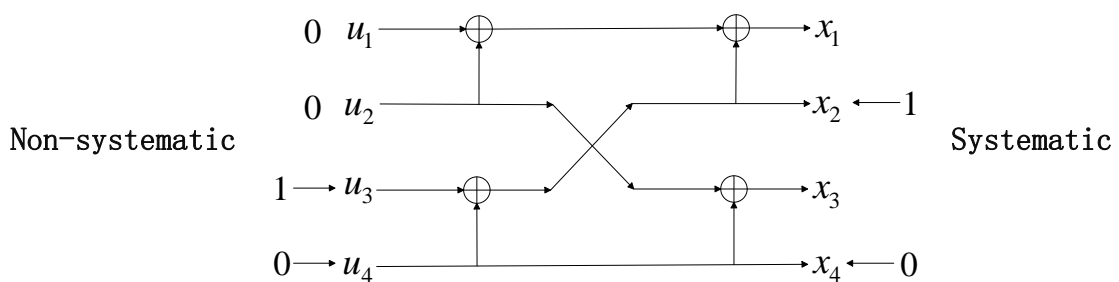


Figure 2.6 An example of polar encoding.

The encoding structure and explicit mathematical formulation of polar codes are shown in Figure 2.4 and (2.31), respectively. In the original polar coding, the non-systematic form is used. Given the code length  $N = 2^n, n \in \{1, 2, \dots\}$ , and information set  $\mathbb{A}, |\mathbb{A}| = K$ , the  $K$  information bits are assigned to  $u_i, i \in \mathbb{A}$  and  $u_i, i \in \mathbb{A}^c$  are set to fixed values known to both the sender and receiver in advance,

generally all zero. The non-systematic polar codes (NSMPC) gives a one-to-one mapping from  $u_i, i \in \mathbb{A}$  to  $x_1^N$ .

In systematic polar codes (SMPC), similar to non-systematic form,  $u_i, i \in \mathbb{A}^c$  are set to fixed values. The difference is the  $K$  information bits are assigned to  $x_i, i \in \mathbb{A}'$ , where  $\mathbb{A}'$  is the image of  $\mathbb{A}$  under the permutation represented by  $G_N$  in (2.21). It has been proved in [82] that there exists a one-to-one mapping from  $x_i, i \in \mathbb{A}'$  to  $x_1^N$  in the systematic form. SMPC shows the same block error rate (BLER) performance with that of NSMPC, but has an improvement in terms of bit error rate (BER) as shown in Figure 2.14. This surprising result is explained in [111]. It presented a detailed method to compute the partial distance spectrum of polar codes. The BLER and BER bounds are obtained by the partial distance spectrum. From the perspective of the union bounds, SMPC have advantages in BER performance but the same BLER performance with respect to NSMPC. An example of polar coding is given in Figure 2.6.  $N = 4$  and  $\mathbb{A} = \{3, 4\}$ ,  $u_1, u_2$  are set to 0. In non-systematic polar code, information bits are assigned to  $\{u_3, u_4\}$ , while in systematic polar code, information bits are assigned to  $\{x_2, x_4\}$ , so the input itself is included in the output codeword.

It has been shown in Theorem 2.2.4 that polar codes with successive cancellation (SC) decoding can achieve the symmetric capacity of B-DMCs. Let  $u_1^N$  be the input sequences to the polar encoder,  $x_1^N$  be the corresponding codeword and  $y_1^N$  be the channel observations. The SC decoding is done in a sequential manner. The estimation of bit  $u_i$  is based on the received vector  $y_1^N$  and estimations  $\hat{u}_1^{i-1}$  of the previous bits  $u_1^{i-1}$  as follows.

$$\hat{u}_i = \begin{cases} h_i(y_1^N, \hat{u}_1^{i-1}), & \text{if } i \in \mathbb{A}, \\ u_i, & \text{if } i \in \mathbb{A}^c, \end{cases} \quad (2.33)$$

where, when  $i \in \mathbb{A}^c$ , the value of the bit is known to both sides in advance, thus decision can be made directly as  $\hat{u}_i = u_i$ . When  $i \in \mathbb{A}$ , the bit conveys information, the decision function is

$$h_i(y_1^N, \hat{u}_1^{i-1}) = \begin{cases} 0, & \text{if } \ln \frac{W_N^{(i)}(y_1^N, \hat{u}_1^{i-1}|0)}{W_N^{(i)}(y_1^N, \hat{u}_1^{i-1}|1)} \geq 0, \\ 1, & \text{otherwise.} \end{cases} \quad (2.34)$$

The form of LLR is adopted in (2.34). It was identified in [112] that using the LLR values results in a decoder which is more area-efficient than the conventional one with Log-Likelihood (LL) values.

The recursive calculation of LLRs is performed by the basic decoder unit as shown below.

$L_1, L_2$  are two input LLRs. When calculating  $L_{out1}$ , the  $f$  function over the LLR domain is executed as

$$L_{out1} = f(L_1, L_2), \quad (2.35)$$

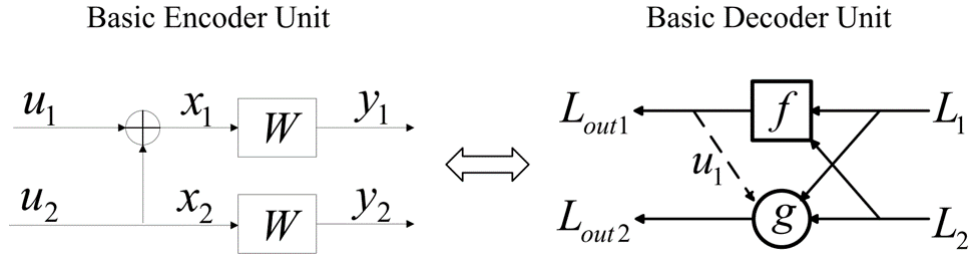


Figure 2.7 Basic Decoder Unit.

and when calculating  $L_{out2}$ , the  $g$  function over the LLR domain is executed as

$$L_{out2} = g(L_1, L_2, U_1), \quad (2.36)$$

where

$$f(x, y) = 2 \operatorname{arctanh} \left( \tanh \left( \frac{x}{2} \right) \tanh \left( \frac{y}{2} \right) \right), \quad (2.37)$$

$$g(x, y, u) = (-1)^u x + y. \quad (2.38)$$

The  $f$  function can be approximated as [113]

$$f(x, y) = \operatorname{sign}(x) \operatorname{sign}(y) \min(|x|, |y|). \quad (2.39)$$

After calculating  $L_N^{(i)} = \ln \left( \frac{W_N^{(i)}(y_1^N, \hat{u}_1^{i-1} | 0)}{W_N^{(i)}(y_1^N, \hat{u}_1^{i-1} | 1)} \right)$ , the estimation of bit  $u_i$  is obtained following (2.33). The estimation is propagated back following the encoder structure and enable the LLR calculation in (2.38) for the rest undecoded bits.

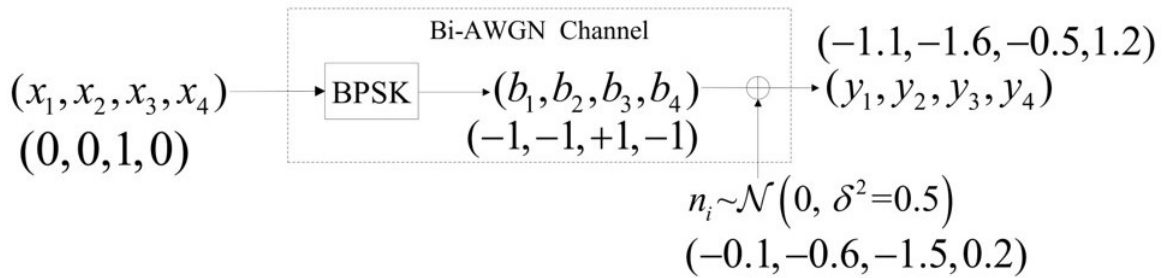


Figure 2.8 Codeword transmission over Bi-AWGN channel.

An example of polar codes with  $N = 4$  and the number of information bits  $K = 2$  is given for a better understanding of polar encoding and SC decoding.  $\mathcal{A} = \{3, 4\}$ , information bit  $u_3 = 1$ ,  $u_4 = 0$ , frozen bit  $u_1 = u_2 = 0$ . The codeword  $x_1^4$  can be obtained by (2.31). As shown in Figure 2.8, after BPSK modulation, the signal goes over a binary-input AWGN (Bi-AWGN) channel with a noise variance  $\sigma^2 = 0.5$ . The received signal at the decoder is  $(-1.1, -1.6, -0.5, 1.2)$ . For an AWGN channel, the

LLRs input into the decoder can be calculated through

$$L_i = \ln \left( \frac{W(y_i|x_i=0)}{W(y_i|x_i=1)} \right) = \frac{2y_i}{\sigma^2}, \quad 1 \leq i \leq N. \quad (2.40)$$

Hence,  $(L_1, L_2, L_3, L_4) = (-4.4, -6.4, -2.0, 4.8)$ . Figure 2.9 to 2.12 illustrate the SC decoding of  $u_1$  to  $u_4$  in detail. Solid and dashed lines indicate the propagation of LLRs and estimations, respectively. The channel side LLRs first pass two-stage  $f$  function to get  $L_{out1}$ . Since  $u_1$  is frozen bit, its estimation is set to 0 directly.

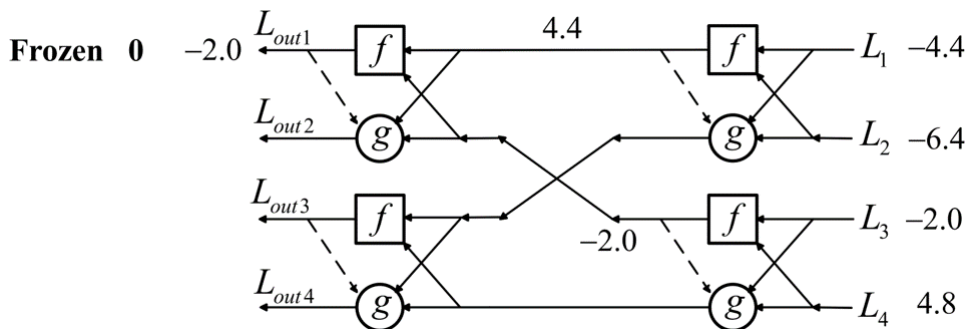


Figure 2.9 SC Decoding of  $u_1$ .

Then, 0 is propagated back and  $g$  function is executed to calculate  $L_{out2}$ . The estimation of  $u_2$  is set to 0 as it is also a frozen bit.

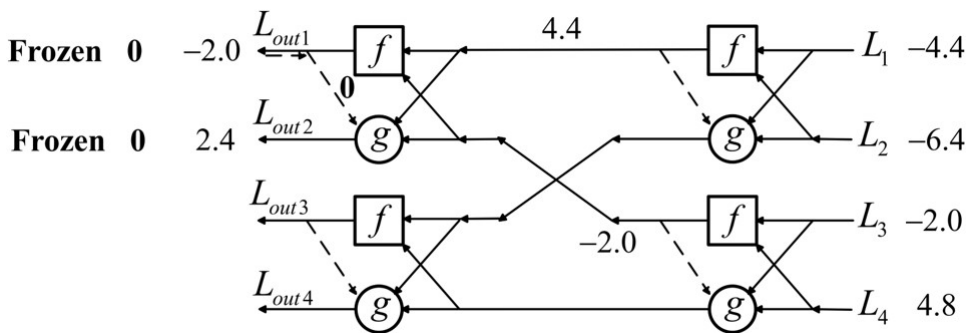


Figure 2.10 SC Decoding of  $u_2$ .

Next, the decoding of  $u_3$ . The estimations of  $u_1$  and  $u_2$  are propagated back to the first stage and two  $g$  functions are executed. The results goes into a  $f$  function and  $L_{out3}$  is output.  $u_3$  is a information bit so its estimation can be got using (2.34), that is 1.

Finally, the decoding of  $u_4$ .  $L_{out4}$  can be calculated by a  $g$  function using the estimation of  $u_3$ .  $u_4$  is a information bit so its estimation can be got using (2.34), that is 0. Thus, the decoding output of the SC decoder is  $\{0, 0, 1, 0\}$ , which is consistent with the information bits.

It can be observed that the hard decision result of  $u_1$  is in fact 1 but it is still set to the correct value 0. That's why these bits corresponding to polarized channels

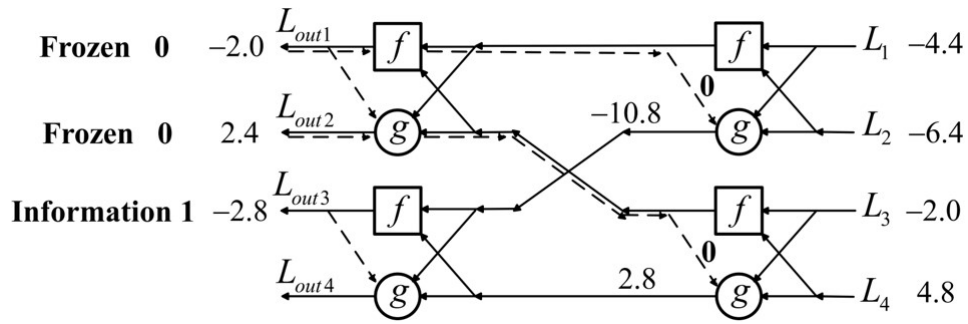


Figure 2.11 SC Decoding of  $u_3$ .

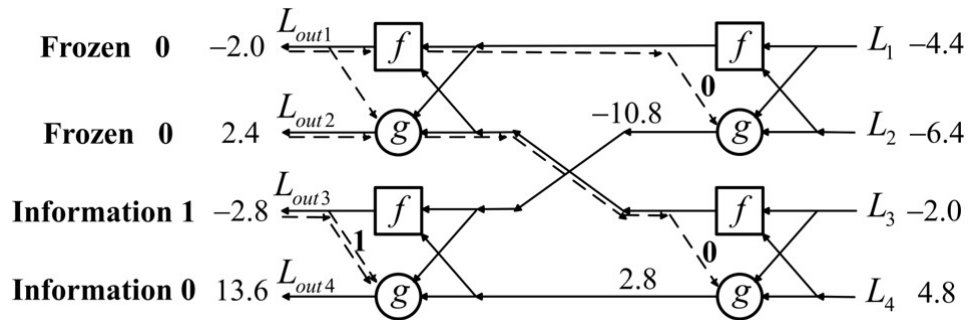


Figure 2.12 SC Decoding of  $u_4$ .

with low reliability are frozen. In this way, their estimations are guaranteed to be correct and thus help to improve the reliability of decoding of the information bits by providing extra information.

The SC decoding process above can also be viewed as the search of a code tree in Figure 2.13.

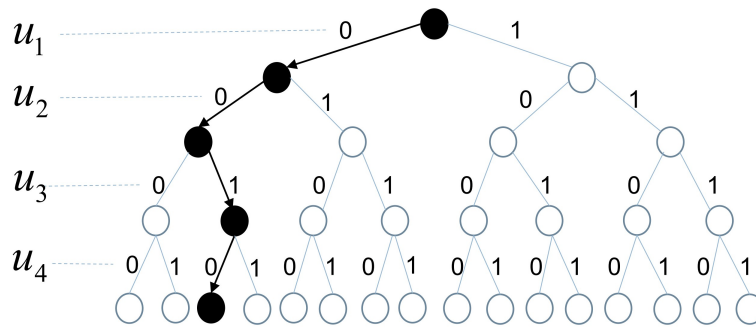


Figure 2.13 Code Tree.

We can find SC decoding makes decision at every bit and keeps only one searching path. This inspires us to think about the use of more than one searching paths, which may improve the probability of finding the right codeword. The successive cancellation list (SCL) decoding has been proposed in [84] for achieving this goal. Different from SC decoding, it does not make decision directly at every bits, instead, it generates two new paths by extending the original path by adding a 0 or 1 when

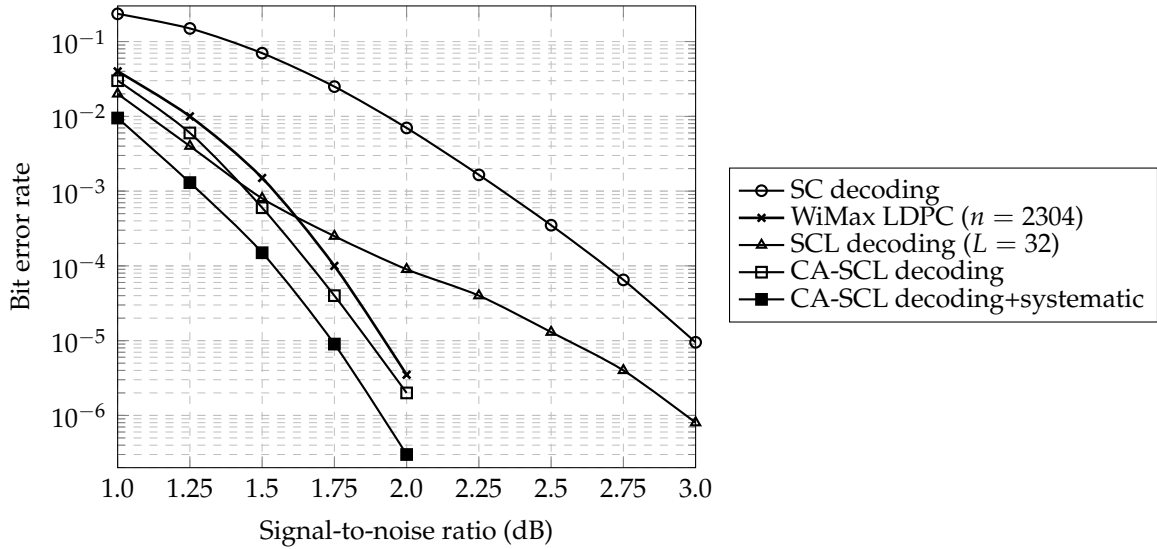


Figure 2.14 Comparison of conventional SC, SCL, CA-SCL decoders to an implementation of the WiMax standard take from [114]. All codes are rate 1/2. The length of the polar code is 2048 while the length of the WiMax code is 2304. The list size is  $L = 32$ . The CRC is 16 bits long.

encountering an information bit, and extends the original path by adding a 0 when encountering a frozen bit. Then, it calculates the path metric of all generated paths. Path metric is defined as

$$PM_i = \ln \left( P \left( \hat{u}_1^i | y_1^N \right) \right). \quad (2.41)$$

Its update function is

$$PM_i[\ell] = PM_{i-1}[\ell] - \ln \left( 1 + e^{-(1-2\hat{u}_i[\ell]) \cdot L_N^{(i)}[\ell]} \right). \quad (2.42)$$

$PM_i[\ell]$  represents the metric of path  $\ell$  with length  $i$ .  $\hat{u}_i[\ell]$  represents the estimated value of the  $i$ -th bit of path  $\ell$ .  $L_N^{(i)}[\ell]$  represents  $L_N^{(i)}$  of path  $\ell$ . At most  $L$  paths with the best path metrics are retained at each decoding length. The new path generation and selection operations are executed every time the decoding length is increased by 1. When the decoding length reaches  $N$ , the path with the best path metric will be selected from the  $L$  candidates as the decoding output. Searching  $L$  decoding paths in parallel brings SCL decoding improvement in error-correction performance with respect to SC decoding. It can be seen from Figure 2.14 that the performance of SCL decoding outperforms that of SC decoding significantly.

It is also observed in [84] that in most decoding failure cases, the correct path is amongst the final  $L$  candidates but does not have the best path metric. This means that performance could be further improved if we had a genie aided decoder capable of telling us which path to pick from the list at the final stage. Such a genie can be easily implemented, for example using a cyclic redundancy check (CRC) precoding.

SCL decoding utilizing CRC bits is called CRC-aided SCL (CA-SCL) decoding [85]. In encoding, for the  $K$  information bits that are free to set, instead of setting all of them to useful information, a simple concatenation scheme is employed. For a constant  $r < K$ , the first  $K - r$  bits are set to information and the last  $r$  bits will hold the  $r$ -bit CRC value of the first  $K - r$  bits. In decoding, when the decoding length reaches  $N$ , CRC will be performed on all  $L$  paths. If there exist paths that pass CRC, the path that both passes CRC and has the best path metric will be output. If no path passes CRC, the path with the best path metric will be output. CA-SCL decoding incurs a penalty in rate, since the rate is now  $\frac{K-r}{N}$  instead of the previous  $\frac{K}{N}$ . But a significant performance improvement is gained compared to SCL decoding as shown in Figure 2.14. What's more, the performance of polar codes under CA-SCL decoding is comparable to state-of-the-art LDPC codes.

In terms of computational complexity, in SC decoding,  $(N/2 \cdot \log_2 N)$   $f$  function and  $(N/2 \cdot \log_2 N)$   $g$  function operations need to be executed. Hence, the total computational complexity required by the SC decoder is  $O(N \log_2 N)$ . As a result of searching  $L$  paths in parallel, the computational complexity of the SCL decoder is  $O(L \cdot N \log_2 N)$ . The high complexity of SCL decoder makes achieving high error-correction performance with low decoding complexity an attractive research direction.

Successive cancellation stack (SCS) [86] decoding is one of the candidate solutions. Similar to SCL decoding, it also follows multiple paths on the decoding tree. However, instead of expanding  $L$  paths simultaneously, only the path with the best path metric is expanded at each step. The error-correction performance of SCS decoding is almost the same with that of SCL decoding, but with much lower average computational complexity. It's worth noting that with the increasing of signal-to-noise ratio, the decoding complexity of SCL decoding remains constant, while, that of SCS decoding falls down quickly and tends to approach that of SC decoding. As a trade for the low complexity, SCS decoding requires a larger memory and a longer runtime than SCL decoding.

### Fast simplified successive-cancellation decoding

The SC decoding has strong data dependencies that limit the amount of parallelism that can be exploited within the algorithm because the estimation of each bit depends on the estimation of all previous bits. This leads to a large latency in the SC decoding algorithm.

By grouping all the operations that can be performed in parallel, SC decoding can be represented as the traversal of a binary tree. Figure 2.15 shows the example of SC decoding of polar codes with  $N = 4$  and  $\mathcal{A} = \{3, 4\}$ . Each node in the tree corresponds to a constituent code. For any node  $v$ ,  $v^p$  is its parent node.  $v^l, v^r$  are its left and right child nodes, respectively.  $\alpha$  denotes the LLRs input into the node and  $\beta$  denotes the estimate output from the node. The traversal starts from the top node fed by the LLRs received from the channel and follows a depth-first principle with



priority to the left. For a node  $v$  at stage  $t$ , the LLR values passed to its child node are calculated as

$$\alpha^l[k] = f(\alpha[2k-1], \alpha[2k]), \quad 1 \leq k \leq 2^{t-1}, \quad (2.43)$$

$$\alpha^r[k] = g(\alpha[2k-1], \alpha[2k], \beta^l[k]), \quad 1 \leq k \leq 2^{t-1}, \quad (2.44)$$

where  $f$  and  $g$  functions are defined in (2.39) and (2.38), respectively.

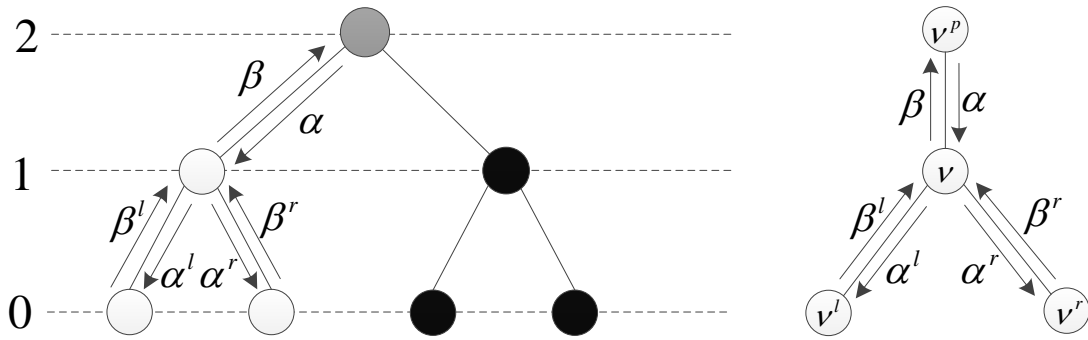


Figure 2.15 Binary tree representation of a SC decoder for a polar code with  $N = 4$  and  $\mathbb{A} = \{3, 4\}$ .

When the LLR value of the  $k$ -th leaf node at stage zero is calculated, the estimation of the  $k$ -th bit can be obtained according to (2.33). The hard-valued messages are propagated back to the parent node as

$$\hat{\beta}[k] = \begin{cases} \hat{\beta}^l \left[ \frac{k+1}{2} \right] \oplus \hat{\beta}^r \left[ \frac{k+1}{2} \right], & \text{if } \text{mod}(k, 2) = 1, \\ \hat{\beta}^r \left[ \frac{k}{2} \right], & \text{if } \text{mod}(k, 2) = 0. \end{cases} \quad (2.45)$$

After traversing all the nodes in the tree, the SC decoding finishes and the estimates at leaf nodes are output as the decoding result.

It was first shown in [88, 115] that multi-bit decoding can be performed directly in nodes at intermediate level instead of bit-by-bit sequential decoding at level 0. In this way, fewer nodes in the SC decoding tree will be traversed and consequently, the latency caused by data computation and exchange will be reduced. The multi-bit decoding is performed only when specific special node types are encountered. The sequences of information and frozen bits of these node types have special bit patterns. Therefore, they can be decoded more efficiently without the need for exhaustive search. This class of fast decoding employing special node is named fast simplified SC decoding algorithm. To distinguish between frozen and information bits, a vector of flags  $\mathbf{d} = (d[1], d[2], \dots, d[N])$  is used where each flag  $d[k]$  is

assigned as

$$d[k] = \begin{cases} 0, & \text{if } k \in \mathbb{A}^c, \\ 1, & \text{otherwise.} \end{cases} \quad (2.46)$$

Using the vector  $\mathbf{d}$ , the five special node types proposed in [88] are described as:

- Rate-0 node: all bits are frozen bits,  $\mathbf{d} = (0, 0, \dots, 0)$ .
- Rate-1 node: all bits are non-frozen bits,  $\mathbf{d} = (1, 1, \dots, 1)$ .
- REP node: all bits are frozen bits except the last one,  $\mathbf{d} = (0, \dots, 0, 1)$ .
- SPC node: all bits are non-frozen bits except the first one,  $\mathbf{d} = (0, 1, \dots, 1)$ .
- ML node: a code of length 4 with  $\mathbf{d} = (0, 1, 0, 1)$ .

By merging the aforementioned special nodes, additional nodes were proposed, namely, the REP-SPC node, which is a REP node followed by a SPC node and the P-01/P-0SPC node, which is generated by merging a Rate-0 node with a Rate-1/SPC node. In [89], five additional special node types and their corresponding fast decoders were introduced. This enhanced fast SC decoding algorithm can achieve a lower decoding latency than that in [88]. The five special node types are:

- Type-I node: all bits are frozen bits except the last two,  $\mathbf{d} = (0, \dots, 0, 1, 1)$ .
- Type-II node: all bits are frozen bits except the last three,  $\mathbf{d} = (0, \dots, 0, 1, 1, 1)$ .
- Type-III node: all bits are non-frozen bits except the first two,  $\mathbf{d} = (0, 0, 1, \dots, 1)$ .
- Type-IV node: all bits are non-frozen bits except the first three,  $\mathbf{d} = (0, 0, 0, 1, \dots, 1)$ .
- Type-V node: all bits are frozen bits except the last three and the fifth to last,  $\mathbf{d} = (0, \dots, 0, 1, 0, 1, 1, 1)$ .

A generalized fast SC (GFSC) decoding algorithm was proposed in [90] by introducing the G-PC node and the G-REP node. The G-PC node is a node at level  $j$  having all its descendants as Rate-1 nodes except the leftmost one at a certain level  $r < j$ , that is a Rate-0 node. The G-REP node is a node at level  $j$  for which all its descendants are Rate-0 nodes, except the rightmost one at a certain level  $r < j$ , which is a generic node of rate  $C$  (Rate- $C$ ). In [91], in addition to the nodes in [88], three special node types are added: the REP1 node, with REP node on the left and Rate-1 node on the right; the 0REPSPC node, which is a concatenation of Rate-0, REP, and SPC nodes; and the 001 node, whose left 3/4 of bits are frozen bits and right 1/4 of bits are unfrozen bits. Besides, REP-REPSPC, REP-Rate1, and Rate0-ML nodes are adopted as the merging of special nodes in [92].

Except for special nodes and their merger, node-branch mergers and branch operation mergers were also exploited to further reduce the latency. Branch operations refer to (2.43), (2.44) and (2.45). Node-branch mergers include the P-R1/P-RSPC

node [88], which is generated by merging a  $g$  function branch operation in (2.44) with the following Rate-1/SPC node; and the F-REP node [93], which is generated by performing an  $f$  function operation followed by a REP node. In [92] and [93], in addition to merging special nodes, the following branch operation mergers are further introduced:

- $F^{\times 2}$ : two consecutive  $f$  function operations in (2.43).
- $G0^{\times 2}$ : two consecutive  $G0$  operations, where  $G0$  is the  $g$  function in (2.44) assuming hard-valued messages of all zeros.
- $C^{\times 2}, C^{\times 3}$ : up to three consecutive operations in (2.45).
- $C0^{\times 2}, C0^{\times 3}$ : up to three consecutive operations in (2.45), assuming the estimations from the left branch are all zeros.
- G-F:  $g$  function operation followed by an  $f$  function operation.
- F-G0:  $f$  function operation followed by a  $G0$  operation.

The key advantage of using specific parallel decoders for the aforementioned special nodes is that, since the SC decoding tree is not traversed when one of these nodes is encountered, significant latency saving can be achieved. For example, if node  $v$  is a Rate-0 node, the estimates are all 0 as  $v$  corresponds to all frozen bits. If node  $v$  is a Rate-1 node at level  $t$ , hard decision decoding can be used to immediately obtain the decoding result as

$$\beta_i = h(\alpha_i) = \begin{cases} 0, & \text{if } \alpha_i \geq 0, \\ 1, & \text{otherwise,} \end{cases} \quad 1 \leq i \leq 2^t. \quad (2.47)$$

if node  $v$  is a REP node, the bit estimates can be found as

$$\beta_i = \frac{1}{2} \left( 1 - h \left( \sum_{k=1}^{2^t} a_k \right) \right), \quad 1 \leq i \leq 2^t. \quad (2.48)$$

If node  $v$  is a SPC node at level  $t$ , hard decision based on (2.47) is first derived followed by the calculation of the parity of the output using modulo-2 addition. The index of the least reliable bit is found as

$$i' = \arg \min_i |\alpha_i|, \quad 1 \leq i \leq 2^t. \quad (2.49)$$

Finally, the bits in a SPC node are estimated as

$$\beta_i = \begin{cases} h(\alpha_i) \oplus \text{parity}, & \text{if } i = i', \\ h(\alpha_i), & \text{otherwise.} \end{cases} \quad (2.50)$$

Likewise, simplified decoding algorithms can also be found for other special node types. All of the aforementioned fast SC decoding algorithms perform parallel decoding at an intermediate level of the decoding tree in order to reduce the number of traversed nodes. An efficient decoding algorithm that can decode a node at a higher level of the decoding tree generally results in more savings in terms of latency than the one that decodes a node at a lower level of the decoding tree.

## 2.3 Summary

This chapter started with a brief introduction of channel coding and some related concepts and definitions. On this basis, the theory of polar codes, especially the proof of its capacity-achieving property, was provided. The code construction, encoding and decoding of polar codes were then reviewed, especially different types of decoders, which lays the foundations for the theoretical innovations in Chapter 4, Chapter 5 and Chapter 6.



## CHANNEL CODING REQUIREMENTS FOR OPTICAL WIRELESS COMMUNICATIONS

To find out the channel coding requirements of OWC system, this chapter first gives an in-depth analysis of a typical OWC system. To be more specific, it will focus on the OWC system using pencil beams, which is employed in the BROWSE project to implement an indoor communication network [19] as shown in Figure 3.1.

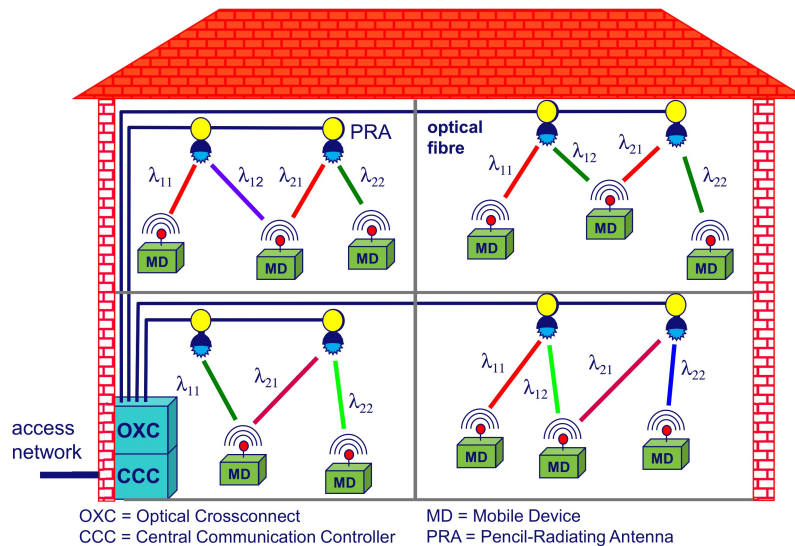


Figure 3.1 Hybrid OWC/radio network using wavelength-controlled 2D infrared steered beams from [9].

This network architecture utilizes an optical crossconnect (OXC) to route the optical signals to the rooms through an indoor fiber backbone network. A central communication controller (CCC) provides the intelligent governance of this process. Each room has several pencil-radiating antennas (PRAs), which receive the signal from the fiber and steer multiple beams two-dimensionally in directions determined by their wavelengths. Using wavelength tuning, a PRA can transmit signal to any mobile device (MD) within its coverage through an unshared link. Note that the beam wavelength, the PRA position and the room number corresponding to a MD are all taken care of by the CCC according to the MD's location information so the PRA is fully passive, in other words, without any electrical powering or electrical control signals. The PRA, the free-space channel and the receiver at the MD constitute the OWC system using pencil beams.

### 3.1 Typical optical wireless communication system

The system structure is shown in Figure 3.2. A laser diode is employed to generate an optical carrier intensity modulated by a Mach Zehnder modulator (MZM) which is driven by a Pseudo-Random Binary Sequence (PRBS) generator of NRZ-OOK data. The modulated optical signal goes through a variable optical attenuator (VOA) followed by an Erbium-Doped Fiber Amplifier (EDFA).

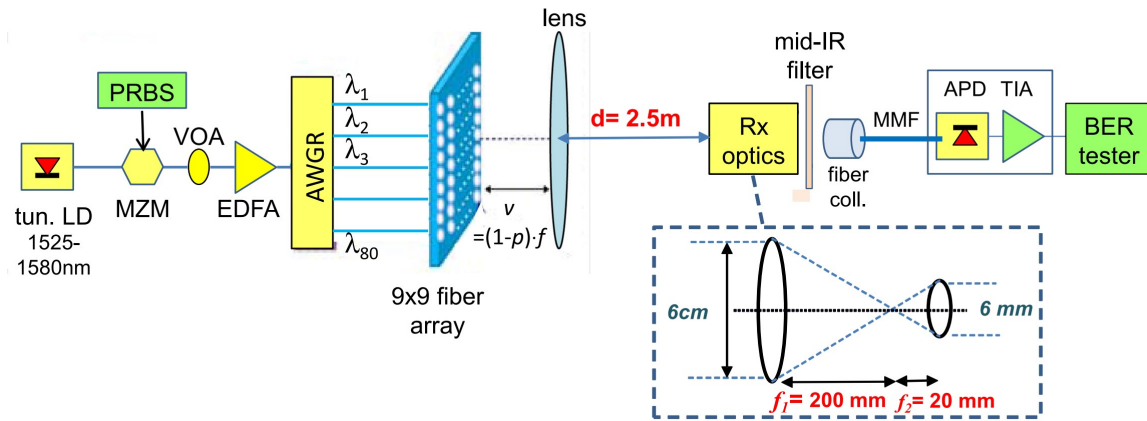


Figure 3.2 BS-ILC system from [9].

Then the signal will enter the beam steering module which includes an AWGR whose  $N \cdot M$  output fibers are arranged in an  $N \times M$  2D fiber array, which is put in the object plane of a lens. The position of a fiber in the object plane determines in which 2D direction its corresponding beam is emitted after the lens [9]. The size of the beam steering module can be reduced significantly by a defocusing approach: when putting the fiber array out of the focus and closer to the lens, the emitted beams are slightly diverging and the spot diameter increases, leading to a larger coverage area. Conversely, for a given coverage area, defocusing results in a significantly smaller focal length and diameter of the lens, and smaller spacing between the fibers in the array, and thus to a significant size reduction of the beam steering module. Applying defocusing technology, the object distance between the array and the lens  $v$  is defined by  $v = (1 - p) \cdot f$ , where  $p$ ,  $0 \leq p < 1$  is the relative defocusing parameter and  $f$  is the focal length.

At the receiver side, a telescopic lens system is used for beam narrowing, followed by a mid-IR filter which suppresses ambient light and passes the IR beam. Subsequently, a fiber collimator feeds the IR beam to an avalanche photodetector (APD) followed by a transimpedance amplifier (TIA) via a  $50 \mu\text{m}$  core multimode fiber (MMF). The APD-TIA converts the optical signal to electrical current signal for further signal processing.

### 3.2 Channel properties

Intensity modulation with direct detection (IM/DD) is adopted in BS-ILC system, which is a viable and practical way of implementing optical wireless systems due to its low cost and complexity. In an infrared link with IM/DD, the transceiver modulates the desired waveform onto the instantaneous power of the optical carrier. The photodetector at the receiver produces a current proportional to the integral of the received instantaneous optical power over the entire photodetector surface.

An IM/DD-based optical wireless system has an equivalent baseband model as shown in Figure 3.3.

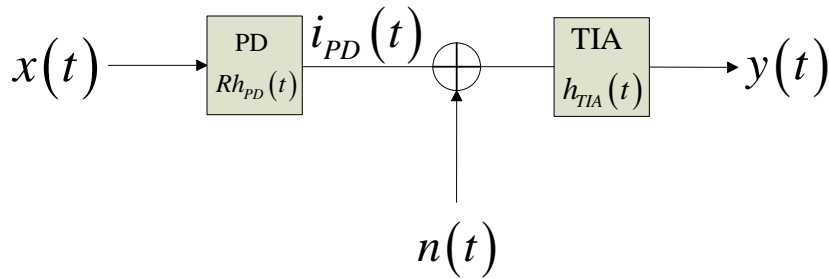


Figure 3.3 Equivalent baseband model of an infrared wireless system using IM/DD. (PD=photodiode, TIA=transimpedance amplifier)

$R$  is the photodetector responsivity.  $h_{PD}(t)$  and  $h_{TIA}(t)$  are the impulse responses of photodiode and transimpedance amplifier, respectively.  $i_{PD}(t)$  is the photocurrent.  $n(t)$  is the signal-independent additive noise.  $y(t)$  is the receiver's output signal. The impulse response  $h(t)$  is the time evolution of the received signal when an infinitely short pulse is sent, which allows precalculating the system's output for any arbitrary input in the time domain. Mathematically, the baseband channel model is summarized as

$$\begin{aligned}
 y(t) &= (i_{PD}(t) + n(t)) \otimes h_{TIA}(t) \\
 &= [Rx(t) \otimes h_{PD}(t) + n(t)] \otimes h_{TIA}(t) \\
 &= Rx(t) \otimes h_{PD}(t) \otimes h_{TIA}(t) + n(t) \otimes h_{TIA}(t),
 \end{aligned}
 \tag{3.1}$$

where the  $\otimes$  symbol denotes convolution. While (3.1) is simply a conventional linear filter channel with additive noise, the infrared link is different from that of electrical or radio in two aspects. First,  $x(t)$  represents the optical power rather than the amplitude signal, which means it must be nonnegative, that is

$$x(t) \geq 0. \tag{3.2}$$



Secondly, the average optical transmit power  $P_t$  is given by

$$P_t = \lim_{T \rightarrow \infty} \frac{1}{2T} \int_{-T}^T x(t) dt. \quad (3.3)$$

This is in contrast to the time-averaged value of the signal  $|x(t)|^2$ , which is the case for the conventional RF channels when  $x(t)$  represents amplitude. The optical transmit power  $P_t$  is limited by the eye safety requirement and must not exceed 10mW for  $\lambda > 1400\text{nm}$ , according to IEC 60825, ANSI Z136 standards [116].

The average received  $y(t)$  is given by

$$\overline{y(t)} = \lim_{T \rightarrow \infty} \frac{1}{2T} \int_{-T}^T y(t) dt = RP_t H_{PD-TIA}(0), \quad (3.4)$$

where  $H_{PD-TIA}(0) = \int_{-\infty}^{\infty} h_{PD}(t) \otimes h_{TIA}(t) dt$ .

In indoor scenarios, infrared links are operated in the presence of infrared and visible background light. While received background light can be minimized by optical filtering, it still adds shot noise, which can be modeled as white, Gaussian [117], and independent of  $x(t)$ . When little or no ambient light is present, the dominant noise source is receiver preamplifier noise, which is signal-independent and Gaussian. Thus  $n(t)$  is usually modeled as a signal-independent additive white Gaussian noise (AWGN).

Based on above analysis, the electrical SNR at the receiver can be expressed as

$$SNR = \frac{\left(\overline{y(t)}\right)^2}{R_b N_0} = \frac{R^2 P_t^2 H_{PD-TIA}(0)^2}{R_b N_0}. \quad (3.5)$$

Here,  $N_0$ ,  $R_b$  denote the noise spectral density (W/Hz) and the achievable bit rate (bps), respectively.

### 3.3 Requirements on channel coding

Considering the characteristics of the OWC system and its channel properties, channel codes employed in OWC system need to meet the following three requirements.

**Low latency:** Low decoding latency leads to high throughput. On the other hand, the indoor short-reach OWC system is latency-constrained inherently [118]. In the free-space channel, the light travels faster than through fiber (3.3ns/m in air, versus 5ns/m in silica fiber) and the reach of the link is only several meters commonly in indoor application scenarios. Therefore, the transmission delay of an indoor OWC link is small. To have a comparable signal processing speed, the encoding/decoding latency of channel encoder/decoder should be low. Due to the two reasons above, the design of decoder with low decoding latency is an urgent need of the OWC system.

**High error-correction performance:** OWC system is latency-constrained which means channel codes with short and moderate length are more favored for their relatively low decoding latency. However, shorter code length usually means worse error-correction performance. In polar codes, short code length will lead to insufficient number of polarizing transformation, thus the error-correction potential of polar codes can not be fully utilized. Therefore, how to improve the error-correction performance of channel codes with short and moderate length to meet the application requirement of OWC system is a big challenge.

**Low complexity:** For an indoor OWC system which serves mobile devices, the battery lifetime of mobile devices is an issue that has to be considered. As one of the most energy consuming components in the receiver, reducing the energy demand of decoder of channel codes is an effective way to prolong the duration of battery life. Since the energy consumption is generally proportional to the computational complexity, the solution turns to the reduction of computational complexity of decoder embedded in the mobile receiver. Therefore, the OWC system requires that the decoder can achieve the targeted decoding latency and error-correction performance with as low computational complexity as possible.



# **COMPLEXITY-ADJUSTABLE SC DECODING OF POLAR CODES**

---

This chapter is adapted from: H. Zheng, B. Chen, L. F. Abanto-Leon, Z. Cao and T. Koonen, "Complexity-adjustable SC decoding of polar codes for energy consumption reduction", *IET Communications*, 2019, 13(14): 2088-2096.

Aimed at the challenge 1 of low complexity defined in Section 1.3.2, this chapter proposes an enhanced list-aided successive cancellation stack (ELSCS) decoding algorithm with adjustable decoding complexity. In addition, a logarithmic likelihood ratio (LLR)-threshold based path extension scheme is designed to further reduce the memory consumption of stack decoding. Numerical simulation results show that without affecting the error correction performance, the proposed ELSCS decoding algorithm provides a flexible tradeoff between time complexity and computational complexity, while reducing storage space up to 70% with respect to conventional SCS decoding. Based on the fact that most mobile devices operate in environments with stringent energy budget to support diverse applications, the proposed scheme is a promising candidate for meeting requirements of different applications while maintaining a low computational complexity and computing resource utilization.

## **4.1 Introduction**

With the ever-increasing amount of wireless communication data, power consumption becomes a tremendous challenge for mobile communication devices, especially for those operating under very stringent energy budget. To obtain high energy efficiency, the high energy-consuming error-correction code (ECC) decoder is a pivotal component in the communications chain that needs to be suitably designed. However, different applications have diverse set of communication requirements, an one-size-fits-all energy-efficiency decoder is unfeasible. Therefore, a flexible decoder that can adjust the tradeoff between the conflicting demands for energy, throughput and error rate performance is essential for achieving low energy expenditure in different applications.

Considering their explicit decoding construction and low decoding complexity, polar codes are strong candidates to fulfill such requirements [36]. The most widely used high-performance decoding algorithm for polar codes is CRC-aided successive cancellation list (CA-SCL) decoding [84, 112]. However, CA-SCL still requires a large list size to achieve a good error rate performance, which inevitably leads to

excessive computational complexity. Another drawback is that the complexity of CA-SCL remains constant in different signal-to-noise ratio (SNR) regimes when in fact fewer computations are required for correct decoding in the high SNR regime, thus causing unnecessary energy expenditure.

Previous works have studied the simplified calculation of successive cancellation (SC)/successive cancellation list (SCL) decoding [115, 119, 88, 120, 121]. They found removable and replaceable redundant calculations when encountering frozen bit and particular information patterns, while keeping the error rate performance unaltered. With a similar principle, a new class of relaxed polar codes was introduced by deactivating polarization units when the bit-channels are sufficiently good or bad [122]. Irregular polar codes generalized this idea to further consider all possible selections of the inactivated polarization units and obtained a significant decoding complexity reduction [123].

In addition to simplification of the decoding process, path pruning is also an efficient way to reduce the computational complexity of SCL decoding. Chen et al. proposed to set up a flexible path metric threshold to prune decoding paths with small path metric [124, 125]. In [126], the concept of relative path metric (RPM) was introduced and the correct candidate was found to have a low probability of having a large RPM value. Paths which do not satisfy this property are pruned and the computational complexity can thereby be effectively decreased.

Adaptive SCL algorithm was introduced in [127]. It sets the initial value of the list size to 1. If the decoding attempt succeeds, the decoding process stops and the decoding result is output. Otherwise, the list size is doubled and the SCL decoding algorithm is restarted. This procedure is repeated until the list size reaches a predefined maximum threshold. If the decoding attempt still fails, a decoding failure is declared. This tentative early-stopped strategy enables adaptive SCL algorithm to have SNR-adaptive capability. Furthermore, this kind of strategy is also applicable to other decoding algorithms. As long as the algorithm itself has a lower computational complexity than SCL, the resultant decoding complexity of the new hybrid using this strategy will also be lower.

In fact, when decoding latency is not so stringent and throughput demand is not very high, the successive cancellation stack (SCS) decoding algorithm [86] is a more energy-efficient choice than SCL. By means of trading storage complexity for computational complexity reduction, SCS has a much lower and SNR-adaptive computational complexity. It was employed inside the kernel processor to reduce the decoding complexity of polar codes with Reed-Solomon kernel substantially [128]. In addition, the efficient software implementation of the SCS decoding algorithm has been investigated [129]. Moreover, the computational complexity reduction schemes mentioned above are equally applicable to SCS, can further decrease the required amount of computations. However, lower energy consumption leads to the disadvantage of smaller throughput. Because of employing breadth-first instead of depth-first search, the decoding delay of SCS will be larger than that of SCL with the same error rate performance, especially in the low SNR regime.

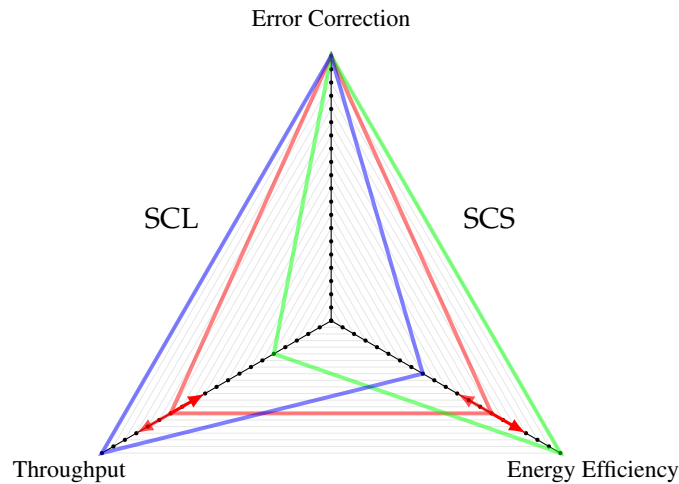


Figure 4.1 Comparison of performance metrics for SCL and SCS decoding algorithms. The adjustable performance is highlighted with red lines.

From the discussions above, we can observe that SCL and SCS have complementary properties. This inspired us to combine the advantages of both and propose an adjustable decoder which can make flexible tradeoff between energy consumption and throughput for a certain given error rate performance as shown in Figure 4.1. The main contributions of this chapter are described in the following. First, a logarithmic likelihood ratio (LLR)-threshold based path extension method is designed to reduce the memory requirements for stack decoding. Second, a new list-aided successive cancellation stack (LSCS) decoding algorithm is proposed. The tradeoff between its computational complexity and time complexity can be adjusted freely for a certain given error rate performance. Thus, it can meet requirements of different applications at as low computational complexity and computing resource utilization as possible. Third, an enhanced ELSCS algorithm is proposed on the basis of LSCS in order to further decrease the time complexity without escalating computational complexity.

The rest of this chapter is organized as follows. Section 4.2 gives an in-depth analysis to the principles of SCL and SCS decoding algorithms. In Section 4.3, an LLR-threshold based path extension method is provided, based on which we propose a novel LSCS algorithm in Section 4.4. Afterwards, an enhanced version ELSCS is presented in Section 4.5 while the simulation results are shown in Section 4.6. Finally, Section 4.7 gives a summary of the chapter.

## 4.2 SCL and SCS decoding algorithms

The error rate performance of polar codes mainly depends on the decoding algorithm [80]. SC is the first proposed decoding algorithm for polar codes. SCL and

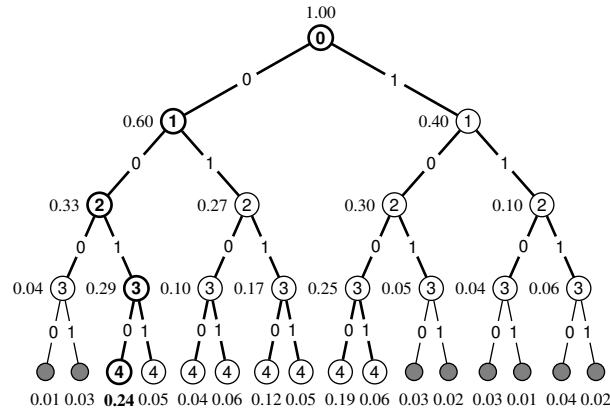


Figure 4.2 An example of the SCL decoding with  $L = 4$ .

SCS are derived from SC but incorporate remarkable improvements to achieve significant error rate performance gain. Moreover, when SCL and SCS are concatenated with cyclic redundancy check [85, 130], their decoding performance can be further boosted.

Unlike the SC decoder, which uses hard-decision for each bit and has only one search path, SCL and SCS extend the decoding path to two new paths by appending a bit 0 or a bit 1 when an unfrozen bit is encountered. Figure 4.2 and Figure 4.3 show simple examples of the tree search process of SCL and SCS decoding algorithms, respectively. The bold branches represent the decoding path. The number next to each node gives the *a posteriori* probability of the decoding path from the root to that node. The nodes generated after path extensions are represented by the numbered circles with the numbers indicating the extension stage order and the node with bold number is the correct decoding path. The gray ones are nodes that are not visited during the search process.

SCL will firstly sort the new generated paths by their path metrics (PM) defined in (2.41). Thereupon, a maximum of  $L$ , where  $L$  denotes list size, paths with the same length and the largest metrics are selected for next path extension stage without storing other paths. The path length is increased by one at each path extension stage. When decoding reaches the block-length  $N$ , the path with the maximum metric is output as decoding sequence. SCS stores all generated paths in a stack and selects the top path with the maximum metric to extend each time. An additional parameter, search width  $Q$ , is added to limit the number of extending paths with certain lengths in the decoding process. Whenever the top path with the largest metric in the stack reaches block-length  $N$ , the decoding process stops and outputs this path. Compared with SC, SCL and SCS have much wider search range and less probability to fall into a local optimum, which certainly ensures an improved error rate performance.

As SCL is a depth-first search algorithm, it extends  $L$  paths with the largest metrics at the same length each time and the decoding length keeps increasing. Different from SCL, SCS employs breadth-first search, thus selecting only one path

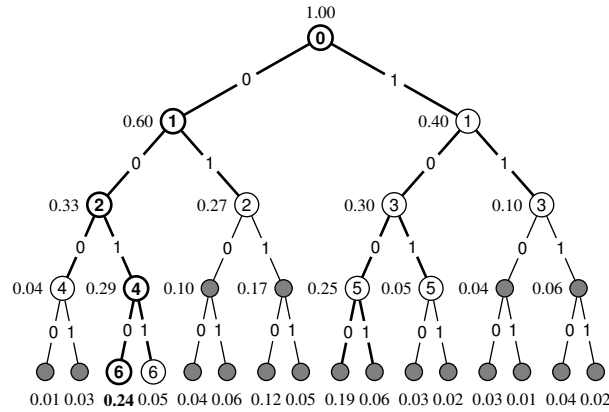


Figure 4.3 An example of the SCS decoding.

with the maximum metric to extend each time. The decoding path length in SCS is not always increasing after each extension. Therefore, it takes the decoding path a long time to reach block-length  $N$ . This explains why SCS has a larger decoding latency compared to SCL. As shown in the examples, SCL only needs to extend 4 times to complete the decoding while SCS needs 2 more. Another difference is that SCS uses a stack to store some candidate partial decoding paths with different lengths. In case of good channel quality, the correct path has a larger metric than incorrect ones. For the sake of using a stack, SCS can selectively choose candidate partial path with large metric to search instead of searching  $L$  paths blindly like SCL. Therefore, compared to SCL with the same error rate performance (when  $Q = L$ ), a lot of unnecessary search is avoided. The computational complexity of SCL scales like  $L \cdot N \cdot \log N$  and that of SCS is variable and much less than that of SCL decoding. We can see from the two examples in Figure 4.2 and Figure 4.3, 5 path extension operations are saved by SCS compared to SCL.

In CRC-aided decoding, the source information contains a built-in CRC and it chooses the path with the maximum metric from those who pass CRC as output when the decoding length reaches  $N$ . In the case of poor channel quality, the correct path does not necessarily have the largest metric, but certainly passes the CRC verification. For this reason, CRC aided decoding can further improve the probability of successful decoding.

### 4.3 LLR-threshold based path extension scheme

To combine the advantages of SCS and SCL, we need to adopt both stack decoding and list decoding. However, in stack decoding, much space is occupied by the stack to store candidate partial decoding paths with different lengths. If list decoding is further considered, more candidate partial paths need to be stored, which will occupy considerable device memory. Aiming at decreasing the required space, we propose an LLR-threshold based path extension (LTPE) scheme. Before illustrating



it, a conjecture is given, which will provide some intuition for the scheme. If the decoding bit of the correct path is an unfrozen bit, the below phenomenon happens with a large probability. Among the two generated paths after path extension, the metric of one path remains almost unchanged and that of the other path will decrease a lot. The higher reliability of the corresponding polarization channel, the greater probability of this phenomenon. Next, we will give an empirical evidence of this conjecture.

The update function of metric used for path selection in (2.42) can be expressed as  $\text{PM}_i[\ell] = \phi(\text{PM}_{i-1}[\ell], L_N^{(i)}[\ell], \hat{u}_i[\ell])$  where  $\text{PM}_i[\ell]$  represents the metric of path  $\ell$  with length  $i$ .  $\hat{u}_i[\ell]$  represents the estimated value of the  $i$ -th bit of path  $\ell$ .  $L_N^{(i)}[\ell]$  represents  $L_N^{(i)}$  of path  $\ell$ . The function  $\phi(\mu, \lambda, u) \triangleq \mu - \ln(1 + e^{-(1-2u)\lambda})$  can be approximated to

$$\tilde{\phi}(\mu, \lambda, u) \triangleq \begin{cases} \mu, & \text{if } u = \frac{1}{2}[1 - \text{sign}(\lambda)], \\ \mu - |\lambda|, & \text{otherwise.} \end{cases} \quad (4.1)$$

Thus, the update function can be recast as

$$\widetilde{\text{PM}}_i[\ell] \triangleq \begin{cases} \text{PM}_{i-1}[\ell], & \text{if } \hat{u}_i[\ell] = \frac{1}{2}[1 - \text{sign}(L_N^{(i)}[\ell])], \\ \text{PM}_{i-1}[\ell] - |L_N^{(i)}[\ell]|, & \text{otherwise.} \end{cases} \quad (4.2)$$

Next, we use  $L_i$  to represent  $L_N^{(i)}$  of the correct path and analyze the value of  $L_i$ . Consider a Gaussian channel, under the assumption of Gaussian approximation,  $L_i \sim \mathcal{N}(\mathbb{E}[L_i], 2|\mathbb{E}[L_i]|)$  [131], where,

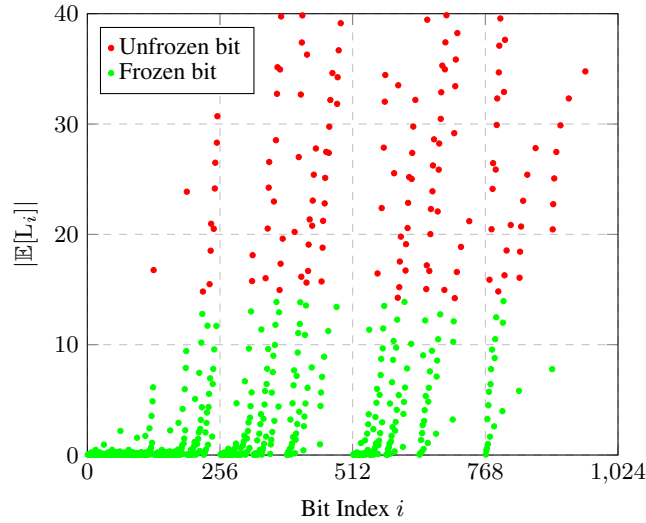
$$\mathbb{E}[L_i] = \begin{cases} \mathbb{E}[L_i(0)], & \text{if } u_i = 0, \\ -\mathbb{E}[L_i(0)], & \text{if } u_i = 1. \end{cases} \quad (4.3)$$

$\mathbb{E}[L_i(0)]$  denotes the expectation of  $L_i$  for the all-zero code word transmitted over a binary-input additive white Gaussian noise channel (BI-AWGNC) with binary phase shift keying (BPSK) modulation and noise variance  $\sigma_n^2$ , which can be calculated recursively by the following equations.

$$\begin{aligned} m_1^{(1)} &= 2/\delta_n^2, \\ m_{2n}^{(2j-1)} &= \varphi^{-1}(1 - [1 - \varphi(m_n^{(j)})]^2), \\ m_{2n}^{(2j)} &= 2m_n^{(j)}, \\ \mathbb{E}[L_i(0)] &= m_N^{(i)}, \end{aligned} \quad (4.4)$$

where

$$\varphi(x) = \begin{cases} 1 - \frac{1}{\sqrt{4|x|}} \int_{-\infty}^{\infty} \tanh\left(\frac{u}{2}\right) e^{-\frac{(u-x)^2}{4|x|}} du, & x \neq 0, \\ 0, & x = 0. \end{cases} \quad (4.5)$$

Figure 4.4  $|\mathbb{E}[L_i]|$  with different bit indices  $i$ .Table 4.1  $\min(|\mathbb{E}[L_i]|)$  for unfrozen bits with different  $\gamma$ .

$\gamma$ ( $N = 1024$ )	1.0 dB	1.5 dB	2.0 dB	2.5 dB	3.0 dB
$\min( \mathbb{E}[L_i] ),$ $i$ is unfrozen bit	5.38	9.38	14.20	23.00	30.00

Given  $\delta_n^2$ ,  $\mathbb{E}[L_i(0)]$  and  $\mathbb{E}[L_i]$  can be calculated in an off-line manner.

The relationship between the reliability of polarized channel and  $\mathbb{E}[L_i]$  is shown as below.

$$\begin{aligned}
 P_e(u_i) &= \int_{-\infty}^0 \frac{1}{2\sqrt{\beta|\mathbb{E}[L_i]|}} \cdot \exp\left(\frac{-(x-|\mathbb{E}[L_i]|)^2}{4|\mathbb{E}[L_i]|}\right) dx \\
 &= Q(\sqrt{|\mathbb{E}[L_i]|}/2),
 \end{aligned} \tag{4.6}$$

where  $P_e(u_i)$  represents the probability that  $u_i$  is incorrectly estimated in terms of the subchannel  $W_N^{(i)}(y_1^N, u_1^{i-1}|u_i)$ , given the correct prior bits  $u_1^{i-1}$ . The smaller  $P_e(u_i)$ , the higher the channel reliability.  $Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^{+\infty} e^{-\frac{t^2}{2}} dt$  is a monotonically decreasing function, thus the channel reliability increases with the increasing value of  $|\mathbb{E}[L_i]|$ .

The unfrozen bits are chosen according to the channel reliability and corresponds to the  $N \cdot R$  polarization channels with the smallest  $P_e(u_i)$  and the largest  $|\mathbb{E}[L_i]|$ . Consequently, for unfrozen bit  $i$ ,  $|L_i|$  has a high probability to be a large value, and the greater reliability polarization channel  $i$  has, with higher probability this happens. We can see from Figure 4.4, where the points above 40 are projected onto the  $\geq 40$  line for convenience, that when  $N = 1024$ ,  $\gamma = 2$  dB, the smallest  $|\mathbb{E}[L_i]|$  for unfrozen bits is 14.2, the values at some other  $\gamma$  are also shown in Table 4.1.

When the correct decoding path comes to unfrozen bit  $i$  in a sequential decoding, the value of  $|L_i|$  is large, especially when the corresponding polarization channel has high reliability. Consider (4.2), it happens with large probability that the metric of one path remains almost unchanged after path extension and another will decrease considerably.

The one with smaller metric is very difficult to become the final decoding result because of the big metric gap. However, these kind of paths occupy a lot of storage space. Hence, we consider to set an LLR threshold. If the metric difference is larger than the threshold, the one with smaller metric will be deleted. The other one whose metric almost remains constant will continue to be extended at the next path extension stage and does not need to be considered in sorting stage. Let  $\delta$  ( $\delta > 0$ ) denote the LLR threshold. The LTPE scheme is summarized in Algorithm 1.

---

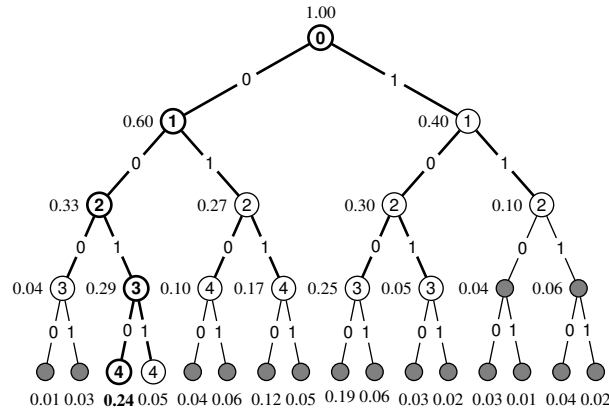
**Algorithm 1:** LLR-threshold Based Path Extension Scheme

---

$\mathcal{A}$  denotes the stack for storing paths to be extended;  
 $\mathcal{B}$  denotes the stack for storing generated paths after extension;  
**if**  $u_{i+1}$  is unfrozen bit **then**  
    **if**  $|L_{i+1}[\ell]| \geq \delta$  **then**  
        Delete  $(u_1^i[\ell], u_{i+1}), u_{i+1} \neq \frac{1}{2}[1 - \text{sign}(L_{i+1}[\ell])]$ ;  
        Reserve  $(u_1^i[\ell], u_{i+1}), u_{i+1} = \frac{1}{2}[1 - \text{sign}(L_{i+1}[\ell])]$  in stack  $\mathcal{A}$  for next path extension stage directly;  
    **else**  
        Push both paths into stack  $\mathcal{B}$  for sorting process;  
    **end**  
**else**  
    Push  $(u_1^i[\ell], 0)$  into stack  $\mathcal{B}$  for sorting process;  
**end**

---

In Algorithm 1,  $L_{i+1}[\ell]$  represents  $L_N^{(i+1)}$  of the decoding path  $\ell$ ,  $(u_1^i[\ell], u_{i+1})$  denotes the generated new path after path  $\ell$  chooses  $u_{i+1}$  to extend at the  $(i+1)$ -th bit. This LTPE strategy can reduce the number of paths inserted to the stack, avoiding the excessive space occupied by those paths that are difficult to become the final decoding output. As fewer bits are required for the quantization of LLR than metric in hardware implementation, it is easier to compare LLR than metric [112]. In addition, the deleted paths in LLR-based scheme are not required to execute (4.2) for metric computation. Hence, the proposed LLR-based path pruning scheme is more convenient to implement in practice, compared with existing metric-based path pruning schemes. Of course, we can also use metric-based schemes after the proposed LLR-based scheme to further lower the memory requirements, at the cost of higher implementation complexity.

Figure 4.5 An example of the LSCS decoding with  $L = 2$ ,  $Q = 4$ .

For how to suitably select the threshold  $\delta$  to reduce the storage requirements while maintaining error rate performance unaffected, we will give an in-depth analysis and discussion in simulation part in Section 4.6.

## 4.4 List-aided successive cancellation stack decoding algorithm

Motivated by the advantage of SCL, we introduce list decoding to speed up the breadth search of stack decoding. A new LSCS decoding is proposed on the basis of the path extension method in Algorithm 1. At each extension stage, LSCS chooses a maximum of  $L$  paths with the largest metrics in the stack to extend simultaneously. The detailed steps are shown in Algorithm 2 and an example is given in Figure 4.5. This algorithm can offer a flexible tradeoff between time complexity and computational complexity by adjusting the value of  $L$ . The adjustable complexity performance varies among maximum time/minimum computational complexity (similar to SCS) and minimum time/maximum computational complexity (similar to SCL) and some intermediate complexity states freely, while ensuring a stable error rate performance.

Considering the iterative property of LLR operations in  $f/g$  function in (2.37) and (2.38), it is difficult to be completed synchronously, which directly affects the decoding delay and throughput. For example,  $\log_2 N$  iterations are required to calculate the value of  $L_N^{(1)}$ . Each iteration needs to call  $f/g$  function. If one time step is required for each execution of  $f/g$  function, the calculation of  $L_N^{(1)}$  requires at least  $\log_2 N$  time steps. In this manner, when  $L$  paths are processed in parallel, to simplify the time complexity evaluation of LSCS decoding, we can suppose that all the parallelizable executions of  $f/g$  function are performed in one time step [132] and measure the time complexity in terms of the maximum number of time steps for each decoding.

**Algorithm 2:** The LSCS( $Q, L, D$ ) Decoder

---

Define  $Q$  as the maximum tolerable number of CRC check;  
Define  $L$  ( $1 \leq L \leq Q$ ) as the number of paths that are extended simultaneously at each extension stage;

**1) Initialization**  
Create a stack  $\mathcal{A}$  with depth  $L$ ;  
Create a stack  $\mathcal{B}$  with depth  $D$ ;  
Push a null path into  $\mathcal{A}$ , set path metric to 0;  
Initialize counter,  $q_1^N = \text{zeros}(N, 1)$ ;

**2) Path competition**  
Define  $u_1^i[\ell]$  as path  $\ell$  with a length  $i$  in stack  $\mathcal{A}$ ;  
**if**  $u_1^i[\ell] \neq \text{null}$  **then**  
     $q_i = q_i + 1$ ;  
    **if**  $q_i = Q$  **then**  
        Delete all paths in  $\mathcal{B}$  with length less than or equal to  $i$ ;  
    **end**  
**end**

**3) Path extension**  
Extend the paths in stack  $\mathcal{A}$  according to Algorithm 1;

**4) Path selection and sorting**  
After all  $L$  paths in stack  $\mathcal{A}$  complete path extension;  
**if**  $|\mathcal{A}| = L$  **then**  
    Sort the paths in stack  $\mathcal{B}$  in descending metric;  
**else**  
    Push the  $L - |\mathcal{A}|$  paths with the maximum metrics in stack  $\mathcal{B}$  to stack  $\mathcal{A}$   
    and sort the rest paths in stack  $\mathcal{B}$  in descending metric;  
**end**

**5) CRC-aided termination decision**  
**if** exist paths in  $\mathcal{A}$  with a length  $N$  **then**  
    Perform CRC detection;  
    **if** CRC detection pass **then**  
        Output the path with the maximum metric as decoding sequence;  
    **else**  
         $q_N = q_N + 1$ ;  
        **if**  $q_N < Q$  **then**  
            Go to step 2;  
        **else**  
            Declare a decoding failure;  
        **end**  
    **end**  
**else**  
    Go to step 2;  
**end**

---

The time complexity of LSCS decoding is given by

$$T_{\text{LSCS}} \triangleq \sum_{k=1}^K \left( \max_{j \in \llbracket L \rrbracket} \mathcal{O}_t(\ell_j^{(k)}) \right), \quad (4.7)$$

where  $\ell_j^{(k)}$  denotes the  $j$ -th extending path at the  $k$ -th extension stage.  $\mathcal{O}_t(\cdot)$  is the time complexity calculation operation and  $\mathcal{O}_t(\ell_j^{(k)})$  represents the number of time steps required by the  $j$ -th path at the  $k$ -th extension stage, which is in the range of 1 to  $\log_2 N$ .  $K$  denotes the total number of extension stages in the decoding process.

For the computational complexity, we refer to [85] and use the number of LLR operations in  $f/g$  function to measure the decoding computational complexity, which determines the amount of energy consumption approximately. The computational complexity of LSCS decoding is expressed as

$$C_{\text{LSCS}} \triangleq \sum_{k=1}^K \left( \sum_{j=1}^L \left( \mathcal{O}_c(\ell_j^{(k)}) \right) \right), \quad (4.8)$$

where  $\mathcal{O}_c(\cdot)$  is the computational complexity calculation operation.  $\mathcal{O}_c(\ell_j^{(k)})$  represents the number of LLR operations required by the  $j$ -th path at the  $k$ -th extension stage.

Because the change of list size only influences the path searching speed and is unrelated to the search range, the error rate performance will not be affected. It is expected that with the increasing number of paths searched simultaneously at each extension stage, the search speed accelerates and LSCS will need less extension stages  $K$  to find the correct path compared with SCS. Therefore, according to (4.7), this kind of parallelized search strategy will reduce the average decoding time complexity. However, the downside is the increase in computational complexity. Although  $K$  decreases, the reduction magnitude of  $K$  is not proportional to the size of  $L$ . This is because more incorrect paths will be searched when  $L$  increases, which reduces the search efficiency. Meanwhile,  $\sum_{j=1}^L \mathcal{O}_c(\ell_j^{(k)})$  increases almost linearly with the size of  $L$ . The increasing number of LLR operations for an extension stage becomes the dominant influencing factor instead of the decreasing  $K$ . As a consequence, according to (4.8), the average computational complexity will increase.

When  $L = 1$ , the LSCS is similar to SCS, extending one path each time. In this case, the time complexity is maximized whereas the computational complexity is minimized. When  $L = Q$ , LSCS is similar to SCL, extending  $Q$  paths with the same length simultaneously, the time complexity is minimized whereas the computational complexity is maximized at this time. When  $L$  increases from 1 to  $Q$ , the time complexity decreases and the computational complexity increases. The simulation results shown in Section 4.6 match well with the above analysis. LSCS can flexibly change its complexity performance among the two extreme states and

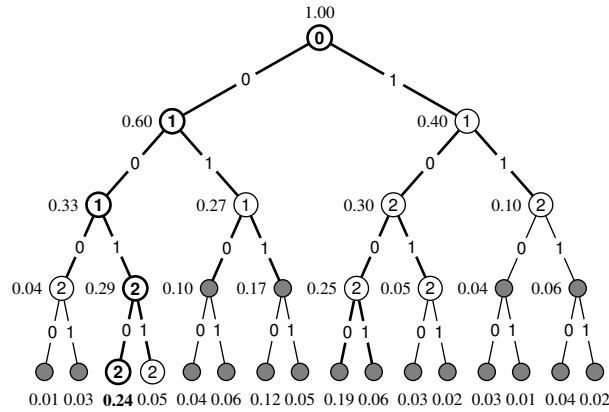


Figure 4.6 An example of the ELSCS decoding with  $L = 2, Q = 4$ .

their intermediate states by adjusting the list size while the error rate performance remains almost unaltered.

## 4.5 Enhanced list-aided successive cancellation stack decoding algorithm

Conversely to SCL, where all the  $L$  extending paths have the same length, in LSCS the  $L$  extending paths may have different lengths if  $1 < L < Q$ . If the length  $i$  of a extending path is odd, the number of time steps to obtain  $L_N^{(i+1)}$  is only 1. Otherwise, more time steps are required to calculate the value of  $L_N^{(i+1)}$ . For example, it will take  $\log_2 N$  time steps to calculate the value of  $L_N^{(1)}$  if  $i = 0$ . Considering parallel processing, every extending path is assigned a processing element (PE). According to the above analysis, at the extension stage, some PEs will run for a long time while some will complete their tasks quickly and wait until other PEs stop working. This not only can not make full use of PE resources but also increases the decoding latency. To solve this problem, we allow each path extend two sequential bits at the extension stage, which means that each extension stage consists of two extensions. For example, when path  $\ell$  generates new paths after the extension 1 at its first bit  $i$ , it does not need to wait until all other paths finish the extension 1 at their first bit, the new generated path with larger metric is chosen to continue to extend at the second bit  $i + 1$ . When all paths have finished extension 2 at their second bit, the sorting operation is performed. As a result of extending two sequential bits, the indices of the two extended bits must have an odd and an even number, which makes the runtime of PEs corresponding to different paths slightly differ from each other. This improves the PE resource utilization efficiency. Thus, an enhanced LSCS (ELSCS) is proposed in Algorithm 3 (Its steps 1, 2, 4, 6, 7 are the same as Algorithm 2) to further decrease the time complexity and a simple example is shown in Figure 4.6.

Despite this difference between the path extension schemes of ELSCS and LSCS, the path search range of ELSCS does not diminish compared with that of LSCS due to the use of a stack. Thus, the error rate performance will remain unchanged.

The time complexity of ELSCS decoding is given by

$$T_{\text{ELSCS}} \triangleq \sum_{k=1}^K \left( \max_{i \in \llbracket 2 \rrbracket, j \in \llbracket L \rrbracket} \mathcal{O}_t(\ell_j^{(k_i)}) + 1 \right), \quad (4.9)$$

where  $\ell_j^{(k_i)}$  denotes the  $j$ -th extending path at extension  $i$  of the  $k$ -th extension stage.  $\mathcal{O}_t(\ell_j^{(k_i)})$  indicates the number of time steps required by the  $j$ -th extending path  $\ell$  at extension  $i$  of the  $k$ -th extension stage.  $\max_{i \in \llbracket 2 \rrbracket, j \in \llbracket L \rrbracket} \mathcal{O}_t(\ell_j^{(k_i)})$  is the maximum number of time steps required to extend paths with an even length. 1 is the number of time steps required to extend the paths with an odd length.  $K$  denotes the total number of extension stages in the decoding process.

---

**Algorithm 3:** The Enhanced LSCS( $Q, L, D$ ) Decoder

---

Define  $Q$  as the maximum tolerable number of CRC check;  
 Define  $L$  ( $1 \leq L \leq Q$ ) as the number of paths that are extended simultaneously at each extension stage;

**1) Initialization**

**2) Path competition**

**3) Path extension 1:** Extend path  $u_1^i[\ell]$  to length  $i + 1$

**if**  $u_{i+1}$  is unfrozen bit **then**

Reserve  $(u_1^i[\ell], u_{i+1}), u_{i+1} = \frac{1}{2}[1 - \text{sign}(L_{i+1}[\ell])]$  in stack  $\mathcal{A}$  for path extension 2 directly;

**if**  $|L_{i+1}[\ell]| \geq \delta$  **then**

Delete another path;

**else**

Push another path into stack  $\mathcal{B}$  for sorting process;

**end**

**else**

Reserve  $(u_1^i[\ell], 0)$  in stack  $\mathcal{A}$  for path extension 2 directly;

**end**

**4) Path competition**

**5) Path extension 2:** Extend path  $u_1^{i+1}[\ell]$  to length  $i + 2$

Extend the paths in stack  $\mathcal{A}$  according to Algorithm 1;

**6) Path selection and sorting**

**7) CRC-aided termination decision**

---

The maximum number of time steps for an extension stage  $k$  in LSCS is



$$\max_{j \in \llbracket L \rrbracket} \mathcal{O}_t(\ell_j^{(k)}) \quad (4.10)$$

The maximum number of time steps for an extension stage  $k$  in ELSCS is

$$\max_{i \in \llbracket 2 \rrbracket, j \in \llbracket L \rrbracket} \mathcal{O}_t(\ell_j^{(k_i)}) + 1 \quad (4.11)$$

Comparing (4.10) and (4.11), the average maximum number of time steps for an extension stage in ELSCS may be slightly larger than that in LSCS.

However, because of extending two bits in each extension stage, the average number of extension stages required for each decoding in ELSCS are much fewer than that in LSCS. Thus, when the average total maximum number of time steps for a decoding process is considered, ELSCS performs better than LSCS. In other words, extension 1 reserves the extended path with larger metric for extension 2 directly, allowing each path to extend two bits in succession without waiting for other paths, increasing the PE resource utilization efficiency. Therefore, it will take ELSCS less time to complete the computations for a decoding process compared with LSCS.

The computational complexity of ELSCS decoding is expressed as

$$C_{\text{ELSCS}} \triangleq \sum_{k=1}^K \left( \sum_{j=1}^L \left( \sum_{i=1}^2 \left( \mathcal{O}_c(\ell_j^{(k_i)}) \right) \right) \right), \quad (4.12)$$

where  $\mathcal{O}_c(\ell_j^{(k_i)})$  indicates the number of LLR operations required by the  $j$ -th extending path  $\ell$  at extension  $i$  of the  $k$ -th extension stage.  $\sum_{j=1}^L (\sum_{i=1}^2 (\mathcal{O}_c(\ell_j^{(k_i)})))$  means the number of LLR operations for an extension stage.

For extending two sequential bits at each extension stage, compared to LSCS, the average number of LLR operations for an extension stage in ELSCS doubles while the average number of extension stages required for each decoding in ELSCS nearly halves. Thus, the average total number of LLR operations for a decoding process in ELSCS will not differ much from that in LSCS. In Section 4.6, simulation results also show that ELSCS has almost the same computational complexity with LSCS.

In terms of hardware resources, ELSCS and LSCS both need up to  $Q$  PEs. The number is identical with that of SCL when they have the same error rate performance. However, the  $Q$  PEs in LSCS/ELSCS are not necessarily all used. The exact number of used PEs is equal to list size  $L$ . If  $L < Q$ , only partial PEs are used for decoding, the remaining PEs can be used by other modules such as demodulation and equalization. So LSCS/ELSCS will have a lower computing resource occupancy compared with SCL at this time. Another difference is more storage space for LSCS/ELSCS when  $L < Q$ . As the algorithm is adjustable, if there is not enough memory,  $L$  can be set equal to  $Q$  in LSCS/ELSCS and the required stack depth will reduce to a minimum of  $2L$  like in SCL, at the cost of higher computational complexity.

## 4.6 Numerical results

Numerical simulation results for BI-AWGNC are presented in this section to compare the block error rate (BLER) and complexity of different polar codes decoding algorithms.  $10^7$  code blocks are transmitted. All the used codes have code length  $N = 1024$  and code rate  $R = 1/2$ , a CRC-24 code with generator polynomial  $g(D) = D^{24} + D^{23} + D^6 + D^5 + D + 1$  used in [85] is adopted.

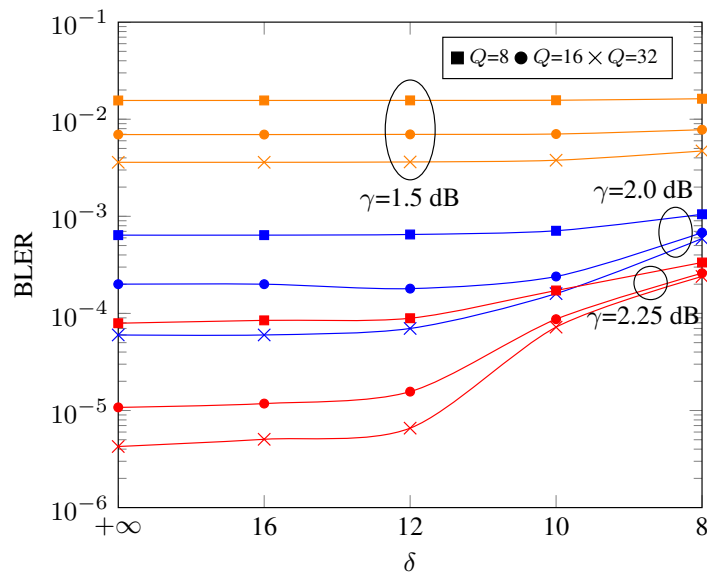


Figure 4.7 BLER performance of ELSCS ( $L=1$ ) with different  $\delta$ .

Figure 4.7 shows the BLER of ELSCS decoder with different  $Q$  and  $\gamma$  under different LLR threshold  $\delta$ . When  $\delta$  decreases, the error rate performance degrades, especially when  $\delta < 12$ . The explanation for this behaviour is that the correct path has a higher probability to be pruned under a smaller  $\delta$ , resulting in a larger decoding failure probability. It can also be observed that this trend is not significant in the low SNR regime. Because low SNR leads to small  $|\mathbb{E}[L_i]|$ , less pruning will happen and the correct path is more likely to be retained. The BLER curves of ELSCS with different list sizes all follow the same change trend as in Figure 4.7. By comprehensive consideration of both the pruning capability and influence on BLER performance,  $\delta$  is set to 12 in the following simulation analysis. The deterioration of the error rate performance is negligible at this time and a uniform integer value threshold makes threshold comparison simple and easy for hardware implementation. In addition, we find that when the value of  $Q$  is 16, an ideal error rate performance ( $\text{BLER} \leq 2 \times 10^{-4}$ ) can be obtained from  $\gamma$  greater than 2.0 dB. When  $Q = 32$ , although the error rate performance is superior, the computational complexity will be much higher as it is generally proportional to the value of  $Q$ . Thus, considering the convenience of practical realization, the value of  $Q$  is set to 16.

Figure 4.8 illustrates that the error rate performance of LSCS/ELSCS almost remains unchanged with different  $L$  values and is consistent with that of SCS and

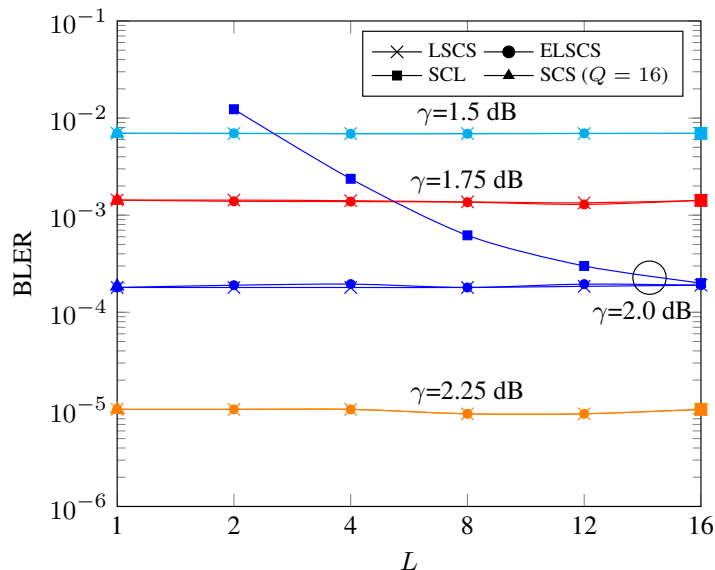


Figure 4.8 BLER performance of different decoding algorithms with different  $L$  values and  $\gamma$  when  $Q = 16$ ,  $\delta = 12$ .

SCL. Since the increasing list size only speeds up the path search and does not reduce the path search range, the probability of successful decoding does not decline. Particularly, when  $L = 16$ , the decoding principle and performance of ELSCS is similar to SCL, they both extend 16 paths with the same length each time, the difference is that SCL chooses 16 paths with the maximum metrics after sorting for next extension stage, while, ELSCS may reserve some paths for next extension directly without sorting operation. However, simulation results show that the performance deterioration caused by this difference can be neglected. This is because the path extension characteristics of the reserved paths match well with that of the correct one when  $\delta$  is large enough, so this path reservation strategy barely reduces the existence probability of the correct path among the decoding paths.

The BLER curve of SCL with different  $L$  at  $\gamma = 2.0$  dB is also plotted in Figure 4.8. It increases with the decrease of  $L$  and is higher than that of LSCS/ELSCS with the same  $L$ . When  $L = 4$ , ELSCS can obtain a performance gain of more than 0.25 dB compared with SCL at the BLER of  $1.4 \times 10^{-3}$  and when  $L = 2$ , the performance gain increases to 0.5 dB at the BLER of  $7 \times 10^{-3}$ .

Figure 4.9 shows the normalized time complexity  $\eta_t$  and computational complexity  $\eta_c$  of different decoding algorithms with different  $L$  values. They are defined by

$$\eta_t = \frac{\bar{T}_*}{\bar{T}_{\text{SCS}}}, \quad (4.13)$$

$$\eta_c = \frac{\bar{C}_*}{\bar{C}_{\text{SCL}}}, \quad (4.14)$$

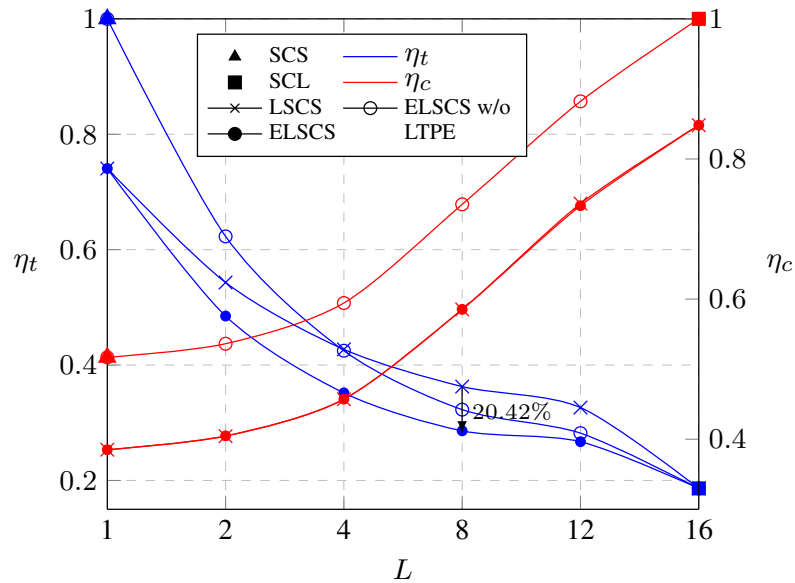


Figure 4.9 Normalized time and computational complexities of different decoding algorithms with different  $L$  values at  $\gamma = 2$  dB when  $Q = 16$ ,  $\delta = 12$ .

where  $\bar{T}_*$  and  $\bar{C}_*$  indicate the average time and computational complexity of different decoding algorithms, respectively.  $\bar{T}_{\text{SCS}}$  denotes the average time complexity of SCS with  $Q = 16$  and  $\bar{C}_{\text{SCL}}$  denotes the average computational complexity of SCL with  $L = 16$ . The average time and computational complexity of LSCS and ELSCS are calculated using (4.7), (4.8), (4.9), (4.12), respectively.

We can see that it is difficult to obtain both optimal computational and time complexity performance under certain given error performance. Minimum computational complexity often means maximum time complexity and vice versa. For LSCS and ELSCS, we can adjust the tradeoff between computational and time complexity by changing the list size. In Figure 4.9, the BLER is constant at  $2 \times 10^{-4}$ . When  $L = 1$ , the decoding principle and performance of ELSCS are similar to SCS, achieving the minimum computational complexity and the maximum time complexity. On the contrary, when  $L = 16$ , ELSCS extends 16 paths with the same length each time, similar to SCL, thus, has the maximum computational complexity and the minimum time complexity. When list size is between 1 and 16, there will be many intermediate performance states. As list size increases, the average maximum number of time steps for each extension stage may become slightly larger. However, the average number of path extension stages required for each decoding drops, leading to the declination of the average total maximum number of time steps for each decoding. Consequently, the time complexity decreases. When considering the computational complexity, the average number of LLR operations for each extension stage increases dramatically with increasing list size and becomes the dominant influencing factor. Hence, the average total number of LLR operations rises and the computational complexity increases.

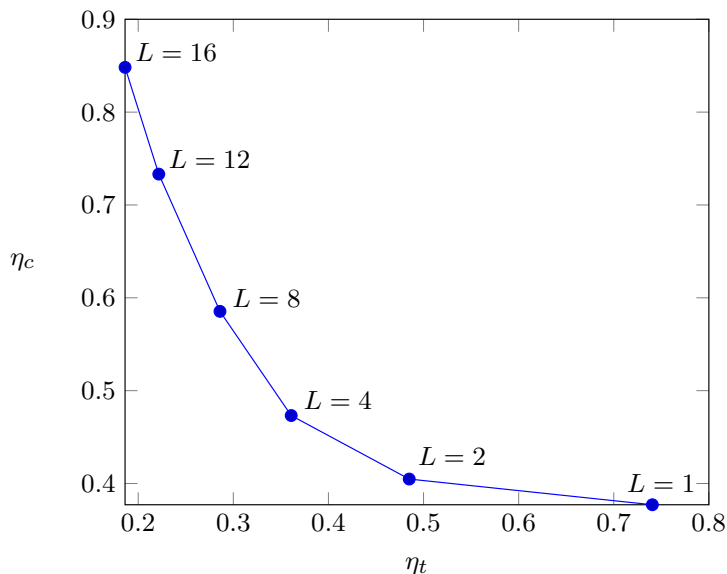


Figure 4.10 Normalized time and computational complexities of ELSCS with different modes at  $\gamma = 2$  dB when  $Q = 16$ ,  $\delta = 12$ .

As a comparison, ELSCS without LTPE scheme is depicted. It can be found that LTPE scheme reduces both time and computational complexity. This is because fewer candidate partial decoding paths are inserted into the stack as a result of the path pruning capability of LTPE scheme. The decoder does not need to search among many paths that are improbable to become the final decoding output. Consequently, ELSCS with LTPE scheme can get lower decoding time and computations compared to ELSCS without LTPE scheme. In addition, ELSCS without LTPE scheme can still provide a flexible tradeoff between time complexity and computational complexity. The complexity performance of two extreme cases when  $L = 1$  and  $L = 16$  is the same as that of SCS and SCL, respectively. Thus, even without LTPE scheme, ELSCS is still able to provide a low time complexity or computational complexity through the adjustment of list size  $L$ .

We can also observe that the computational complexity curves of LSCS and ELSCS overlap but the time complexity of ELSCS reduces in comparison to that of LSCS. As ELSCS extends two sequential bits at each path extension stage, the PE resource utilization efficiency is improved and the time complexity has a significant reduction. When  $L = 8$ , the relative time complexity of ELSCS is reduced by 20.42% compared with that of LSCS.

As shown in Figure 4.10, we can consider choosing  $L = 1, 2, 4, 8, 12, 16$  as different modes of ELSCS algorithm, thereby providing an adjustable decoding algorithm for different cases. A proper configuration of  $L$  can make the algorithm meet the specific decoding latency and throughput demand, while, reducing the computational complexity and the occupied number of PEs to the least possible degree. As a comparison, although SCL ( $L = 16$ ) can provide smaller latency, it requires more computational complexity and occupies more computing resources. When the decoding latency of ELSCS already meets the throughput requirement of applica-

tion, a smaller decoding latency with higher cost is not necessary. In the practical application of ELSCS algorithm, the value of  $Q$  is chosen firstly, which determines the error rate performance. Generally,  $Q$  is set to 16. Then the value of  $L$  is adjusted to select the appropriate algorithm mode, ensuring that the required throughput is achieved with the lowest computational complexity and minimum computing resources.

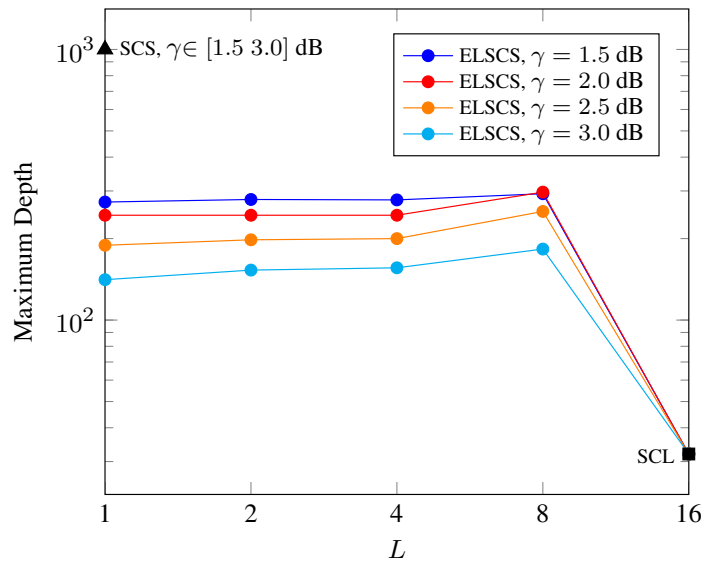


Figure 4.11 Maximum stack depth under different decoding algorithms with different  $L$  values and  $\gamma$  when  $Q = 16, \delta = 12$ .

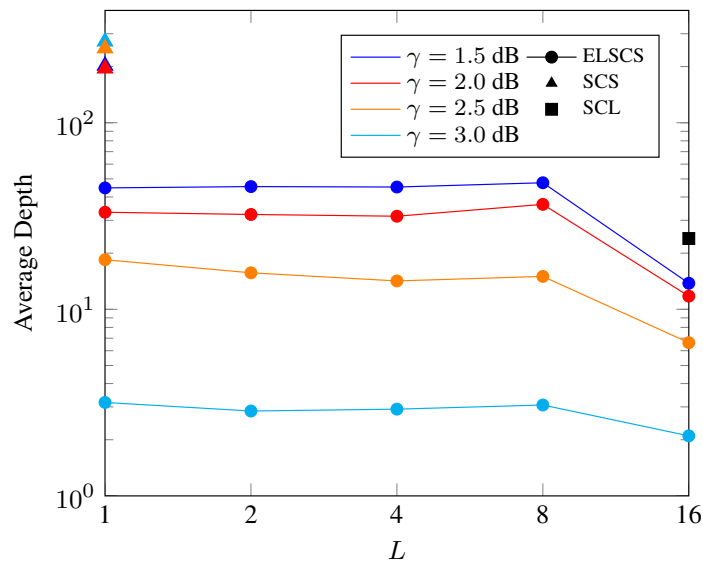


Figure 4.12 Average stack depth under different decoding algorithms with different  $L$  values and  $\gamma$  when  $Q = 16, \delta = 12$ .

Figure 4.11 and Figure 4.12 show the statistical results of the maximum stack depth and the average stack depth required by SCS and ELSCS. The stack depth is 1000 and represents the total depth of stack  $\mathcal{A}$  and stack  $\mathcal{B}$ . We observe that LTPE scheme reduces the required storage size dramatically. Compared with SCS, the maximum stack depth drops 70%, from 1000 to around 300, which means that a stack with depth of 300 is enough for the proposed algorithm. At  $\gamma = \{1.5 \text{ dB}, 2.0 \text{ dB}, 2.5 \text{ dB}, 3.0 \text{ dB}\}$ , the average storage space of ELSCS reduces to 22.27%, 15.92%, 6.57%, 1.19% of that of SCS, respectively. According to analysis in Section 4.3, the value of  $|\mathbb{E}[L_i]|$  at unfrozen bit will become larger when  $\gamma$  increases. Thus more paths will be pruned because of a large value of  $|L_i|$ . This explains why both the maximum and average depths become smaller when  $\gamma$  increases. The simulation results corroborate the effectiveness of LTPE scheme in path pruning. We can see that a large amount of storage space can be optimized by using ELSCS decoder for polar codes instead of conventional stack decoder.

## 4.7 Conclusion

In this chapter, we have proposed a complexity-adjustable multimode decoding algorithm ELSCS for polar codes. We firstly study the LLR characteristics of the correct path in decoding process. It was observed that if the decoding bit of the correct path is an unfrozen bit, it happens with a large probability that the path metric difference between two generated paths after path extension is large. A LTPE scheme is designed using this fact to reduce the storage space. Based on the proposed scheme, we employ both the ideas of SCL and SCS to introduce a novel LSCS algorithm, which can combine their complementary advantages. By changing the list size of extending paths, LSCS can adjust the tradeoff between time complexity and computational complexity flexibly while retaining the error performance unchanged. Driven by the low PE utilization problem in LSCS, LSCS is improved to obtain an enhanced version, ELSCS. ELSCS extends two sequential bits at extension stage to make the runtime of different PEs close and reduce the occurrences of some PEs staying idle and waiting. Thus, the time complexity can further decrease.

Performance and complexity analyses based on simulations show that without affecting error rate performance, ELSCS can not only reduce storage size but also provide a flexible tradeoff between time and computational complexity. Making use of this property, we can choose different modes of ELSCS algorithm to meet different application requirements at a low computational complexity and computing resource occupancy, thus helping mobile devices in reducing energy consumption as much as possible.

---

**HIGH ERROR-CORRECTION PERFORMANCE  
DECODER OF POLAR CODES**

---

This chapter is adapted from: 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*, 2019, 23(9): 1462-1465.

This chapter focuses on improving the error-correction performance of polar codes to cope with the challenge 2 of high error-correction performance. A new inter-frame correlated polar coding scheme is proposed, where two consecutive frames are correlated-encoded and assist each other during decoding. The correlation is achieved by dynamic configuration of the frozen bits. The frozen bits of the second frame partially depend on the unfrozen bits of the first frame in encoding and the number of bits that are viewed as frozen by decoder is alterable in different decoding modes. Using this new encoding/decoding scheme, a failed decoded frame can be decoded again with extra information which corrects the errors in its highly unreliable unfrozen bits. Thus the probability of successful decoding is improved. Simulation results show that the performance of the proposed polar codes outperforms that of the classical counterpart significantly with negligible memory and complexity increment.

## **5.1 Introduction**

Recent research work has been focused on improving the finite-length error-correction performance of polar codes, especially at moderate and short code length. By applying the partially information coupling technique [103–105], every two consecutive polar code blocks in a transport block (TB) are coupled by sharing a few information bits in [109]. In this way, the TB error rate can be improved while the equivalent code length becomes longer. Polar subcodes with dynamic frozen symbols are proposed in [133], where frozen bits values are determined by some information bits before them in the same frame. This kind of polar coding scheme has large minimum distance, but it only shows advantage over CA-SCL decoding when the list size is sufficiently large (above 32). Star polar subcodes further introduce the concept of star trellis, which enables a parallel decoding and obtains a slight error-correction performance gain with respect to polar subcodes in specific cases [134]. Research on SC flip decoding shows that in most failed decoding cases, the first error occurs in unfrozen bits with low reliabilities, i.e., the ones with small average log-likelihood



ratio (LLR) magnitudes. Once it is corrected, the probability of successful decoding increases significantly.

Inspired by this observation, we propose an inter-frame polar coding scheme with dynamic frozen bits. The values of frozen bits are set dynamically in order to make two consecutive frames correlated. The correlation is leveraged to correct the errors in the highly unreliable unfrozen bits of a failed decoded frame. Our results for a polar code of length 1024 and rate 1/2 show a coding gain of 0.28 dB over classical polar coding scheme at the block error rate (BLER) of  $10^{-4}$ . The rest of this chapter is organized as follows. The inter-frame polar coding is introduced in Section 5.2, including encoding and decoding schemes. Section 5.3 analyses the memory requirement and computational complexity of the proposed strategy. Simulation results are given in Section 5.4 and the summary is made in Section 5.5.

## 5.2 Inter-frame polar coding

### 5.2.1 Inter-frame correlated encoding scheme

A polar code of length  $N$  is constructed with a vector of relative reliabilities of bit indices  $\mathbf{v} = \{v_0, \dots, v_{N-1}\}$ , where bit index  $v_i$  is less reliable than bit index  $v_j$  if  $i < j$ . Figure 5.1 illustrates the classical polar encoding scheme and the proposed one. The classical polar coding scheme divides all  $N$  source bits into two sets according to  $\mathbf{v}$  and the number of unfrozen bits  $K$ . Unfrozen bits set  $\mathcal{A} = \{v_{N-K}, \dots, v_{N-1}\}$  contains the indices of  $K$  bits with higher reliabilities, and those of the remaining  $N - K$  bits form frozen bits set  $\mathcal{A}_c = \{v_0, \dots, v_{N-K-1}\}$ . In order to achieve a reasonable error-correction performance for polar codes with finite length, a CRC of length  $r$  is concatenated with polar codes and CA-SCL decoding is used. The  $r$  CRC bits and the  $K - r$  information bits are assigned to the bits with indices in  $\mathcal{A}$ . The frozen bits with indices in  $\mathcal{A}_c$  are fixed to predefined values known to the decoder.

The proposed polar coding scheme uses the same set  $\mathcal{A}$  for information and CRC bits. However, the difference between the proposed and the classical polar coding scheme is that some of the bits in  $\mathcal{A}_c$  are not fixed to predefined values that are known to the decoder. Here we set the predefined values that are known to the decoder to 0. Let  $\mathcal{A}_c^\wedge = \{v_{N-K-m}, \dots, v_{N-K-1}\}$  denote the set of  $m$  most reliable frozen bits (MRFBs) and  $\mathcal{A}^\vee = \{v_{N-K}, \dots, v_{N-K+m-1}\}$  denote the set of  $m$  most unreliable unfrozen bits (MUUBs). Note that  $\mathcal{A}_c^\wedge \in \mathcal{A}_c$ ,  $\mathcal{A}^\vee \in \mathcal{A}$ , and  $|\mathcal{A}_c^\wedge| = |\mathcal{A}^\vee| = m$ . We assign the MRFBs of a frame with the MUUBs of its preceding frame and we keep the remaining frozen bits as zeros. More formally, let  $C_i^{v_k}$  denote the value of the  $v_k$ -th bit of the  $i$ -th transmitted frame. The frozen bits assignment scheme is summarized in Algorithm 4.

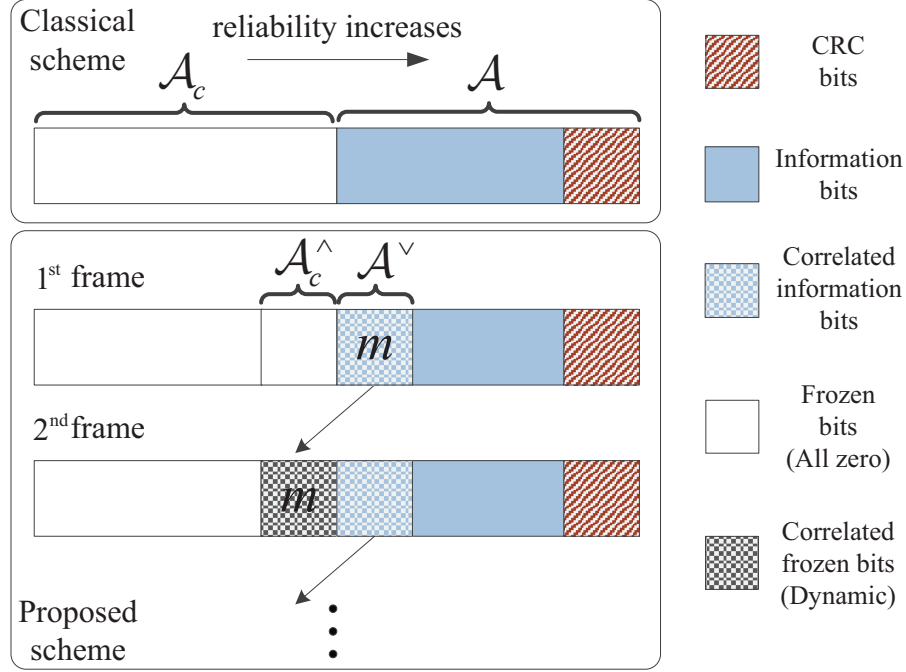


Figure 5.1 The classical and the proposed polar encoding schemes.

**Algorithm 4:** Frozen bits assignment scheme

```

for the  $i$ -th ( $i \geq 1$ ) transmitted frame do
  for  $k \leftarrow 0$  to  $N - K - 1$  do
    if  $i = 1$  then
       $C_i^{v_k} = 0$ ;
    else
      if  $k < N - K - m$  then
         $C_i^{v_k} = 0$ ;
      else
         $C_i^{v_k} = C_{i-1}^{v_{k+m}}$ ;
      end
    end
  end
end
end

```

**5.2.2 Inter-frame assisted decoding scheme**

Let  $\alpha_i$  represent the vector of channel LLR values for the  $i$ -th received frame and  $\widehat{C}_i^{v_k}$  represent the decoding estimation of bit  $v_k$  for the  $i$ -th received frame. We use

CA-SCL decoding algorithm to decode each frame. The classical CA-SCL decoder sets  $\widehat{C}_i^{v_k} = 0$  for  $v_k \in \mathcal{A}_c$ , and estimates the values of  $K$  unfrozen bits according to the output of CA-SCL decoding algorithm. As a result, the information from other frames is not required and whether the decoding succeeds or not does not affect the decoding of other frames. However, since consecutive frames are correlated-encoded in the proposed scheme, the decoding of a frame may require information from the two adjacent frames and the decoding result impacts the next decoding decision directly.

The proposed inter-frame assisted (IFA) SCL (IFA-SCL) decoding scheme is summarized in Algorithm 5. It uses an indicator  $I_f$  to indicate whether the decoding succeeded ( $I_f = 0$ ) or failed ( $I_f = 1$ ), an indicator  $I_g$  that shows whether consecutive decoding failures occurred ( $I_g = 1$ ) or not ( $I_g = 0$ ) in the decoding of the two latest frames, and is composed of four decoding modes as follows:

**M0** Perform CA-SCL decoding on frame  $i$  with the frozen bits defined as

$$\widehat{C}_i^{v_k} = 0, \quad 0 \leq k < N - K. \quad (5.1)$$

Set  $I_f$  in accordance with the decoding result.

**M1** Perform CA-SCL decoding on frame  $i$  with the frozen bits defined as

$$\begin{cases} \widehat{C}_i^{v_k} = 0, & 0 \leq k < N - K - m, \\ \widehat{C}_i^{v_k} = \widehat{C}_{i-1}^{v_{k+m}}, & N - K - m \leq k < N - K. \end{cases} \quad (5.2)$$

Set  $I_f$  in accordance with the decoding result.

**M2** Perform CA-SCL decoding on frame  $i$  with the frozen bits defined as

$$\widehat{C}_i^{v_k} = 0, \quad 0 \leq k < N - K - m. \quad (5.3)$$

Set  $I_f$  in accordance with the decoding result.

**M3** Perform CA-SCL decoding on frame  $i - 1$  with  $\alpha_{i-1}$  and partial frozen bits  $\widehat{C}_{i-1}^{v_k}$ ,  $N - K - m \leq k < N - K$ , fetched from memory, and define other frozen bits as

$$\begin{cases} \widehat{C}_{i-1}^{v_k} = 0, & 0 \leq k < N - K - m, \\ \widehat{C}_{i-1}^{v_k} = \widehat{C}_i^{v_{k-m}}, & N - K \leq k < N - K + m. \end{cases} \quad (5.4)$$

In order to determine whether the decoding is successful or not, we utilize the small undetected error probability of CRC [135] and define a successful decoding as when the decoding output passes the CRC. Algorithm 5 shows that if the decoding of frame  $i - 1$  succeeds, the decoding of frame  $i$  is executed in mode **M1**. The decoder uses the values of MUUBs of frame  $i - 1$  as those of MRFBS of frame  $i$ . On the other hand, if the decoding of frame  $i - 1$  fails, the decoding of frame  $i$  is executed in

---

**Algorithm 5:** IFA-SCL decoding scheme

---

```

Initialize  $I_g = 0$ ;
for the  $i$ -th ( $i \geq 1$ ) received frame do
    if  $i = 1$  then
        | M0;
    else
        if  $I_f = 0$  then
            | M1;
            if  $I_f = 0$  then
                | Store  $\widehat{C}_i^{v_k}$  for  $v_k \in \mathcal{A}^V$ ;
            else
                | Store  $\alpha_i$  and  $\widehat{C}_i^{v_k}$  for  $v_k \in \mathcal{A}_c^\wedge$ ;
            end
        else
            | M2;
            if  $I_f = 0$  then
                | Store  $\widehat{C}_i^{v_k}$  for  $v_k \in \mathcal{A}_c^\wedge$  and  $v_k \in \mathcal{A}^V$ ;
                if  $I_g = 0$  then
                    | M3;
                else
                    |  $I_g = 0$ ;
                end
            else
                |  $I_g = 1$ ;
            end
        end
    end
end
end

```

---

mode **M2**. The decoder regards the MRFBs as unfrozen bits since correct values of these bits cannot be obtained from frame  $i - 1$ . The indicator  $I_g$  ensures that frame  $i - 1$  undergoes a secondary decoding operation in mode **M3**, only if both frames  $i - 2$  and  $i$  are decoded correctly. In this case, the MRFB values of frame  $i - 1$  which are taken from frame  $i - 2$ , and the MUUB values of frame  $i - 1$  which are taken from frame  $i$ , are correct. This guarantees that the frames help each other in order to improve the error-correction performance.

Fig. 5.2 gives a detailed example of the proposed polar decoding scheme. The decoding of the first frame is the same with the classical decoding, that is, decoding mode **M0**. If it succeeds, decoding mode **M1** will be performed on the second frame. It also decodes  $K$  unfrozen bits and is only different from classical decoding in the values of MRFBs. They are assigned the values of MUUBs in the preceding frame

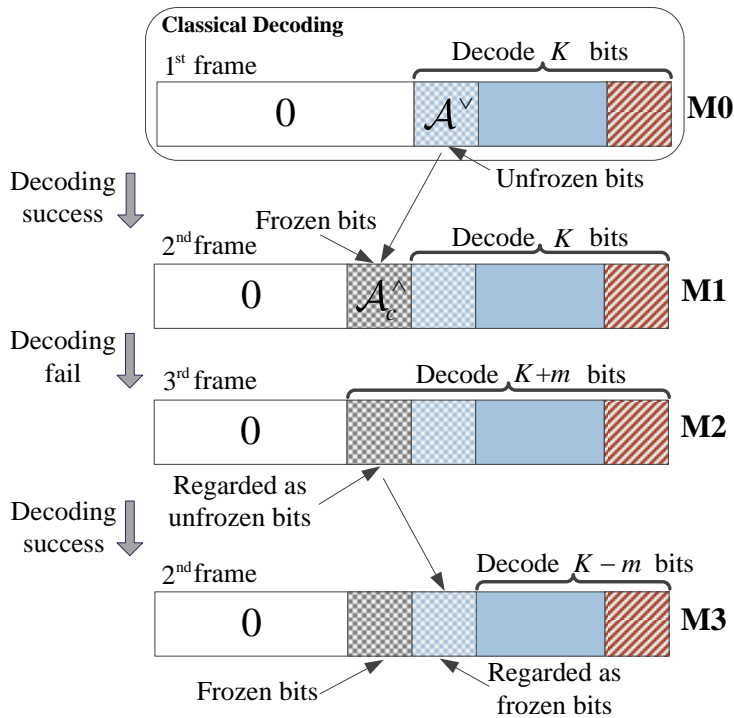


Figure 5.2 The classical and the inter-frame polar decoding examples.

instead of all zeros. Like this, if the decoding of the second frame succeeds, the decoding of the third frame will continue to use **M1**. Otherwise, the third frame will go to decoding mode **M2**, where  $K + m$  bits will be decoded as unfrozen bits since the values of MRFBs cannot be obtained from the preceding frame. If the decoding of the third frame succeeds, its MRFBs will assist a re-decoding of the decoding failed second frame using decoding mode **M3**. As **M3** assigns the values of these MRFBs to those of MUUBs in the second frame, the re-decoding only needs to decode  $K - m$  unfrozen bits, which increases the successful decoding probability compared to **M1**. In this way, the frames help each other in decoding. The probability that the decoding of two adjacent frames both fail due to noise is low. Thus, the overall probability of decoding success will be improved.

In the proposed inter-frame polar coding scheme, the frozen bits are dynamic in two ways: first, their values are not all fixed to predefined values that are known to the decoder, and second, number changes in different decoding modes. It should be noted that the proposed scheme is different from the information re-transmission scenario in which all or part of a frame is re-transmitted until a successful data delivery is acknowledged. In the proposed scheme, the MUUB values of frame  $i - 1$  are assigned to the MRFBs of frame  $i$ . Therefore, the code rate remains unchanged. On the other hand, the information re-transmission scheme inevitably reduces the code rate in order to re-transmit the same frame.

### 5.3 Complexity analysis

The proposed IFA-SCL decoding scheme does not modify the underlying CA-SCL decoding processes. Therefore, different decoding modes use the same CA-SCL decoder. Considering  $M$  quantization bits for channel and internal LLR values, the memory requirement of the CA-SCL decoder with list size  $L$  can be approximated<sup>1</sup> as [136]

$$M(N + L(N - 1)) + L(2N - 1) \text{ [bits]}. \quad (5.5)$$

The IFA-SCL decoder requires additional  $MN + 3m$  bits to store  $N$  channel LLR values of a failed decoded frame,  $m$  bit estimations of a failed decoded frame and at most  $2m$  bit estimations of a successfully decoded frame according to Algorithm 5. Thus, the memory required by the proposed IFA-SCL decoding scheme is

$$M(2N + L(N - 1)) + L(2N - 1) + 3m \text{ [bits]}. \quad (5.6)$$

It should be noted that only a secondary decoding attempt in mode **M3** leads to extra computational complexity in the proposed IFA-SCL decoding scheme which makes the total computational complexity of decoding a frame almost twice as that of the classical scheme. However, as will be shown in Section 5.4, the decoding in mode **M3** occurs with a small probability for good channel conditions or large list sizes. This results in a small increment in the average computational complexity for the proposed IFA-SCL decoding scheme, in comparison with that of the classical CA-SCL decoding.

### 5.4 Simulation results

To verify the effectiveness of the proposed strategy, the performance of the inter-frame polar coding scheme is evaluated. Additive white Gaussian noise (AWGN) channel model is assumed and binary phase-shift keying (BPSK) modulation is adopted. For the simulation results in this section, the relative reliability sequence in the 5G standard [137] is used for the polar code of length  $N = 1024$ , and for the polar code of length  $N = 2048$ , the relative reliability vector is generated by the method in [78], using a design signal-to-noise ratio of 2 dB. A CRC of length  $r = 16$  in the 5G standard with generator polynomial  $D^{16} + D^{12} + D^5 + 1$  is used for CA-SCL decoding and the rate of the code is defined as  $R = (K - r) / N$ .

Figure 5.3 shows the BLER and bit error rate (BER) of polar codes of different code parameters with respect to energy per bit to noise power spectral density ratio ( $E_b/N_0$ ). It can be seen that the proposed inter-frame polar coding scheme with IFA-SCL decoding algorithm brings a performance gain of 0.28 dB at a BLER of  $10^{-4}$  in comparison with the classical polar coding scheme with CA-SCL decoding when  $N = 1024$ ,  $R = 1/2$ ,  $L = 16$ , and  $m = 40$ . It is worth mentioning that the

<sup>1</sup>A small memory is required to store path metrics which is neglected here.

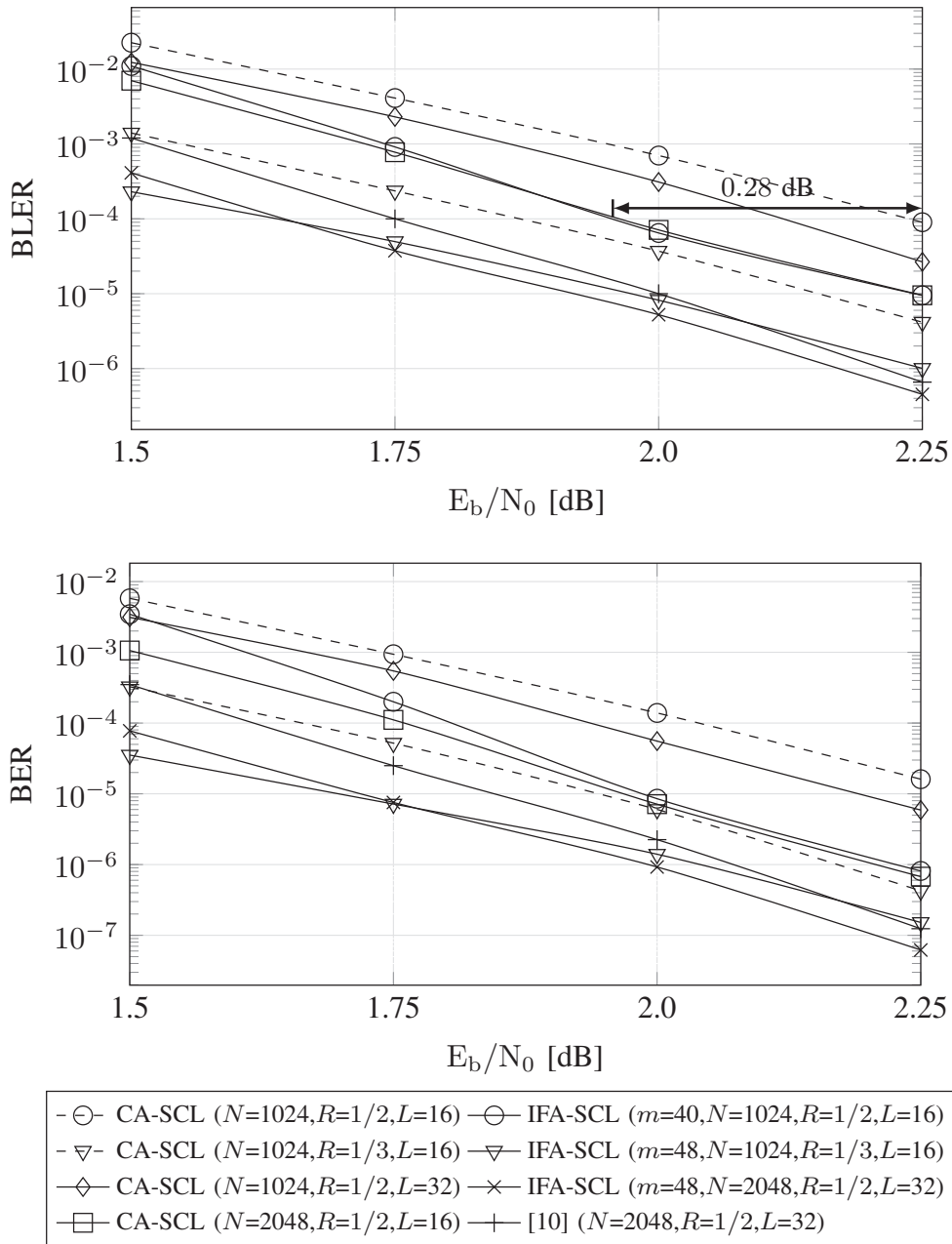


Figure 5.3 BLER (top) and BER (bottom) comparisons between the proposed polar coding scheme, the classical polar coding scheme, and star polar subcodes of [134], with different decoding parameters using a CRC of length 16.

BLER performance of the proposed inter-frame polar coding scheme with  $N = 1024$ ,  $R = 1/2$ ,  $L = 16$ , and  $m = 40$  is 0.15 dB better than that of the classical polar coding scheme of the same code length and rate with  $L = 32$ , and it is almost similar to that of the classical polar coding scheme of the same code rate and list size with  $N = 2048$ , at a target BLER of  $10^{-4}$ . This result is particularly important since considering

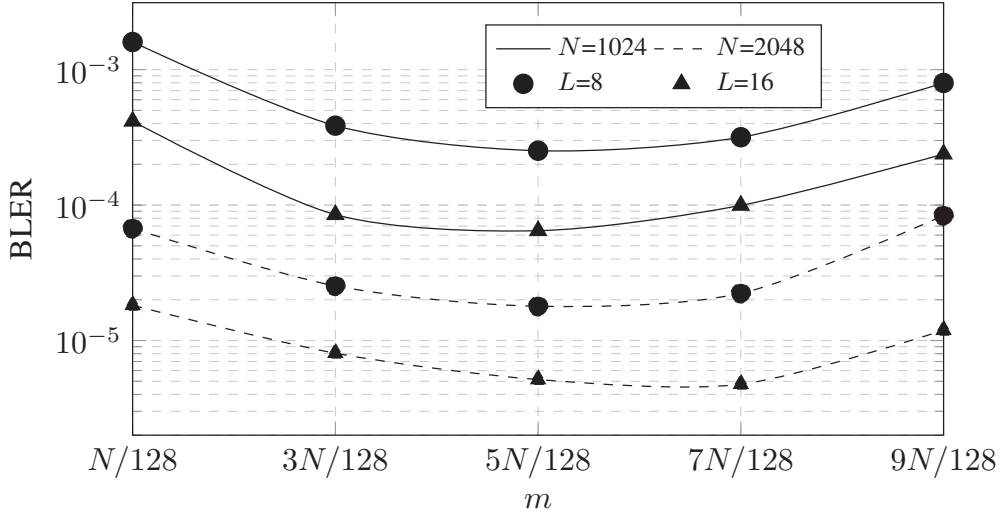


Figure 5.4 Error-correction performance of the proposed polar coding scheme with different values of  $m$  when  $R = 1/2$ , CRC length is 16, and  $E_b/N_0 = 2.0$  dB.

$M = 6$ , the memory requirement of the proposed scheme with  $N = 1024$ ,  $R = 1/2$ ,  $L = 16$ , and  $m = 40$  is 143368 bits, while it is 268064 bits for the classical polar coding scheme with  $N = 1024$  and  $L = 32$ , and 274320 bits for the classical polar coding scheme with  $N = 2048$  and  $L = 16$ . The BLER gain of the proposed scheme in comparison with the classical scheme is also seen when  $R = 1/3$ . Figure 5.3 also provides the BLER and BER of star polar subcodes in [134] and it can be seen that the proposed IFA-SCL decoding scheme provides 0.1 dB performance gain with respect to star polar subcodes when  $N = 2048$ ,  $R = 1/2$ ,  $L = 32$ , and  $m = 48$ , at a target BLER of  $10^{-4}$ . A similar trend can also be observed when comparing BER of different schemes in Figure 5.3.

Figure 5.4 displays the effect of the value of  $m$  on error-correction performance of the proposed inter-frame polar coding scheme at  $E_b/N_0 = 2.0$  dB and  $R = 1/2$ , where two code lengths of 1024 and 2048, and two list sizes of 8 and 16 are considered. Empirically, it can be seen that in all cases, there is a specific value of  $m$  with which the best BLER performance can be achieved. This is due to the fact that for small values of  $m$ , there are not enough additional frozen bits to help the re-decoding process of a failed decoded frame in mode **M3**, while for large values of  $m$ , too many frozen bits are considered as unfrozen to decode a frame in mode **M2**.

Table 5.1 reports the average and the maximum computational complexity of the proposed IFA-SCL decoding scheme in comparison with that of the classical CA-SCL decoding scheme in terms of the number of arithmetic operations (2.37) and (2.38) performed by each scheme<sup>2</sup>. The number of arithmetic operations is calculated by decoding  $10^6$  frames with  $N = 1024$ ,  $R = 1/2$ ,  $L = 16$ ,  $r = 16$ , and  $m = 40$  at different  $E_b/N_0$  values. It can be seen that while the proposed scheme

<sup>2</sup>For the classical scheme, the average and the maximum number of arithmetic operations are equivalent.



Table 5.1 Average computational complexity comparison in terms of the number of arithmetic operations performed

$E_b/N_0$	1.5 dB	1.75 dB	2.0 dB	2.25 dB
$\mathcal{C}_{\text{classical}}$	126882	126882	126882	126882
$\mathcal{C}_{\text{proposed}}^{\text{average}}$	128920	127340	126986	126890
$\mathcal{C}_{\text{proposed}}^{\text{max}}$	251818	251818	251818	251818

has a maximum computational complexity  $\mathcal{C}_{\text{proposed}}^{\text{max}}$  which is about twice as that of the classical CA-SCL decoding scheme  $\mathcal{C}_{\text{classical}}$ , it has an average computational complexity  $\mathcal{C}_{\text{proposed}}^{\text{average}}$  which is only up to 2% higher than that of the classical CA-SCL decoding scheme. It is worth mentioning that the average computational complexity gap between the proposed and the classical decoding schemes reduces as the  $E_b/N_0$  value increases.

## 5.5 Conclusion

An inter-frame polar coding scheme with dynamic frozen bits is proposed to improve the error-correction performance of polar codes in this chapter. In the proposed decoding scheme, consecutive frames share information to help each other when a decoding operation fails. We showed that for a polar code of length 1024 and rate 1/2, the proposed inter-frame polar coding scheme can provide an error-correction performance gain of 0.28 dB over the classical scheme at the block error rate of  $10^{-4}$ . Moreover, we showed that this error-correction performance advantage is achieved with negligible increment in memory requirements and average computational complexity.

---

---

**FAST SUCCESSIVE-CANCELLATION DECODER OF  
POLAR CODES**

---

---

This chapter is adapted from: H. Zheng, S. A. Hashemi, A. Balatsoukas-Stimming, Z. Cao, A. M. J. Koonen, J. Cioffi, A. Goldsmith, "Fast Successive-Cancellation Decoding of Polar Codes", IEEE Transactions on Communications, major revision.

As mentioned in Chapter 2, Fast SC decoding has been a simple and efficient method to meet the demand of low latency in Challenge 3. It overcomes the latency caused by the serial nature of the SC decoding by identifying new nodes in the upper levels of the SC decoding tree and implementing their fast parallel decoders. In this chapter, we first present a novel sequence repetition node corresponding to a particular class of bit sequences. Most existing special node types are special cases of the proposed sequence repetition node. Then, a fast parallel decoder is proposed for this class of node. To further speed up the decoding process of general nodes outside this class, a threshold-based hard-decision-aided scheme is introduced. The threshold value that guarantees a given error-correction performance in the proposed scheme is derived theoretically. Analysis and hardware implementation results on a polar code of length 1024 with code rates  $1/4$ ,  $1/2$ , and  $3/4$  show that our proposed algorithm reduces the required clock cycles by up to 8%, and leads to a 10% improvement in the maximum operating frequency compared to state-of-the-art decoders without tangibly altering the error-correction performance. In addition, using the proposed threshold-based hard-decision-aided scheme, the decoding latency can be further reduced by 57% at  $E_b/N_0 = 5.0$  dB.

## 6.1 Introduction

Although SC decoding provides a low-complexity capacity-achieving solution for polar codes with long block length, its sequential bit-by-bit decoding nature leads to high decoding latency, which constrains its application in low-latency communication scenarios such as the ultra-reliable low-latency communication (URLLC) [138] scheme of 5G and indoor short reach high-speed optical wireless communication [25]. Therefore, the design of fast SC-based decoding algorithms for polar codes with low decoding latency has received a lot of attention [139].

A look-ahead technique was adopted to speed up SC decoding in [140–142] by pre-computing all the possible likelihoods of the bits that have not been decoded yet, and selecting the appropriate likelihood once the corresponding bit is estimated. Us-

ing the binary tree representation of SC decoding of polar codes, instead of working at the bit-level that corresponds to the leaf nodes of the SC decoding tree, parallel multi-bit decision is performed at the intermediate nodes of the SC decoding tree. An exhaustive-search decoding algorithm is used in [143, 144, 119, 145] to make multi-bit decisions and to avoid the latency caused by the traversal of the SC decoding tree to compute the intermediate likelihoods. However, due to the high complexity of the exhaustive search, this method is generally only suitable for nodes that represent codes of very short lengths. In Section 2.2.3, the fast simplified SC decoding has been elaborated. Instead of making exhaustive-search at every node, it only makes decision when some special node types are encountered in decoding. These special node types have special frozen bit patterns which can be exploited to design simple and efficient parallel decoders. However, many node types has been proposed and employed in fast SC decoding, and each class of node requires the design of a separate decoder, which inevitably increases the implementation complexity. In addition, as shown in this chapter, the achievable parallelism in decoding can be further increased without degrading the error-correction performance.

For general nodes that do not fall in one of the above node categories, [146] proposed a hard-decision scheme based on node error probability. Specifically, in that work it was shown that extra latency reduction can be achieved when the communications channel has low noise. However, the hard-decision threshold is calculated empirically rather than for a desired error-correction performance. In [147], a hypothesis-testing-based strategy is designed to select reliable unstructured nodes for hard decision. However, additional operations are required to be performed to calculate the decision rule, thus, incurring extra decoding latency. For all the existing hard-decision schemes, a threshold comparison operation is required each time a general constituent code is encountered in the course of the SC decoding algorithm.

In this chapter, a fast SC decoding algorithm with a higher degree of parallelism than the state of the art is proposed. First, a class of *sequence repetition* (SR) nodes is proposed which provides a unified description of most of the existing special nodes. This class of nodes is typically found at a higher level and has a higher frequency of occurrence in the decoding tree than other existing special nodes. 5G polar codes with different code lengths and code rates can be represented using only SR nodes. Utilizing this class of nodes, a simple and efficient fast simplified SC decoding algorithm called the *SR node-based fast SC* (SRFSC) decoding algorithm is proposed, which achieves a high degree of parallelism without degrading the error-correction performance. Employing only SR nodes, the SRFSC decoder requires a lower number of time steps than most existing works. Moreover, SR nodes can be implemented together with other operation mergers, such as node-branch and branch mergers, to outperform the state-of-the-art decoders in terms of the required number of time steps. In addition, if a realistic hardware implementation is considered, the proposed SRFSC decoder provides up to 8% reduction in terms of the required number of clock cycles and 10% improvement in terms of the maximum operating frequency

with respect to the state-of-the-art decoders on a polar codes of length 1024 with code rates 1/4, 1/2, and 3/4.

Second, a threshold-based hard-decision-aided (TA) scheme is proposed to speed up the decoding of the nodes that are not SR nodes for a binary additive white Gaussian noise (BAWGN) channel. Consequently, a TA-SRFSC decoding algorithm is proposed that adopts a simpler threshold for hard-decision than that in [146]. The effect of the defined threshold on the error-correction performance of the proposed TA-SRFSC decoding algorithm is analyzed. Moreover, a systematic way to derive the threshold value for a desired upper bound for its block error rate (BLER) is determined. Performance results show that, with the help of the proposed TA scheme, the decoding latency of SRFSC decoding can be further reduced by 57% at  $E_b/N_0 = 5$  dB on a polar code of length 1024 and rate 1/2. In addition, a multi-stage decoding strategy is introduced to mitigate the possible error-correction performance loss of TA-SRFSC decoding with respect to SC decoding, while achieving average decoding latency comparable to TA-SRFSC decoding.

The rest of this chapter is organized as follows. Section 6.2 gives the binary tree representation of fast SC decoding. In Section 6.3, the SRFSC decoding algorithm is introduced. With the help of the proposed TA scheme, the TA-SRFSC decoding is presented in Section 6.4. Section 6.5 analyzes the decoding latency and simulation results are shown in Section 6.6. Finally, Section 6.7 gives a summary of the chapter and concluding remarks.

## 6.2 Binary tree representation and fast SC decoding

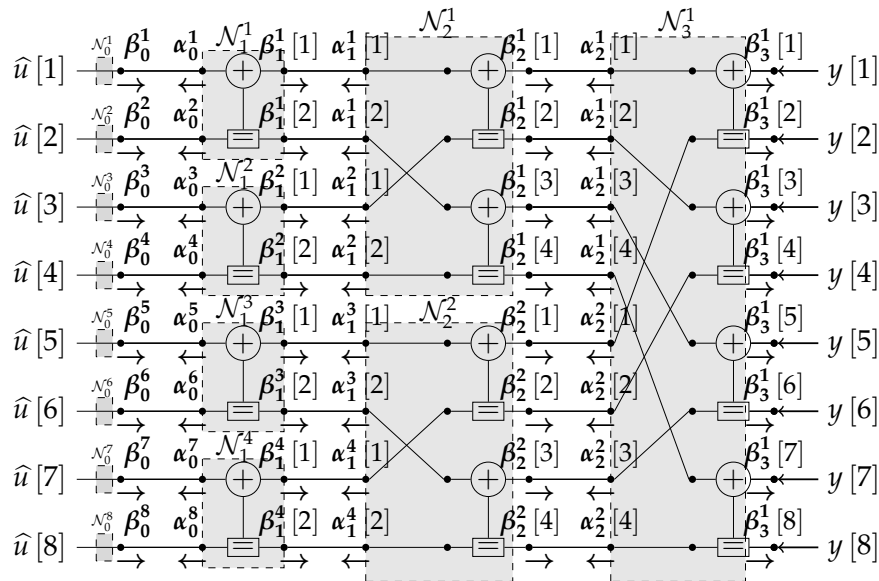


Figure 6.1 SC decoding on the factor graph of a polar code  $N = 8$ .

SC decoding can be illustrated on the factor graph of polar codes as shown in Figure 6.1. For a polar code with length  $N = 2^n$ , the factor graph consists of  $n + 1$  levels and, by grouping all the operations that can be performed in parallel, SC decoding can be represented as the traversal of a binary tree as shown in Figure 6.2. At level  $j$  of the SC decoding tree with  $n + 1$  levels, there are  $2^{n-j}$  nodes ( $0 \leq j \leq n$ ), and the  $i$ -th node at level  $j$  ( $1 \leq i \leq 2^{n-j}$ ) of the SC decoding tree is denoted as  $\mathcal{N}_j^i$ . The left and the right child nodes of  $\mathcal{N}_j^i$  are  $\mathcal{N}_{j-1}^{2i-1}$  and  $\mathcal{N}_{j-1}^{2i}$ , respectively, as illustrated in Figure 6.2. For  $\mathcal{N}_j^i$ ,  $\alpha_j^i[k]$ ,  $1 \leq k \leq 2^j$ , indicates the  $k$ -th input logarithmic likelihood ratio (LLR) value, and  $\beta_j^i[k]$ ,  $1 \leq k \leq 2^j$ , denotes the  $k$ -th output binary hard-valued message. For the AWGN channel, the received vector  $\mathbf{y} = (y[1], y[2], \dots, y[N])$  from the channel can be used to calculate the channel LLR as  $2\mathbf{y}/\sigma^2$ , where  $\sigma^2$  is the variance of the Gaussian noise. SC decoding starts from  $\mathcal{N}_n^1$  by setting  $\alpha_n^1[1:N] = (\alpha_n^1[1], \alpha_n^1[2], \dots, \alpha_n^1[N]) = 2\mathbf{y}/\sigma^2$  and follows a depth-first principle with priority to the left, which has been detailed in Section 2.2.3. After traversing all the nodes in the SC decoding tree,  $\hat{\mathbf{u}}$  contains the decoding result. Thus the latency of SC decoding for a polar code of length  $N$  in terms of the number of time steps can be represented by the number of nodes in the SC decoding tree as [36]

$$\mathcal{T}_{\text{SC}} = 2N - 2. \quad (6.1)$$

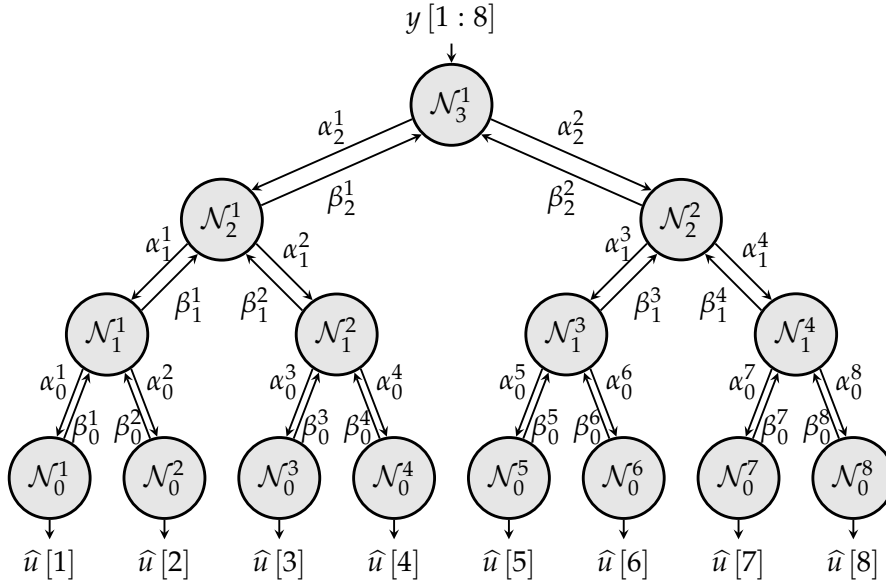


Figure 6.2 Binary tree representation of a SC decoder for a polar code with  $N = 8$ .

The SC decoding has strong data dependencies that limit the amount of parallelism that can be exploited within the algorithm because the estimation of each bit depends on the estimation of all previous bits. This leads to the large latency in (6.1). It was shown in [119] that for a node  $\mathcal{N}_j^i$ ,  $\beta_j^i[1:2^j] = (\beta_j^i[1], \beta_j^i[2], \dots, \beta_j^i[2^j])$  can

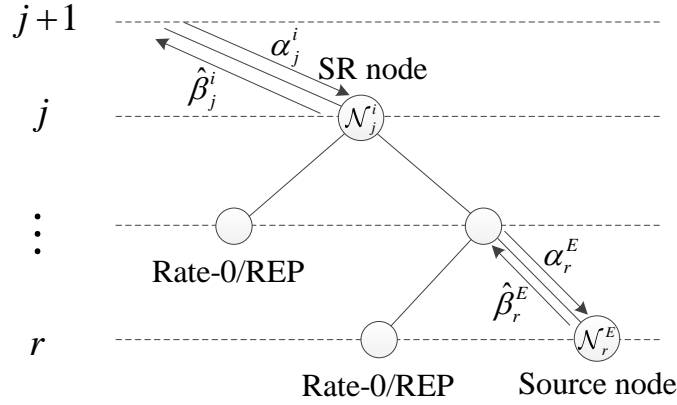


Figure 6.3 General structure of SR Node.

be estimated without traversing the decoding tree as

$$\hat{\beta}_j^i [1 : 2^j] = \arg \max_{\beta_j^i [1:2^j] \in \mathcal{C}_j^i} \sum_{k=1}^{2^j} (-1)^{\beta_j^i[k]} \alpha_j^i [k], \quad (6.2)$$

where  $\mathcal{C}_j^i$  is the set of all the codewords associated with node  $\mathcal{N}_j^i$ . Multibit decoding can be performed directly in an intermediate level instead of bit-by-bit sequential decoding at level 0, in order to traverse fewer nodes in the SC decoding tree and consequently, to reduce the latency caused by data computation and exchange. However, the evaluation of (6.2) generally requires exhaustive search over all the codewords in the set  $\mathcal{C}_j^i$  which is computationally intensive in practice. To avoid exhaustive search, different fast decoding nodes have been proposed as introduced in Section 2.2.3. Each node type has an efficient fast decoding algorithm by exploiting its corresponding special frozen bit pattern, resulting in savings in terms of latency.

## 6.3 Fast SC decoding with sequence repetition nodes

This section introduces a class of nodes that is at a higher level of the SC decoding tree and shows how parallel decoding at this class of nodes can be exploited to achieve significant latency savings in comparison with the state of the art.

### 6.3.1 Sequence repetition (SR) node

Let  $\mathcal{N}_j^i$  be a node at level  $j$  of the tree representation of SC decoding as shown in

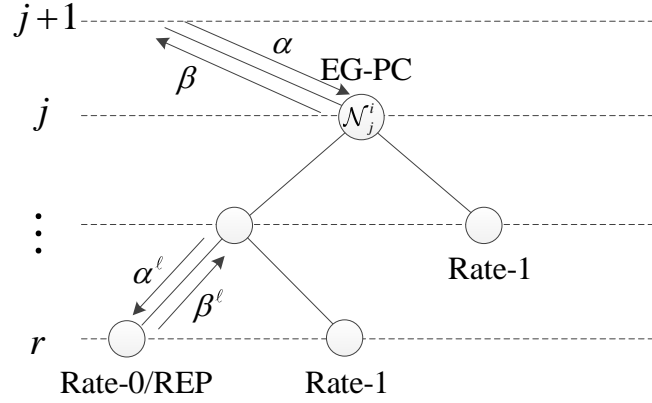


Figure 6.4 General structure of Extended G-PC Node.

Figure 6.2. An SR node is any node at stage  $j$  whose descendants are all either Rate-0 or REP nodes, except the rightmost one at a certain stage  $r$ ,  $0 \leq r \leq j$ , that is a generic node of rate  $C$ . The structure of an SR node is depicted in Figure 6.3. The rightmost node  $\mathcal{N}_r^{i \times 2^{j-r}}$  at stage  $r$  is denoted as the source node of the SR node  $\mathcal{N}_j^i$ . Let  $E = i \times 2^{j-r}$  so the source node can be denoted as  $\mathcal{N}_r^E$ .

An SR node can be represented by three parameters as  $\text{SR}(v, \text{SNT}, r)$ , where  $r$  is the level of the SC decoding tree in which  $\mathcal{N}_r^E$  is located, SNT is the source node type, and  $v = (v[j], v[j-1], \dots, v[r+1])$  is a vector of length  $(j-r)$  such that for the left child node of the parent node of  $\mathcal{N}_r^E$  at level  $k$ ,  $r < k \leq j$ ,  $v[k]$  is calculated as

$$v[k] = \begin{cases} 0, & \text{if the left child node is a Rate-0 node,} \\ 1, & \text{if the left child node is a REP node.} \end{cases} \quad (6.3)$$

Note that when  $r = j$ ,  $\mathcal{N}_j^i$  is a source node and thus  $v$  is an empty vector denoted as  $v = \emptyset$ .

### 6.3.2 Source node

To define the source node type, an extended class of G-PC (EG-PC) nodes is first introduced. The structure of the EG-PC node is depicted in Figure 6.4. The EG-PC node is different from the G-PC node in its leftmost descendant node that can be either a Rate-0 or a REP node. The bits in an EG-PC node satisfy the following parity check constraint,

$$z = \bigoplus_{m=(k-1)2^{j-r}+1}^{k \times 2^{j-r}} \beta_j^i[m], \quad (6.4)$$

Table 6.1 SR node representation of common node types.

Node Type	SR Node Representation	Length of $v$
Rate-0	$\text{SR}(\emptyset, \text{Rate-0}, j)$	0
REP	$\text{SR}((0, \dots, 0), \text{Rate-1}, 0)$	$j$
SPC	$\text{SR}(\emptyset, \text{EG-PC}, j)$	0
Rate-1	$\text{SR}(\emptyset, \text{Rate-1}, j)$	0
P-01	$\text{SR}((0), \text{Rate-1}, j-1)$	1
P-0SPC	$\text{SR}((0), \text{EG-PC}, j-1)$	1
Type-I	$\text{SR}((0, \dots, 0), \text{Rate-1}, 1)$	$j-1$
Type-II	$\text{SR}((0, \dots, 0), \text{EG-PC}, 2)$	$j-2$
Type-III	$\text{SR}(\emptyset, \text{EG-PC}, j)$	0
Type-IV	$\text{SR}(\emptyset, \text{EG-PC}, j)$	0
Type-V	$\text{SR}((0, \dots, 0, 1), \text{EG-PC}, 2)$	$j-2$
G-PC	$\text{SR}(\emptyset, \text{EG-PC}, j)$	0
G-REP	$\text{SR}((0, \dots, 0), \text{Rate-C}, r)$	$j-r$
REP-SPC	$\text{SR}((1), \text{EG-PC}, 2)$	1
0REPSPC	$\text{SR}((0, 1), \text{EG-PC}, j-2)$	2
001	$\text{SR}((0, 0), \text{Rate-1}, j-2)$	2
REP-REPSPC	$\text{SR}((1, 1), \text{EG-PC}, j-2)$	2
Rate0-ML	$\text{SR}((0, 1, 0), \text{Rate-1}, 0)$	3
REP-Rate1(REP1)	$\text{SR}((1), \text{Rate-1}, j-1)$	1

where  $\oplus$  is the bitwise XOR operation,  $k \in \{1, \dots, 2^r\}$ , and  $z \in \{0, 1\}$  is the parity. Unlike G-PC nodes whose parity is always even ( $z = 0$ ), the EG-PC node can have either even parity ( $z = 0$ ) or odd parity ( $z = 1$ ). The parity of the EG-PC node can be calculated as

$$z = \begin{cases} 0, & \text{if the leftmost node is Rate-0,} \\ h\left(\sum_{k=1}^{2^r} \alpha_k\right), & \text{otherwise,} \end{cases} \quad (6.5)$$

where  $\alpha_k = 2 \tanh^{-1} \left( \prod_{m=(k-1)2^{j-r}+1}^{k2^{j-r}} \tanh\left(\frac{\alpha_j^{[m]}}{2}\right) \right)$ . After computing  $z$ , Wagner decoders [148] can be used to decode the  $2^r$  SPC codes with either even or odd parity constraints. A Wagner decoder performs hard decisions on the bits, and if the parity does not hold, it flips the bit with the lowest absolute LLR value.

SPC, Type-III, Type-IV, and G-PC nodes can be represented as special cases of EG-PC nodes. As a result, most of the common special nodes can be represented as SR nodes, as shown in Table 6.1. Note that the node-branch mergers like P-RSPC and F-Rep, and branch operation mergers such as  $F^{\times 2}$ , G-F,  $G0^{\times 2}$ , F-G0, do not fit into the category of SR nodes. It is worth mentioning that other parameter choices for SR nodes may lead to the discovery of more types of special nodes that can be decoded efficiently, but this exploration is beyond the scope of this work.



### 6.3.3 Repetition sequence

In this subsection, a set of sequences, called repetition sequences, is defined that can be used to calculate the output bit estimates of an SR node based on the output bit estimates of its source node. To derive the repetition sequences,  $v$  is used to generate all the possible sequences that have to be XORed with the output of the source node to generate the output bit estimates of the SR node. Let  $\eta_k$  denote the rightmost bit value of the left child node of the parent node of  $\mathcal{N}_r^E$  at level  $k + 1$ . When  $v[k + 1] = 0$ , the left child node is a Rate-0 node so  $\eta_k = 0$ . When  $v[k + 1] = 1$ , the left child node is a REP node, thus  $\eta_k$  can take the value of either 0 or 1. The number of repetition sequences is dependent on the number of different values that  $\eta_k$  can take. Let  $W_v$  denote the number of '1's in  $v$ . The number of all possible repetition sequences is thus  $2^{W_v}$ . Let  $\mathbf{S} = \{s_1, \dots, s_{2^{W_v}}\}$  denote the set of all possible repetition sequences.

The output bits of SR node  $\beta_j^i[1 : 2^j]$  have the property that their repetition sequence is repeated in blocks of length  $2^{j-r}$ . Let  $\beta_r^E[1 : 2^r]$  denote the output bits of the source node of an SR node  $\mathcal{N}_j^i$ . The output bits for each block of length  $2^{j-r}$  in  $\mathcal{N}_j^i$  with respect to the output bits of its source node can be written as

$$\beta_j^i \left[ (k-1)2^{j-r} + 1 : k2^{j-r} \right] = \beta_r^E[k] \oplus s_l, \quad (6.6)$$

where  $k \in \{1, \dots, 2^r\}$  and  $s_l = \{s_l[1], \dots, s_l[2^{j-r}]\}$  is the  $l$ -th repetition sequence in  $\mathbf{S}$ . To obtain the repetition sequence  $s_l$  and with a slight abuse of terminology and notation for convenience, the Kronecker sum operator  $\boxplus$  is used, which is equivalent to the Kronecker product operator, except that addition in GF(2) is used instead of multiplication. For each set of values that  $\eta_k$ 's can take,  $s_l$  can be calculated as

$$s_l = (\eta_r, 0) \boxplus (\eta_{r+1}, 0) \boxplus \dots \boxplus (\eta_{j-1}, 0). \quad (6.7)$$

**Example 1** (Repetition sequences for SR((1, 1), EG-PC, 2)). Consider an example in which the SR node  $\mathcal{N}_4^1$  is located at level 4 of the decoding tree and its source node  $\mathcal{N}_2^4$  is an EG-PC node located at level 2. Since  $v = (1, 1)$ ,  $W_v = 2$  and  $|\mathbf{S}| = 4$ . For  $\eta_1 \in \{0, 1\}$  and  $\eta_2 \in \{0, 1\}$ ,

$$\begin{aligned} s_1 &= (0, 0) \boxplus (0, 0) = (0, 0, 0, 0), & s_2 &= (1, 0) \boxplus (0, 0) = (1, 1, 0, 0), \\ s_3 &= (0, 0) \boxplus (1, 0) = (1, 0, 1, 0), & s_4 &= (1, 0) \boxplus (1, 0) = (0, 1, 1, 0). \end{aligned}$$

■

For a polar code with a given  $d$ , the locations of SR nodes in the decoding tree are fixed and can be determined off-line. Therefore, the repetition sequences in  $\mathbf{S}$  of all of the SR nodes can be pre-computed and used in the course of decoding.

### 6.3.4 Decoding of SR nodes

To decode SR nodes, the LLR values  $\alpha_{r_l}^E[1 : 2^r]$  of the source node  $\mathcal{N}_r^E$  are calculated based on the LLR values  $\alpha_j^i[1 : 2^j]$  of the SR node  $\mathcal{N}_j^i$  for every repetition sequence  $s_l$  as follows:

**Proposition 6.3.1.** *Let  $\alpha_j^i[1 : 2^j]$  be the LLR values of the SR node  $\mathcal{N}_j^i$  and  $\alpha_{r_l}^E[1 : 2^r]$  be the LLR values of its source node  $\mathcal{N}_r^E$  associated with the  $l$ -th repetition sequence  $s_l$ . For  $k \in \{1, \dots, 2^r\}$  and  $l \in \{1, \dots, 2^{W_v}\}$ ,*

$$\alpha_{r_l}^E[k] = \sum_{m=1}^{2^{j-r}} \alpha_j^i[(k-1)2^{j-r} + m] (-1)^{s_l[m]}. \quad (6.8)$$

*Proof:* See Appendix.

Using (6.6) and (6.8), (6.2) can be written as

$$\begin{aligned} \hat{\beta}_j^i[1 : 2^j] &= \arg \max_{\beta_j^i[1:2^j] \in \mathbb{C}_j^i} \sum_{k=1}^{2^j} (-1)^{\beta_j^i[k]} \alpha_j^i[k] \\ &= \arg \max_{\substack{\beta_r^E[1:2^r] \in \mathbb{C}_r^E \\ s_l \in \mathbb{S}}} \sum_{k=1}^{2^r} (-1)^{\beta_r^E[k]} \sum_{m=1}^{2^{j-r}} \alpha_{j_l}^i[(k-1)2^{j-r} + m] (-1)^{s_l[m]} \\ &= \arg \max_{\substack{\beta_r^E[1:2^r] \in \mathbb{C}_r^E \\ l \in \{1, \dots, |\mathbb{S}|\}}} \sum_{k=1}^{2^r} (-1)^{\beta_r^E[k]} \alpha_{r_l}^E[k]. \end{aligned} \quad (6.9)$$

Thus, the bit estimates of an SR node  $\hat{\beta}_j^i[1 : 2^j]$  can be calculated by finding the bit estimates of its source node  $\beta_r^E[1 : 2^r]$  using (6.9) and the repetition sequences as shown in (6.6).

The decoding algorithm of an SR node  $\mathcal{N}_j^i$  is described in Algorithm 6. It first computes  $\alpha_{r_l}^E$  for  $l \in \{1, \dots, |\mathbb{S}|\}$  and generates  $|\mathbb{S}|$  new paths by extending the decoding path at the  $j - r$  rightmost bits corresponding to  $\eta_r, \eta_{r+1}, \dots, \eta_{j-1}$ . Note that the  $l$ -th path is generated when the repetition sequence is  $s_l$  and  $\alpha_{r_l}^E$ ,  $\hat{\beta}_{r_l}^E$ , and  $\hat{\beta}_{j_l}^i$  are its soft and hard messages. Then, the source node is decoded under the rule of the SC decoding. If the source node is a special node, a hard decision is made directly. Parity check and bit flipping will be performed further using Wagner decoder if the source node is an EG-PC node. Finally, the optimal decoding path index can be selected according to the comparison in (6.10) below and the decoding result is obtained using (6.6).

Based on Algorithm 6, the SR node-based fast SC (SRFSC) decoding algorithm is proposed. It follows the SC decoding algorithm schedule until an SR node is encountered where Algorithm 6 is executed. Note that the  $|\mathbb{S}|$  paths can be

---

**Algorithm 6:** Decoding algorithm of SR node  $\mathcal{N}_j^i$ 


---

**Input:**  $\alpha_j^i [1 : 2^l], \mathbf{S}$ ;  
**Output:**  $\hat{\beta}_j^i [1 : 2^l]$ ;  
 1) Soft message computation  
**for**  $l \in \{1, \dots, |\mathbf{S}|\}$  **do**  
   | Calculate  $\alpha_{r_l}^E$  according to (6.8).  
   **end**  
 2) Decoding of source node  $\mathcal{N}_r^E$   
**for**  $l \in \{1, \dots, |\mathbf{S}|\}$  **do**  
   **if**  $SNT=Rate-C$  **then**  
     | Decode source node  $\mathcal{N}_r^E$  using  $\alpha_{r_l}^E$  and obtain  $\hat{\beta}_{r_l}^E$ .  
   **else**  
     **if**  $SNT=Rate-0$  **then**  
       |  $\hat{\beta}_{r_l}^E [k] = 0, k \in \{1, \dots, 2^r\}$ ,  
     **else** //  $SNT=Rate-1$  or  $SNT=EG-PC$   
       |  $\hat{\beta}_{r_l}^E [k] = h(\alpha_{r_l}^E [k]), k \in \{1, \dots, 2^r\}$ .  
     **end**  
   **end**  
   **if**  $SNT=EG-PC$  **then**  
     | Perform parity check and bit flipping on  $\hat{\beta}_{r_l}^E$  using  $\alpha_{r_l}^E$ .  
   **end**  
**end**  
 3) Comparison and path selection

$$\hat{l} = \arg \max_{l \in \{1, \dots, |\mathbf{S}|\}} \sum_{k=1}^{2^r} |\alpha_{r_l}^E [k]|. \quad (6.10)$$

Return  $\hat{\beta}_{j_l}^i$  to parent node according to (6.6).

---

processed simultaneously and the path selection operation in step 3 of Algorithm 6 can be performed in parallel with the decoding of the source node in step 2 and the following  $g$  function operation. Once the selected index  $\hat{l}$  is obtained, only the  $l$ -th decoding path corresponding to  $s_l$  is retained and the remaining paths are deleted.

## 6.4 Hard-decision-aided fast SC decoding with sequence repetition nodes

In this section, a novel threshold-based hard-decision-aided scheme is proposed, which speeds up the decoding of general nodes with no specific structure in the SRFSC decoding at high signal-to-noise ratios. Consequently, the *threshold-based hard-decision-aided SRFSC* (TA-SRFSC) decoding algorithm is proposed. In addition, a

multi-stage decoding strategy is introduced to mitigate the possible error-correction performance loss of the proposed TA-SRFSC decoding.

#### 6.4.1 Proposed threshold-based hard-decision-aided scheme

For a binary AWGN channel with standard deviation  $\sigma_n$ , it was shown in [79] that, considering all the previous bits are decoded correctly, the LLR value  $a_j^i[k]$ ,  $1 \leq k \leq 2^j$ , input into node  $\mathcal{N}_j^i$  can be approximated as a Gaussian variable using a Gaussian approximation as

$$\alpha_j^i[k] \sim \mathcal{N}\left(M_j^i[k], 2\left|M_j^i[k]\right|\right), \quad (6.11)$$

where  $M_j^i[k]$  is the expectation of  $\alpha_j^i[k]$  [131] such that

$$M_j^i[k] = \begin{cases} m_j^i, & \text{if } \beta_j^i[k] = 0, \\ -m_j^i, & \text{if } \beta_j^i[k] = 1, \end{cases} \quad (6.12)$$

and  $m_j^i$  can be calculated recursively offline assuming the all-zero codeword is transmitted as

$$m_n^1 = 2/\sigma_n^2, \quad m_{j-1}^{2^i-1} = \varphi^{-1}\left(1 - [1 - \varphi(m_j^i)]^2\right), \quad m_{j-1}^{2^i} = 2m_j^i, \quad (6.13)$$

where  $\varphi(\cdot)$  is defined in (4.5). It was shown in [146] that, when the magnitude of the LLR values at a certain node in the SC decoding tree is large enough, the node has enough reliability to perform hard decision directly at the node without tangibly altering the error-correction performance. To determine the reliability of the node, a threshold is defined in [146] as

$$T = c_t \log \frac{1 - \frac{1}{2}\text{erfc}\left(0.5\sqrt{m_j^i}\right)}{\frac{1}{2}\text{erfc}\left(0.5\sqrt{m_j^i}\right)}, \quad (6.14)$$

where  $c_t \geq 1$  is a constant that is selected empirically. The hard-decision estimate of the received LLR values is calculated using

$$\text{HB}_j^i[k] = \begin{cases} 0, & \text{if } \alpha_j^i[k] > T, \\ 1, & \text{if } \alpha_j^i[k] < -T. \end{cases} \quad (6.15)$$

The issue with the method in [146] is that the threshold defined in (6.14) contains complex calculations of complementary error functions  $\text{erfc}(\cdot)$ , making the corresponding calculation inefficient. Moreover, the hard-decision threshold is calculated empirically rather than for a desired error-correction performance and the threshold

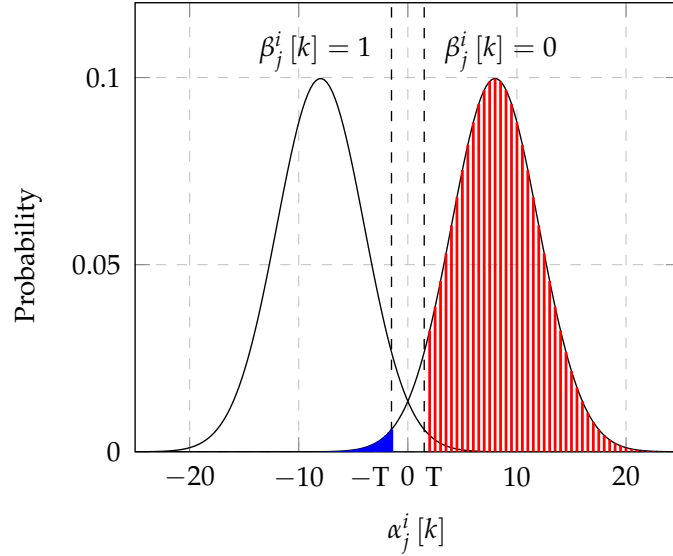


Figure 6.5 Probability distribution of  $\alpha_j^i[k]$  under Gaussian approximation for  $m_j^i = 8$ . The red dashed area represents the probability of correct hard decision and the blue solid area represents the probability of incorrect hard decision when  $\beta_j^i[k] = 0$ .

comparison in (6.15) is performed every time a node with no specific structure is encountered in the SC decoding process.

Figure 6.5 depicts the distribution of  $\alpha_j^i[k]$  of any node  $\mathcal{N}_j^i$  under Gaussian approximation. The red area in Figure 6.5 represents the approximate probability of correct hard decision  $\tilde{P}_c$  and the blue area represents the approximate probability of incorrect hard decision  $\tilde{P}_e$  when  $\beta_j^i[k] = 0$  such that

$$\tilde{P}_c = Q\left(\frac{T - m_j^i}{\sqrt{2m_j^i}}\right), \quad \tilde{P}_e = Q\left(\frac{T + m_j^i}{\sqrt{2m_j^i}}\right), \quad (6.16)$$

where  $Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{-\frac{t^2}{2}} dt$ . The area between the two dashed lines represents the approximate probability that a hard decision is not performed.

To simplify the calculation of the threshold, we take a different approach than [146] by using the Gaussian distribution of  $\alpha_j^i[k]$  and constraining the approximate probability of error when  $\beta_j^i[k] = 0$  to be

$$\tilde{P}_e = Q\left(\frac{T + m_j^i}{\sqrt{2m_j^i}}\right) < Q(c), \quad (6.17)$$

where  $c$  (and thus  $Q(c)$ ) is a positive constant, whose selection method will be given in Observation 1. This is equivalent to  $T > -m_j^i + c\sqrt{2m_j^i}$ . Therefore, the threshold

is

$$T = \left| -m_j^i + c\sqrt{2m_j^i} \right|, \quad (6.18)$$

where the absolute value ensures  $T$  is positive for all values of  $m_j^i > 0$  with any  $c > 0$ .

Under the Gaussian approximation in (6.11), for a node that undergoes hard decision in (6.15), the proposed threshold leads to a bounded probability of correct hard decision as shown in the following observation. Note that the analytical derivations are accurate under the assumption of Gaussian approximation.

**Observation 1.** *Let  $0.5 < \varepsilon < 1$  and  $c > 0$  be real numbers such that*

$$Q(c) \leq 1 - \sqrt[2^n]{\varepsilon}. \quad (6.19)$$

*Performing hard decision in (6.15) with the threshold in (6.18) on nodes  $\mathcal{N}_j^i$  whose  $m_j^i$  satisfy*

$$m_j^i \geq \frac{1}{2} \left[ c - Q^{-1} \left( \frac{Q(c)}{\frac{1}{\sqrt[2^n]{\varepsilon}} - 1} \right) \right]^2, \quad (6.20)$$

*has a probability of correct hard decision that is lower bounded by  $\sqrt[2^{(n-i)}]{\varepsilon}$ , under the assumption of Gaussian approximation and assuming all prior bits are decoded correctly.*

*Explanation:* See Appendix.

The proposed TA scheme performs hard decision on a node  $\mathcal{N}_j^i$  only if (6.20) is satisfied. Furthermore, a hard decision is performed on a node if all of its input LLR values  $\alpha_j^i[k]$  satisfy (6.15). Otherwise, standard SC decoding is applied on  $\mathcal{N}_j^i$  to obtain the decoding result. Compared to the TA scheme in [146], the threshold values of both methods can be computed off-line. However, the proposed TA scheme in this paper has the following three advantages: 1) The proposed method can provide a higher latency reduction than the best case in [146] as shown in the simulation results; 2) The effect of the proposed threshold values on the error-correction performance is approximately predictable according to Observation 1; 3) According to (6.20), only a fraction of nodes undergo hard decision in the decoding process of the proposed TA scheme, which avoids unnecessary threshold comparisons. To speed up the decoding process, the proposed TA scheme is combined with SRFSC decoding that results in the TA-SRFSC decoding algorithm. In TA-SRFSC decoding, when one of the special nodes considered in SRFSC decoding is encountered, SRFSC decoding is performed and when a general node with no special structure is encountered, the proposed TA scheme is applied. The following observation provides an approximate upper bound on the BLER of the proposed TA-SRFSC decoding.

**Observation 2.** *Let  $\text{BLER}_{\text{TA-SRFSC}}$  and  $\text{BLER}_{\text{SRFSC}}$  denote the BLER of the TA-SRFSC decoding and the SRFSC decoding respectively. We can have the following approximate*

inequality

$$\text{BLER}_{\text{TA-SRFSC}} \lesssim 1 - \varepsilon (1 - \text{BLER}_{\text{SRFSC}}). \quad (6.21)$$

*Explanation:* Note that based on Observation 1, any node  $\mathcal{N}_j^i$  that is decoded using (6.15) has a probability of correct hard decision of approximately greater than or equal to  $2^{(n-j)}\sqrt{\varepsilon}$ . For any node that undergoes the SRFSC decoding, the probability of correct decoding is determined by the error rate of SRFSC decoding. Thus, the probability of correct decoding for TA-SRFSC decoding is approximately greater than or equal to  $\varepsilon (1 - \text{BLER}_{\text{SRFSC}})$ . Consequently,  $\text{BLER}_{\text{TA-SRFSC}} \lesssim 1 - \varepsilon (1 - \text{BLER}_{\text{SRFSC}})$ .

Observation 2 provides a method to derive the threshold value for a desired upper bound of the BLER for TA-SRFSC decoding approximately. In fact, a large threshold value results in a better error-correction performance than a small threshold value at the cost of lower decoding speed. Therefore, a trade-off between the error-correction performance and the decoding speed can be achieved with the proposed TA-SRFSC decoding algorithm.

### 6.4.2 Multi-stage decoding

To mitigate the possible error-correction performance loss of the proposed TA-SRFSC decoding, a multi-stage decoding strategy is adopted in which a maximum of two decoding attempts is conducted. In the first decoding attempt, TA-SRFSC decoding is used. If this decoding fails and if there existed a node that underwent hard decision in this first decoding attempt, then a second decoding attempt using SRFSC decoding is conducted. To determine if the TA-SRFSC decoding failed, a cyclic redundancy check (CRC) is concatenated to the polar code and it is verified after the TA-SRFSC decoding. Additionally, the CRC is verified after the second decoding attempt by the SRFSC decoding to determine if the overall decoding process succeeded.

As shown in the next section, most of the received frames are decoded correctly by the proposed TA-SRFSC decoding in the first decoding attempt. As a result, the average decoding latency of the proposed multi-stage SRFSC decoding is very close to that of the TA-SRFSC decoding with CRC bits. However, the error-correction performance of the proposed multi-stage SRFSC decoding algorithm is slightly worse than that of SRFSC decoding due to the addition of CRC bits. As such, multi-stage SRFSC decoding trades off a slight degradation in error-correction performance to obtain a significant reduction in the average decoding latency. It is worth noting that, in order to improve the error-correction performance of the proposed scheme, the proposed multi-stage decoding strategy can be generalized to have more than two decoding attempts. In this scenario the first two attempts are the same as described above, i.e., TA-SRFSC decoding is used first followed by SRFSC decoding. If the TA-SRFSC decoder does not satisfy the CRC, the SRFSC decoding either should start from the beginning or from the first node which underwent the hard decision. Here the SRFSC decoder starts decoding from the first bit to avoid storing intermediate

LLRs. The third and any subsequent attempts would use increasingly more powerful decoding techniques than the first two, such as SCL decoding.

## 6.5 Decoding latency

In this section, the decoding latency of the proposed fast decoders is analyzed under two cases: 1) with no resource limitation and 2) with hardware resource constraints. The former is to facilitate the comparisons with the works that do not consider hardware implementation constraints [89, 90], and the latter is for comparison with those that do [88, 91, 93].

### 6.5.1 No resource limitation

In this case, the same assumptions as in [36, 89] are used and the decoding latency is measured in terms of the number of required time steps. More specifically: 1) there is no resource limitation so that all the parallelizable instructions are performed in one time step, 2) bit operations are carried out instantaneously, 3) addition/subtraction of real numbers and check-node operation consume one time step, and 4) Wagner decoding can be performed in one time step.

#### SRFSC and TA-SRFSC

For any node  $\mathcal{N}_j^i$  that has no special structure, e.g. a parent node of an SR node, and that satisfies (6.20), the threshold comparison in (6.15) is performed in parallel with the calculation of the LLR values of its left child node. Therefore, the proposed hard decision scheme does not affect the latency requirements for the nodes that undergo hard decision.

The number of time steps required for the decoding of the SR node is calculated according to Algorithm 6. In Step 1, the calculation of the LLR values for the source node requires one time step if  $v \neq \emptyset$ . If  $v = \emptyset$ , then the LLR values of the source node are available immediately. Thus, the required number of time steps for Step 1 is

$$\mathcal{T}_1 = \begin{cases} 0, & \text{if } v = \emptyset, \\ 1, & \text{if } v \neq \emptyset. \end{cases} \quad (6.22)$$

The time step requirement of Step 2 depends on the source node type. If SNT = Rate-C, the time step requirement of Step 2 is the time step requirement of the Rate-C node. If SNT = Rate-0 or Rate-1, then there is no latency overhead in Step 2. If SNT = EG-PC and in accordance with Section 6.3.2,  $z$  can be estimated in two time steps if the leftmost node is a REP node (one time step for performing the check-node operation and one time step for adding the LLR values). Also, Wagner decoding can be performed in parallel with the estimation of  $z$  assuming  $z = 0$  or  $z = 1$ . As such, at most two time steps are required for parity check and bit flipping of the EG-PC



node. The required number of time steps for Step 2 is

$$\mathcal{T}_2 = \begin{cases} 0, & \text{if SNT} = \text{Rate-0 or Rate-1,} \\ 1 \text{ or } 2, & \text{if SNT} = \text{EG-PC,} \\ 2^{r+1} - 2, & \text{if SNT} = \text{Rate-C.} \end{cases} \quad (6.23)$$

Step 3 consumes two time steps using an adder tree and a comparison tree if  $|\mathcal{S}| > 1$  and no time steps otherwise. Thus, the number of required time steps for Step 3 is

$$\mathcal{T}_3 = \begin{cases} 0, & \text{if } |\mathcal{S}| = 1, \\ 2, & \text{if } |\mathcal{S}| > 1. \end{cases} \quad (6.24)$$

Since path selection in Step 3 can be executed in parallel with the decoding of source node in Step 2 and the following  $g$  function calculation, the total number of time steps required to decode an SR node can be expressed as

$$\mathcal{T}_{\text{SR}} = \mathcal{T}_1 + \max(\mathcal{T}_2, \mathcal{T}_3 - 1), \quad (6.25)$$

where  $\mathcal{T}_3 - 1$  indicates that at least one time step in  $\mathcal{T}_3$  can be reduced by parallelizing Step 3 and the  $g$  function calculation. Therefore,  $\mathcal{T}_{\text{SR}}$  is a variable that is dependent on its parameters. However, with a given polar code, the total number of time steps required for the decoding of the polar code using SRFSC decoding is fixed, regardless of the channel conditions.

### Multi-stage SRFSC

Let  $\mathcal{T}_{\text{TA-SRFSC}_{\text{crc}}}$  and  $\mathcal{T}_{\text{SRFSC}_{\text{crc}}}$  denote the average decoding latency of the proposed TA-SRFSC decoding and the SRFSC decoding using CRC bits, respectively. The average decoding latency of multi-stage SRFSC decoding,  $\mathcal{T}_{\text{Multi-stage SRFSC}}$ , is

$$\mathcal{T}_{\text{Multi-stage SRFSC}} = \mathcal{T}_{\text{TA-SRFSC}_{\text{crc}}} + P_{\text{Re-decoding}} \mathcal{T}_{\text{SRFSC}_{\text{crc}}}, \quad (6.26)$$

where  $P_{\text{Re-decoding}}$  indicates the probability that TA-SRFSC decoding fails and there is at least one node that undergoes hard decision. Note that  $P_{\text{Re-decoding}}$  is less than or equal to the probability that the output of TA-SRFSC decoding fails the CRC verification, which can be approximated by  $\text{BLER}_{\text{TA-SRFSC}_{\text{crc}}}$ . The approximation is due to the fact that the undetected error probability of CRC is negligible when its length is long enough [135]. In accordance with Observation 2, the approximate average decoding latency requirement for the proposed multi-stage SRFSC decoding is

$$\begin{aligned} & \mathcal{T}_{\text{Multi-stage SRFSC}} \\ & \lesssim \mathcal{T}_{\text{TA-SRFSC}_{\text{crc}}} + (1 - \varepsilon (1 - \text{BLER}_{\text{SRFSC}_{\text{crc}}})) \mathcal{T}_{\text{SRFSC}_{\text{crc}}}. \end{aligned} \quad (6.27)$$

Since the decoding latency of SRFSC decoding is fixed, the average decoding latency and the worst case decoding latency of SRFSC decoding are equivalent. The worst case decoding latency of the proposed TA-SRFSC decoding can be calculated when none of the nodes in the decoding tree undergo hard decision. This occurs when the channel has a high level of noise. Thus, the worst case decoding latency of the proposed TA-SRFSC decoding is equivalent to the decoding latency of the SRFSC decoding. Moreover, when the channel is too noisy,  $P_{\text{Re-decoding}} \approx 0$ , because almost none of the nodes undergo hard decision. Thus, the worst case decoding latency of the proposed multi-stage SRFSC decoding is equivalent to the worst case decoding latency of TA-SRFSC decoding using a CRC, which is the latency of SRFSC decoding with a CRC.

### 6.5.2 With hardware resource constraints

In this case, the decoding latency is measured in terms of the number of clock cycles in the actual hardware implementation. Contrary to Section 6.5.1, a time step may take more than one clock cycle in a realistic hardware implementation. First, the idea of a semi-parallel decoder in [149] is adopted in which at most  $P$  processing elements can run at the same time. Thus, parallel  $f$  function or  $g$  function operations on more than  $P$  processing elements require more than one clock cycle. Second, the adder tree used for the calculation of LLR values in step 1 of Algorithm 6, and the compare-and-select (CS) tree used by the Wagner decoder to find the index of the least reliable input bit in step 2 of Algorithm 6, both require pipeline registers to reduce the critical path delay and improve the operating frequency and the overall throughput. The addition of these pipeline registers increases the number of decoding clock cycles. The exact number depends on the implementation and the given polar code.

## 6.6 Results and comparison

In this section, the average decoding latency and the error-correction performance of the proposed decoding algorithms are analyzed and compared with state-of-the-art fast SC decoding algorithms. To derive the results, polar codes of length  $N \in \{128, 512, 1024\}$ , which are adopted in the 5G standard [137], are used. To do a fair comparison, for baseline decoding algorithms without hardware implementation, the latency is calculated under the assumptions in Section 6.5.1. Otherwise, the hardware implementation results are reported.

To simulate the effect of  $\varepsilon$  on the error-correction performance and the latency of the proposed decoding algorithms, the assumption in [146] is adopted and three values of  $\varepsilon \in \{0.9, 0.99, 0.999\}$  are selected. In accordance with (6.19),  $c \geq 3.8$  for  $\varepsilon = 0.9$ ,  $c \geq 4.3$  for  $\varepsilon = 0.99$ , and  $c \geq 4.8$  for  $\varepsilon = 0.999$ . According to (6.17), with the increasing of  $c$ ,  $P_e$  decreases. At the same time, fewer nodes undergo hard decision and the decoding latency increases. To get a good trade-off between error-correction performance and latency, we set  $c = 3.8$  for  $\varepsilon = 0.9$ ,  $c = 4.3$  for  $\varepsilon = 0.99$ , and  $c = 4.8$  for  $\varepsilon = 0.999$ . Consequently,  $m_j^i \geq 9.3891$  for  $\varepsilon = 0.9$ ,  $m_j^i \geq 14.7255$  for  $\varepsilon = 0.99$ ,

and  $m_j^i \geq 16.1604$  for  $\varepsilon = 0.999$  in accordance with (6.20). Using these values, the threshold  $T$  defined in (6.18) and the BLER upper bound for the TA-SRFSC decoding in (6.21) can be calculated for different values of  $\varepsilon$ .

Table 6.2 The number of SR nodes with different  $|S|$  and the number of general nodes in 5G polar codes of lengths  $N \in \{128, 512, 1024\}$  and rates  $R \in \{1/4, 1/2, 3/4\}$ .

N	R	SR					Total	General Total
		S						
		1	2	4	8	16		
128	1/4	1	0	2	1	0	4	3
	1/2	4	3	1	0	0	8	7
	3/4	8	2	0	0	0	10	9
512	1/4	12	2	2	1	0	17	16
	1/2	15	5	2	0	1	23	22
	3/4	13	5	1	0	1	20	19
1024	1/4	17	6	2	2	1	28	27
	1/2	25	8	2	3	1	39	38
	3/4	29	8	2	1	0	40	39

Table 6.2 reports the number of SR nodes with different  $|S|$ , the total number of SR nodes, and the total number of general nodes with no special structure, at different code lengths and rates. It can be seen that when the code length is 128, 512, and 1024, the codes with rate 1/2 have the largest proportion of nodes with  $|S| > 1$ , respectively. This in turn results in more latency savings because a higher degree of parallelism can be exploited with these nodes. Thanks to the binary tree structure and the fact that SR nodes are always present at an intermediate level of the decoding tree, the number of traversed general nodes with no special structure equals the total number of SR nodes minus one, since they are the parent nodes of SR nodes. Table 6.3 shows the length of SR nodes with different  $|S|$  at different code lengths when  $R = 1/2$ . The length of the SR nodes in the decoding tree corresponds to the level in the decoding tree that they are located. SR nodes with larger  $|S|$  that are located on a higher level of the decoding tree contribute more in the overall latency reduction.

Table 6.4 reports the number of time steps required to decode polar codes of lengths  $N \in \{128, 512, 1024\}$  and rates  $R \in \{1/4, 1/2, 3/4\}$  with the proposed SRFSC decoding algorithm, and compares it with the required number of time steps of the decoders in [88–91, 93]. Note that the decoder in [93] has the minimum required time steps amongst all the previous works considered in Table 6.4. Three versions of the proposed SRFSC decoder are considered in Table 6.4: the SRFSC decoder that only utilizes SR nodes; the SRFSC' decoder that considers SR nodes

Table 6.3 Node length of SR nodes with different  $|S|$  in 5G polar codes of lengths  $N \in \{128, 512, 1024\}$  and rates  $R = 1/2$ .

$N$	Length	$ S $				
		1	2	4	8	16
128	8	2	2	0	0	0
	16	0	1	1	0	0
	32	2	0	0	0	0
512	8	7	3	0	0	0
	16	4	1	2	0	0
	32	3	1	0	0	0
	64	1	0	0	0	0
	128	0	0	0	0	1
1024	8	10	6	0	0	0
	16	7	1	2	0	0
	32	4	0	0	3	0
	64	2	1	0	0	1
	128	2	0	0	0	0

and P-0SPC nodes that is adopted in all other baseline decoders in Table 6.4; and the SRFSC'' decoder that uses SR nodes, P-0SPC nodes, and the operation mergers  $F^{\times 2}$  and G-F. It can be seen that the SRFSC'' decoder requires fewer time steps with respect to other decoders, except for the case of  $N = 128$  and  $R = 3/4$ , which has a frequent occurrence of REP-SPC nodes that provides an advantage for [93]. This is due to the fact that the REP-SPC decoder in other works decodes REP and SPC nodes in parallel, consuming only a single time step, while two time steps are needed to decode the REP-SPC node in the proposed SRFSC decoder.

Table 6.5 compares the required number of clock cycles and the maximum operating frequency in the hardware implementation of different decoders for a polar code with  $N = 1024$ ,  $R \in \{1/2, 1/4, 3/4\}$ , and  $P = 64$  for the proposed SRFSC decoding algorithm (taken from [150]) and the decoders in [88, 91–93]. All FPGA results are for an Altera Stratix IV EP4SGX530KH40C2 FPGA device and all ASIC results use TSMC 65nm CMOS technology. The required number of clock cycles for the proposed SRFSC decoder can be further reduced if we consider node-branch merging and branch operation merging similar to [93]. We have verified that the introduction of P-RSPC node to the proposed SRFSC decoder lengthens the critical path of the decoder, but the merging of branch operations,  $F^{\times 2}$  and G-F, does not have an effect on the operating frequency. Thus, an enhanced SRFSC decoder with merging branch operations  $F^{\times 2}$  and G-F is implemented and denoted as SRFSC\* in Table 6.5. It can be seen that the SRFSC\* decoder requires a smaller number of clock cycles compared to other decoders and the reduction with respect to [93] is 8%, 4%,

Table 6.4 Number of time steps for different fast SC decoding algorithms of polar codes of lengths  $N \in \{128, 512, 1024\}$  and rates  $R \in \{1/4, 1/2, 3/4\}$ .

$N$	$R$	[88]	[91]	[89]	[90]	[93]	SRFSC	SRFSC'	SRFSC''
128	1/4	25	23	24	23	10	13	12	10
	1/2	33	31	24	24	21	25	24	20
	3/4	22	22	22	22	16	29	25	20
512	1/4	73	70	63	63	42	57	52	40
	1/2	87	83	77	77	56	72	66	52
	3/4	79	73	64	64	50	63	56	45
1024	1/4	122	115	110	108	68	92	85	66
	1/2	156	144	138	138	94	127	115	90
	3/4	138	132	116	116	89	123	111	86

and 8% at  $R \in \{1/4, 1/2, 3/4\}$ , respectively. Moreover, the SRFSC decoder and the SRFSC\* decoder both have the highest maximum operating frequency  $f_{\max}$  when implemented on the FPGA. Compared with the works that reported  $f_{\max}$  results for FPGA implementations, our decoders achieve an improvement of at least 10%.

Figure 6.6 shows the BLER performance of different decoding algorithms when  $N = 1024$  and  $R = 1/2$ , for different values of energy per bit to noise power spectral density ratio ( $E_b/N_0$ ). For each value of  $\varepsilon$ , the BLER of TA-SRFSC decoding is depicted together with the upper bound calculated by Observation 2. It can be seen that the introduction of the TA scheme results in BLER performance loss for the proposed TA-SRFSC decoding with respect to SC and SRFSC decoding, especially at higher values of  $E_b/N_0$ . Moreover, the simulations confirm that the BLER curves of TA-SRFSC decoding fall below their respective upper bounds. It can also be seen that, as the  $E_b/N_0$  value increases beyond a specific point, the BLER performance of the TA-SRFSC decoding degrades. This is because of the difference in the performance of hard-decision and soft-decision decoding. In accordance with (6.20), more nodes undergo hard decision decoding for larger values of  $E_b/N_0$ . Therefore, while the channel conditions improve, the hard decision decoding introduces errors that reduce the error-correction performance gain associated with these large  $E_b/N_0$  values. As a result, the BLER performance degrades after a certain value of  $E_b/N_0$ . This phenomenon exists as long as there are nodes that can undergo hard decision decoding. After all the nodes are decoded using hard decision, the BLER performance improves again as  $E_b/N_0$  increases.

The proposed multi-stage SRFSC decoder is implemented using three different CRC lengths that are adopted in 5G to identify whether the decoding succeeded or failed: the CRC of length 6 (CRC6) with generator polynomial  $D^6 + D^5 + 1$ , the CRC of length 11 (CRC11) with generator polynomial  $D^{11} + D^{10} + D^9 + D^5 + 1$ , and the CRC of length 16 (CRC16) with generator polynomial  $D^{16} + D^{12} + D^5 + 1$ . For all

Table 6.5 Required number of clock cycles and maximum operating frequency for different decoding algorithms of polar codes of length  $N = 1024$  with rate  $R = \{1/4, 1/2, 3/4\}$  and  $P = 64$ .

	clock cycles			$f_{\max}$ (MHz)	
	R			FPGA	ASIC
	1/4	1/2	3/4		
SRFSC	186	222	200	109.6	—
SRFSC*	155	191	166	109.6	—
[93]	168	198	181	—	430
[92]	189	214	185	89.6	420
[91]	221	252	216	—	450
[88]	225	270	225	99.8	450

values of  $\varepsilon$ , the proposed multi-stage SRFSC decoding results in almost the same BLER performance. Therefore, only the curve with  $\varepsilon = 0.9$  is plotted in Fig. 6.6 for the multi-stage SRFSC decoding. CRC16 provides a better error-correction performance compared to CRC6 and CRC11, especially at high values of  $E_b/N_0$ , due to high undetected probability of error for short CRC lengths. Hence, CRC16 is selected for the polar code of length  $N = 1024$  in this paper. It can be seen that the multi-stage SRFSC decoding with CRC16 demonstrates a slight performance loss compared to the conventional SC and SRFSC decoders as a result of adding extra CRC bits. For polar codes of other lengths and rates, a similar trend can also be observed when comparing the BLER performance of different schemes.

Figure 6.7 presents the average decoding latency in terms of the required number of time steps for the proposed TA-SRFSC decoding and multi-stage decoding with CRC16. In particular, it compares them with the latency of SRFSC decoding and the decoder in [146] at different values of  $E_b/N_0$  when  $N = 1024$  and  $R = 1/2$ . It can be seen that the required number of time steps for the proposed TA-SRFSC decoding decreases as  $E_b/N_0$  increases and is reduced by 40% for  $\varepsilon = 0.999$ , by 48% for  $\varepsilon = 0.99$ , and by 57% for  $\varepsilon = 0.9$ , compared to SRFSC decoding at  $E_b/N_0 = 5$  dB. In addition, the proposed multi-stage SRFSC decoding reduces the required number of time steps by 37% for  $\varepsilon = 0.999$ , by 46% for  $\varepsilon = 0.99$ , and by 53% for  $\varepsilon = 0.9$ , with respect to SRFSC decoding at  $E_b/N_0 = 5$  dB. The required number of time steps for the proposed multi-stage SRFSC decoding outperforms the method in [146] with  $c_t = 1$  by 19% for  $\varepsilon = 0.9$  at  $E_b/N_0 = 5$  dB while providing a significantly better BLER performance. Figure 6.7 also presents the approximate upper bound derived in (6.27). It can be seen in the figure that the upper bound in (6.27) becomes tighter as  $\varepsilon$  increases.

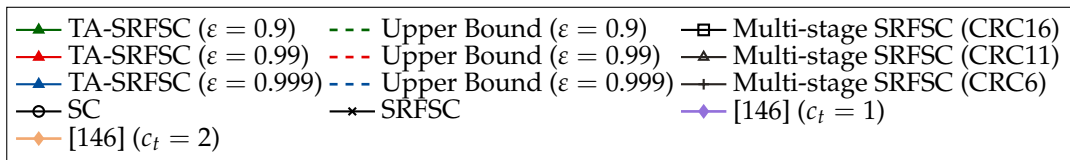
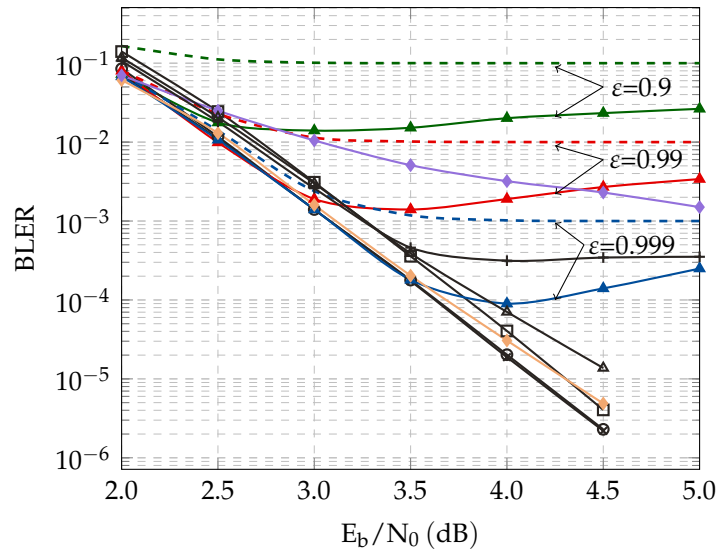


Figure 6.6 BLER performance of different decoding algorithms for the 5G polar code of length  $N = 1024$  and rate  $R = 1/2$ .

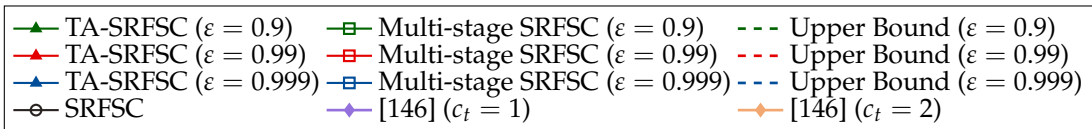
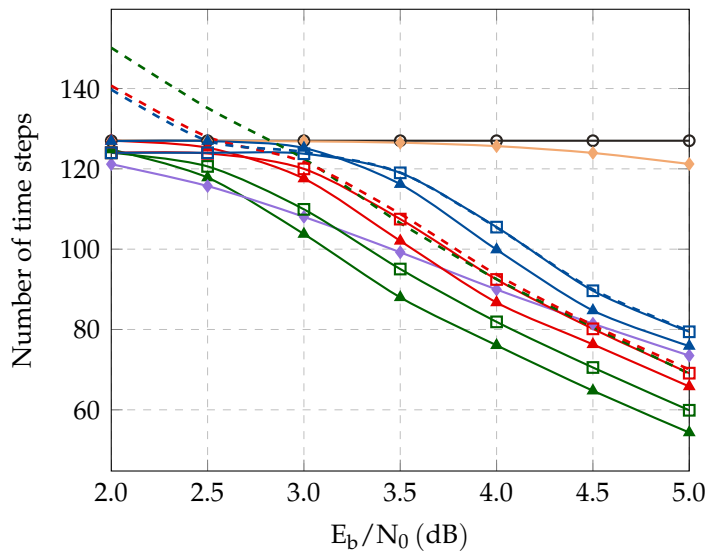


Figure 6.7 Average decoding latency of different decoding algorithms for the 5G polar code of length  $N = 1024$  and rate  $R = 1/2$ .

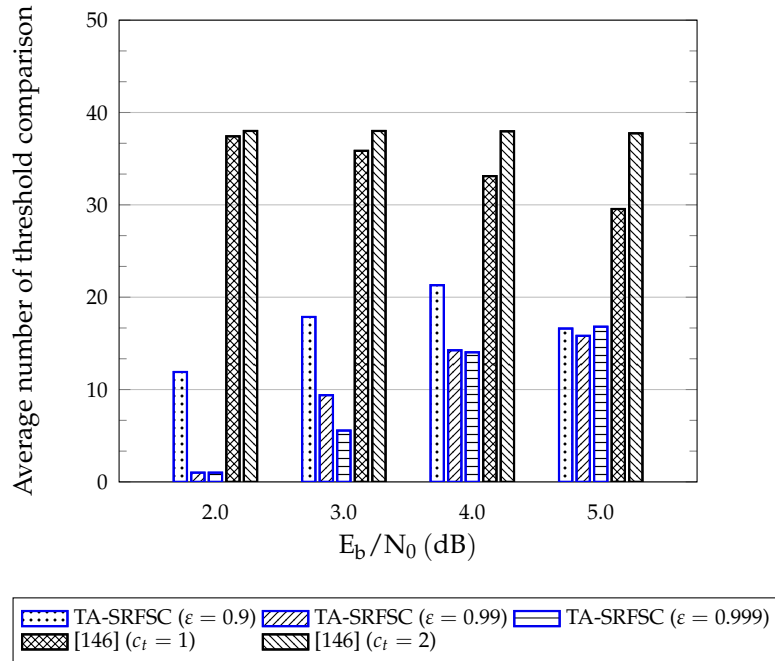


Figure 6.8 Average number of threshold comparisons of the proposed TA-SRFSC decoding in comparison with the hard-decision scheme in [146] for the 5G polar code of length  $N = 1024$  and rate  $R = 1/2$ .

Figure 6.8 compares the average number of threshold comparisons in (6.15) for the proposed TA-SRFSC decoder with  $\epsilon \in \{0.9, 0.99, 0.999\}$ , and the decoder in [146] with  $c_t \in \{1, 2\}$  for the 5G polar code of length  $N = 1024$  and rate  $R = 1/2$ . It can be seen that the proposed TA-SRFSC decoder shows significant benefits with respect to the decoder of [146] in terms of the average number of threshold comparisons. The TA-SRFSC decoder with  $\epsilon = 0.9$  provides at least 36% reduction with respect to [146] with  $c_t = 1$  while having a lower decoding latency. This means the decoder in [146] executes many unnecessary threshold comparison operations, while TA-SRFSC decoding only makes hard decisions when a node satisfies the condition in Equation (6.20).

## 6.7 Conclusion

In this chapter, a new sequence repetition (SR) node is identified in the successive-cancellation (SC) decoding tree of polar codes and an SR node-based fast SC (SRFSC) decoder is proposed. In addition, to speed up the decoding of nodes with no specific structure, the SRFSC decoder is combined with a threshold-based hard-decision-aided (TA) scheme and a multi-stage decoding strategy. We show that this method further reduces the decoding latency at high SNRs. In particular, hardware implementation results on an FPGA for a polar code of length 1024 with code rates  $1/4$ ,  $1/2$ , and  $3/4$  show that the proposed SRFSC decoder with merging branch operations requires up to 8% fewer clock cycles and achieves 10% higher maximum



operating frequency compared to state-of-the-art decoders. In addition, the proposed TA-SRFSC decoding reduces the average decoding latency by 57% with respect to SRFSC decoding at  $E_b/N_0 = 5$  dB on a polar code of length 1024 and rate 1/2. This average latency saving is particularly important in real-time applications such as video. Future work includes the design of a fast SC list decoder using SR nodes.

# **HARDWARE IMPLEMENTATION OF DECODER FOR POLAR CODES**

---

This chapter is adapted from: H. Zheng, A. Balatsoukas-Stimming, Z. Cao, A. M. J. Koonen, “Implementation of a High-Throughput Fast-SSC Polar Decoder with Sequence Repetition Node”, IEEE International Workshop on Signal Processing Systems (SIPS), Portugal, Oct. 2020.

This chapter focuses on the hardware implementation of the SR node-based fast SC (SRFSC) decoder proposed in chapter 6. The implementation results for a polar code with length 1024 and code rate 1/2 show that our implementation has a throughput of 505 Mbps on an Altera Stratix IV FPGA, which is 17.9% higher with respect to the previous work.

## **7.1 Introduction**

A typical fast simplified SC decoder contains three main modules [88]: a memory, an arithmetic logical unit (ALU), and a controller. The memory consists of five separate sub-modules. The channel LLR, internal LLR  $\alpha$ , and estimation  $\beta$  sub-modules feed the ALU. The instruction sub-module stores the operations to be executed and is routed into the controller. Finally, the codeword sub-module stores and outputs the final codeword. The ALU implements the  $f$  function given in (2.39), the  $g$  function given in (2.38), the combining operation given in (2.45), as well as the update rules for various special nodes like the Rep node, SPC node, Rep-SPC node and ML node as shown in Figure 7.1. Finally, the controller tracks which node in the decoding tree is currently being decoded by using a list of instructions that is pre-compiled based on information bit set  $A$  and frozen bit set  $A^c$ .

Chapter 6 proposed a new class of *sequence repetition* (SR) constituent codes, which is a generalization of most existing special nodes. It was also shown that the decoding of SR constituent codes can be highly parallelized to achieve further latency reduction compared to the state of the art without tangibly affecting the error-correcting performance. However, the work of [151] mainly focused on the algorithmic aspects of SR constituent codes and no hardware architecture has been reported in the literature. In this chapter, we describe a hardware architecture for a fast simplified SC decoder that exploits the SR constituent codes described in [151]

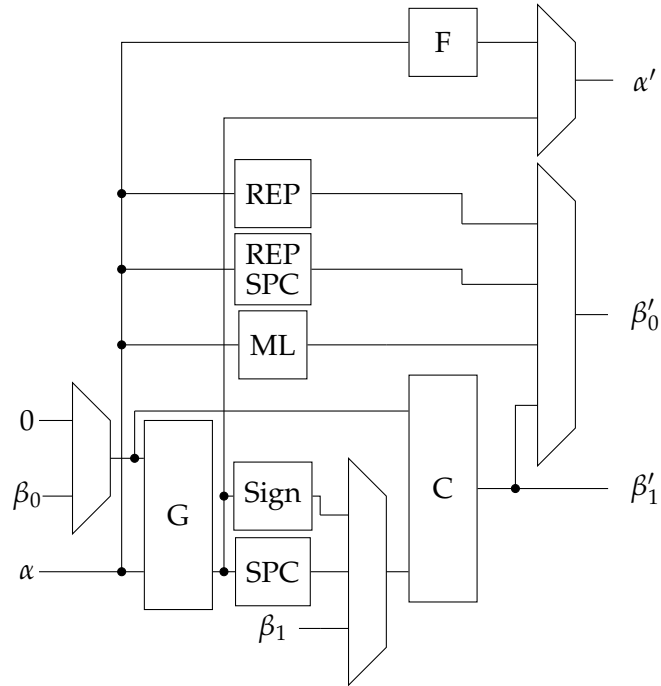


Figure 7.1 Arithmetic logical unit architecture from [88].

and we provide FPGA implementation results. Even though our proposed implementation is not yet highly optimized, it still achieves a 17.9% higher decoding throughput than the state of the art.

The rest of this chapter is organized as follows. Section 7.2 designs the architecture of SRFSC decoder. The FPGA implementation results are shown in Section 7.3. Finally, Section 7.4 gives a summary of the chapter.

## 7.2 Architecture of SRFSC decoder

The top-level architecture of the proposed SRFSC decoder is shown in Fig. 7.2. When decoding starts, the instructions for the polar code that is being decoded are fetched by the controller and the channel LLRs are loaded into memory. The controller decodes the instructions to get the node schedule and updates the decoding stage parameters accordingly. The updates in the controller follow the principle of SC decoding until an instruction corresponding to an SR node is reached, where the SR module is activated to process the LLRs. The estimation results from both the SR module and processing module are routed into the partial sum network (PSN) module, from where the estimated codeword is also output when decoding terminates. In the following, the architecture of the various individual modules is discussed in detail.

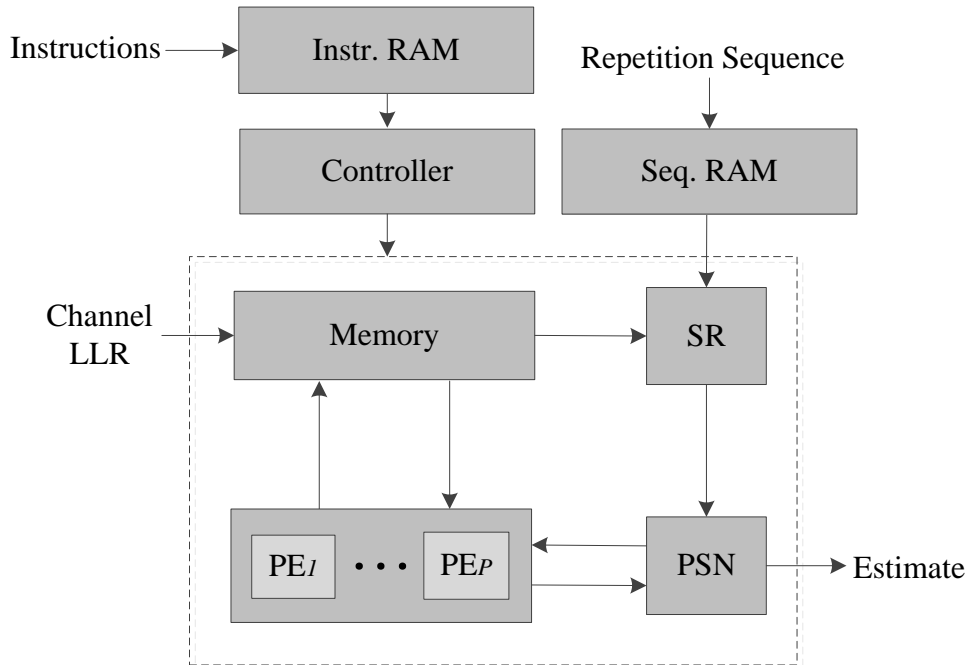


Figure 7.2 Top-level architecture of the proposed SRFSC decoder.

### 7.2.1 Memory, processing, and PSN modules

The architectures of these three modules are identical with those presented in [149] and we thus only describe them on a high level. The memory module stores all soft messages  $\alpha$ . The update of hard estimates  $\beta$  is in the partial sum network (PSN) module. A set of  $P$  processing elements (PEs) is instantiated in the processing module to process up to  $2P$  LLRs in parallel. A PE implements both the  $f$  and the  $g$  function using sign-and-magnitude representation and the appropriate output is selected according to the current decoding stage.

### 7.2.2 Controller module

The operation in the controller module follows the standard SC decoding schedule until an instruction that indicates an SR node is found. When this occurs, the  $2P$  LLRs will be routed to the SR module instead of the processing module to perform the decoding of SR node in Algorithm 6. The required number of clock cycles to decode the SR node by the SR module is pre-calculated and a counter is initialized to this value. All updates in the controller are suspended until the counter reaches zero. Then, the decoding bit index is added the length of the SR node and the updates resume. Although the Rate-0 and Rate-1 nodes can also be represented as

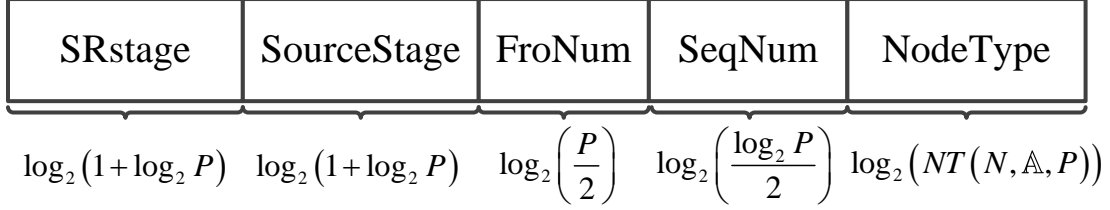


Figure 7.3 Instruction structure of the proposed SRFSC decoder.

special cases of SR nodes, the controller will bypass the SR module and signal the processing module to execute immediate decoding for these two nodes so that there is no additional latency.

The structure of the instructions used in the controller is shown in Fig. 7.3. The instructions contain all the required information to decode an SR node and they are stored in memory according to the visiting order in the decoding tree. The elements SRstage, SourceStage, FroNum, SeqNum and NodeType in the instruction represent the stage of SR node, the stage of source node, the number of frozen bits in source node, the base 2 logarithm of the number of repetition sequences and the node type, respectively. Note that an SR node is represented by three parameters as  $SR(v, SNT, r)$  in Section 6.3.1, here we use FroNum instead of SNT because no SR node with a Rate-C node as its source node is found for the code length ( $N = 1024$ ) and rates ( $R = 1/2, 1/4, 3/4$ ) that we consider in Section 7.3. When  $SNT \notin \text{Rate-C}$ , the source node is a special node whose bits are all non-frozen except the leftmost FroNum bits, where

$$\text{FroNum} = \begin{cases} 0, & \text{if } SNT = \text{Rate-1}, \\ 1, & \text{if } SNT = \text{Rate-0}, \\ 2^h \text{ or } 2^h - 1, & \text{if } SNT = \text{EG-PC}, \end{cases} \quad (7.1)$$

and where  $h$  is the level of the leftmost Rate-0/REP node of the EG-PC node. Moreover, the vector  $v$  is replaced with SRstage, SeqNum and NodeType since these three elements can be used directly in the decoder, so that additional calculations (e.g., (6.7)) can be avoided. NodeType is in fact a pointer to the memory of repetition sequences. As only nodes with SeqNum  $> 0$  have non-zero repetition sequences that need to be stored, NodeType refers to these node types and is used as pointer to find their corresponding repetition sequences in the memory.

The different repetition sequences in the SR node are processed in parallel. Since a maximum of  $2P$  LLRs are input to the SR module each time, we have the constraint

$$2^{\text{SRstage} + \text{SeqNum}} \leq 2P. \quad (7.2)$$

All SR nodes that meet this constraint can be handled, while others are divided into smaller nodes. Therefore, SRstage and SourceStage always have values be-

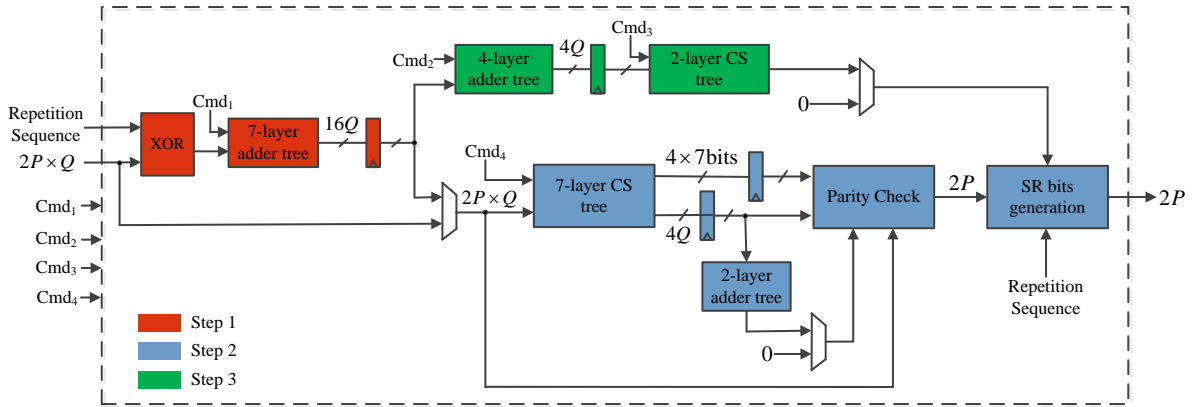


Figure 7.4 Example of the SR module architecture for  $N = 1024$ ,  $R \in \{1/2, 1/4, 3/4\}$ , and  $P = 64$ .

tween 0 and  $1 + \log_2 P$ . FroNum can be calculated according to (7.1) and thus have values between 0 and  $P/2$ . Consider source node with a minimum length of  $2^1$ . Then, the maximum value of SeqNum is constrained by  $2^{1+2\text{SeqNum}} \leq 2P$ . Thus, SeqNum has values between 0 and  $\frac{1}{2} \log_2 P$ . As for NodeType, it has values between 0 and  $NT(N, A, P)$ , where  $NT$  is a function of  $N$ ,  $A$  and  $P$ , which depends on the polar code being decoded.

As an example, we consider a set of 5G polar codes [137] of length  $N = 1024$  and rates  $R = 1/2$ ,  $R = 1/4$ , and  $R = 3/4$ . For a code length of  $N = 1024$ ,  $P = 64$  is shown to be a reasonable choice [92]. With these parameters, in Fig. 7.3, SRstage and SourceStage take values in  $\{0, 1, \dots, 7\}$ , FroNum takes values in  $\{0, 1, \dots, 3\}$ , and SeqNum takes values in  $\{0, 1, 2\}$ . The three considered codes contain a total of six SR nodes with  $\text{SeqNum} > 0$ . As such, NodeType takes values in  $\{0, 1, \dots, 6\}$ . Specifically, when  $\text{NodeType} = 0$ , the node only has an all-zero repetition sequence and the remaining values represent the six SR nodes with  $\text{SeqNum} > 0$ . From the above analysis, the size of each instruction for the considered example is 13 bits.

### 7.2.3 SR module

The ranges of some elements in the instructions are variable and depend on the set of supported polar codes. Thus, some of the data widths in the SR module are also variable and it is difficult to give a fully generic explanation of our proposed architecture. For this reason, we consider the previous example of  $N = 1024$ ,  $R \in \{1/2, 1/4, 3/4\}$ , and  $P = 64$ . The architecture of the SR module for this example is shown in Fig. 7.4. The submodules with red, blue, and green color correspond to the operations in Step 1, Step 2, and Step 3 in Algorithm 6, respectively, and are explained in more detail in the sequel.

<sup>1</sup>Note that the source node has a minimum length of 2 as all the possible frozen bit patterns with length 2 fall into the category in (7.1).

**Step 1:** This part of the SR decoder is used to calculate the input LLRs into the source node if  $\text{SRstage} \neq \text{SourceStage}$ . In the XOR submodule, the first  $2^{\text{SRstage}}$  LLRs in the  $2P$  inputs are repeated  $2^{\text{SeqNum}}$  times so that the decoding for different repetition sequences can be handled in parallel. The repetition sequences are obtained using  $\text{NodeType}$ . They will be XORed with the sign bit of the  $2^{\text{SeqNum}}$  input repetitions according to (6.8). The XOR result of different repetition sequences are concatenated and expanded into a vector of length  $2P$  by appending zeros if  $2^{\text{SRstage} + \text{SeqNum}} < 2P$ . Then, the LLR vector enters a  $(1 + \log_2 P)$ -layer adder tree that performs the addition of LLRs in (6.8). The command signal  $\text{Cmd}_1 = 7 - (\text{SRstage} - \text{SourceStage})$  is pre-calculated in the control module and it is used in the adder tree to decide the addition result of which layer will be output by a multiplexer. Those outputs from the adder tree are the input LLRs of the source node for different repetition sequences. In the considered example, there exist  $2^{\text{SourceStage} + \text{SeqNum}} \leq 16$  for SR node whose  $\text{SRstage} \neq \text{SourceStage}$ . Moreover, all LLRs are quantized using  $Q$  bits. Thus, the data width of the adder tree output is  $16Q$  bits.

**Step 2:** This part of the SR node is used to perform the parity-check and bit-flipping steps for the source node. The LLRs of the source node first enter a  $(1 + \log_2 P)$ -layer compare-select (CS) tree. Processing units in the CS tree execute the  $f$  function to decode SPC node. There are two cases where more than one SPC nodes will be decoded in parallel in our design: 1) when  $\text{FroNum} = 1$  and  $\text{SeqNum} > 0$ , there are  $2^{\text{SeqNum}}$  SPC nodes which correspond to different repetition sequences and are decoded simultaneously, and 2) when  $\text{FroNum} = 2$  and  $\text{FroNum} = 3$ , the decoding of source node can be viewed as a parallel decoding of 2 and 4 SPC nodes, respectively, as analyzed in Section 6.3.2. The length of the SPC node decides the layer from which the index of the least reliable input and the  $f$  function result are selected. As the length of the SPC node can be calculated as  $2^{\text{SourceStage} + 1 - \text{FroNum}}$ , the output layer selection signal  $\text{Cmd}_4$  has the following representation

$$\text{Cmd}_4 = \begin{cases} 7, & \text{FroNum} = 0, \\ 6 - \text{SourceStage} + \text{FroNum}, & \text{otherwise.} \end{cases} \quad (7.3)$$

Since the maximum number of parallel SPC nodes in our example is 4, the output indices and LLRs have a data width of  $4 \times 7$  and  $4Q$  bits, respectively. Note that the output LLRs goes both to the parity check module and a 2-layer adder tree. This is because all SPC nodes have an even parity constraint except when  $\text{FroNum} = 3$ , where SPC nodes can have an even or odd parity constraint which is calculated according to (6.5) and implemented by a 2-layer adder tree.

The parity constraint type, the output indices, and LLRs are then input into the parity-check submodule to do the parity check and bit flipping on these SPC nodes using (2.50). Then, the estimated bits of these SPC nodes are concatenated to form the estimated bits of source node and they are XORed with the repetition sequence to generate the estimated bits of SR node in the SR bits generation submodule according to (6.6). Finally, the SR bits corresponding to the repetition sequence with the index value from Step 3 are selected as the output.

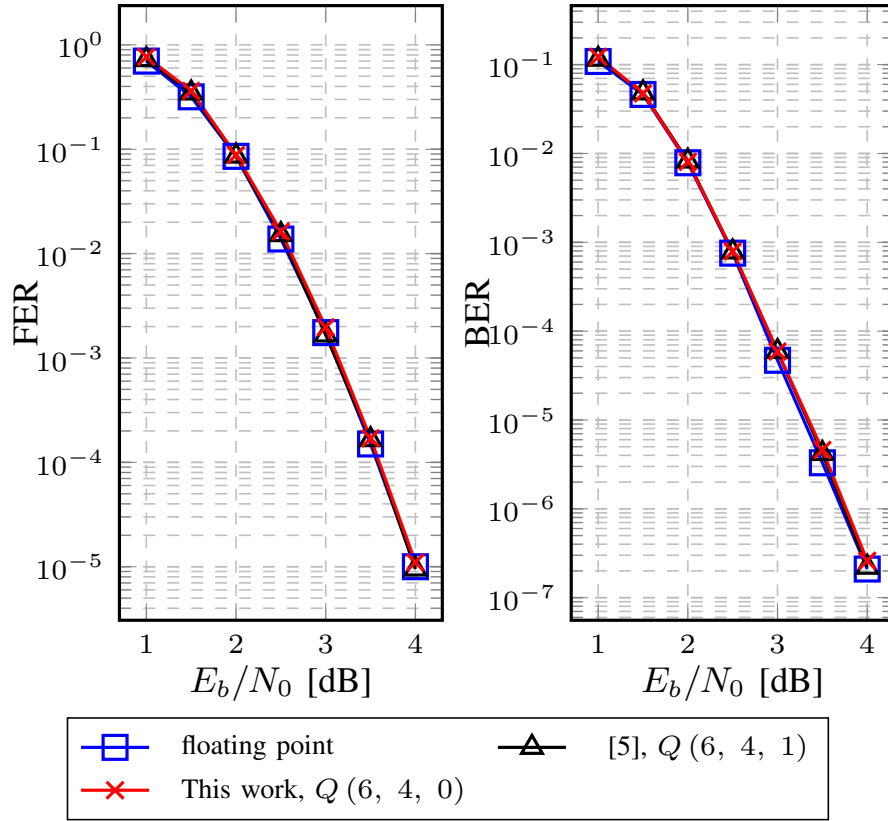


Figure 7.5 Floating-point and fixed-point FER and BER performance for SRFSC decoding of 5G polar codes  $\mathcal{P}(1024, 512)$  [137].

**Step 3:** This part of the SR decoder is executed in parallel with Step 2 to evaluate (6.10) using a  $(\text{SourceStage} + \text{SeqNum})_{\max}$ -layer adder tree and  $\text{SeqNum}_{\max}$ -layer CS tree, where  $(\text{SourceStage} + \text{SeqNum})_{\max}$  is the maximum value of  $(\text{SourceStage} + \text{SeqNum})$  for all SR nodes with  $\text{SeqNum} > 0$  and  $\text{SeqNum}_{\max}$  denotes the maximum value of  $\text{SeqNum}$ . As only magnitudes of LLRs are used for addition in (6.10), all inputs are positive. As a result, the processing unit in the 4-layer adder tree is simpler than that in the 7-layer adder tree in Step 1 because it does not need to compare magnitudes. The output of the adder tree is selected by the output layer selection signal  $\text{Cmd}_2 = 4 - \text{SourceStage}$  and has a bit-width of  $4Q$  as there are at most 4 repetition sequences in the considered example. The four sums are then input into the 2-layer CS tree to find the index of the maximum using selection signal  $\text{Cmd}_3 = 2 - \text{SeqNum}$ . Finally, the index is obtained from a multiplexer and the value is 0 if  $\text{SeqNum} = 0$  and the output from the CS tree otherwise.

### 7.3 Implementation results

The proposed decoder has been implemented using VHDL and targeting an Altera Stratix IV EP4SGX530KH40C2 FPGA device. Channel LLRs are generated by



Table 7.1 FPGA Implementation Results for  $\mathcal{P}$  (1024, 512).

	[88]	[92]	This Work
Quantization	$Q(6, 4, 1)$	$Q(6, 4, 1)$	$Q(6, 4, 0)$
$P$	64	64	64
LUTs	<b>6126</b>	14300	17615
Registers	1223	<b>1216</b>	10505
RAM (bits)	23592	18350	<b>16128</b>
Instruction size	<b>5 bits</b>	6 bits	13 bits
# of Instruction	209	157	<b>41</b>
# of CLKs	266	<b>214</b>	222
$f_{\max}$ (MHz)	99.8	89.6	<b>109.6</b>
$T/P$ (Mps)	384	428.6	<b>505.6</b>

transmitting random codewords through an additive white Gaussian noise (AWGN) channel after binary phase-shift keying (BPSK) modulation. A quantization scheme  $Q(6, 4, 0)$  has been used, where  $Q(Q_i, Q_c, Q_f)$  are the quantization bit size for internal LLRs, channel LLRs, and fraction bit size for both internal and channel LLRs, respectively. This scheme leads to an error-correcting performance that is very close to that of the floating-point implementation, as shown in Fig. 7.5.

Table 7.1 compares the proposed decoder with other state-of-the-art works. As can be seen, the proposed SRFSC decoder provides a 17.9% and 31.7% throughput improvement compared to the architectures presented in [92] and [88], respectively. This is mainly due to a 9.9% and 22.4% higher  $f_{\max}$  with respect to [88] and [92]. The number of CLKs in our work is slightly higher than that in [92]. This is because of the insertion of some registers to decrease certain critical paths and because we have not merged  $f$  and  $g$  operations as was done in [92]. In addition, a total of 186,200 CLKs are required at rates 1/4, 3/4, respectively. In terms of the used LUTs, this work requires an increase of 23.2% and 187.5% compared to [92] and [88], respectively. As far as the memory size is concerned, although our decoder uses fewer RAM bits, the required number of registers is about 8 times higher compared to [92, 88]. The big difference in registers can be mostly attributed to the separate storage of channel and internal LLRs in synthesis. Internal LLRs are stored in RAM and channel LLRs are arranged in registers, while in other works both are stored in RAM.

## 7.4 Conclusion

In this chapter, we presented the first FPGA implementation of the SRFSC decoder for polar codes. To this end, we designed a dedicated architecture for the SR node

processor. For a 5G polar code with length 1024, code rate 1/2 and  $P = 64$  processing units, we obtained a 17.9% improvement in throughput over the previous work. While there is a gain with respect to previous work, it is not huge, which means that it may not be worth to explore nodes at even higher levels than the SR node. So, the SR node seems to be the “ultimate node” from a practical perspective.



---

---

**OPTICAL WIRELESS COMMUNICATION SYSTEM  
VALIDATION OF POLAR CODES**

---

---

This chapter is adapted from: H. Zheng, K. Wu, B. Chen, J. Huang, Y. Lei, C. Li, A. Balatsoukas-Stimming, Z. Cao, A. M. J. Koonen, "Experimental demonstration of 9.6 Gbit/s polar coded infrared wireless communication system", IEEE Photonics Technology Letter, accepted.

In this chapter, a new inter-frame polar coded modulation scheme is experimentally demonstrated in an infrared light communication (ILC) system. The scheme utilizes the Monte Carlo (MC) method to jointly design an inter-frame polar code with 16-ary quadrature-amplitude modulation (16QAM) and orthogonal frequency-division multiplexing (OFDM). The indoor transmission of 9.6 Gbit/s 16QAM OFDM signal is experimentally achieved over a 3.2 km single-mode fiber and 0.8 m free space. The experiment results show that the proposed scheme employing a polar code of length 1024 and cyclic redundancy check aided successive cancellation list (CA-SCL) decoding with a list size of 2 resulted in no errors over  $10^7$  bits. Moreover, the proposed scheme requires negligible extra decoding complexity with respect to its classical counterpart, MC-constructed polar coded modulation. To the best of our knowledge, this is the first experimental demonstration of a polar coded modulation based infrared light communication system.

## **8.1 Introduction**

To increase the reliability and reach of BS-ILC links, forward error-correction (FEC) is one of the vital techniques, especially when the received optical power is low and high-order modulation formats are used. Some previous works have applied polar codes to VLC [72, 152] and proved its superiority over LDPC codes at short and moderate code lengths using CA-SCL decoding [65]. As discussed in Section 1.3.2, short code lengths are generally favored by latency-stringent short-reach OWC systems for their low decoding latency. Nevertheless, short code length cannot release the full potential of polar codes as the consequence of small number of polarizations. In order to break through this bottleneck, using CA-SCL as underlying decoder, a class of inter-frame polar coding with dynamic frozen bits was proposed in Chapter 5. Different from conventional polar coding where the encoding/decoding of each frame is independent of each other, the proposed scheme in Chapter 5 establishes a relationship between two adjacent frames. By exploiting this correlation in decoding,

a remarkable improvement can be achieved in the average probability of correct decoding with negligible complexity increment compared to conventional polar coding.

So far, the application of polar codes in ILC systems has not been reported. In this chapter, we experimentally investigate the application of the inter-frame polar coding in an ILC system. More specifically, the Monte Carlo (MC)-constructed inter-frame polar coded modulation scheme is proposed where MC method is utilized to jointly design the inter-frame polar code with 16-ary quadrature-amplitude modulation (16QAM) and orthogonal frequency-division multiplexing (OFDM). The proposed scheme is specified for the ILC system and the proper values of the number of shared bits  $m$  are found for polar codes with rate  $R = 0.8$  and two lengths  $N = 512$  and  $1024$ . Finally, the experimental setup and results are shown, verifying the effectiveness of the proposed scheme in ILC system successfully. The rest of this chapter is organized as follows. The principle of polar-coded modulation is explained in Section 8.2. Section 8.3 demonstrates the experimental setup and results. Conclusions are drawn in Section 8.4.

## 8.2 Principle

### 8.2.1 Bit-interleaved polar-coded modulation

Bit-interleaved polar-coded modulation (BIPCM) is a common way to realize joint design of polar code and high-level modulation [153]. Conventional BIPCM uses an interleaver to improve the decoding performance. As shown in [72], the interleaver is unnecessary if the code construction takes the mapped bit-level difference into account. In our design, the MC method is applied in the code construction, which not only considers the mapped bit-level difference but also frequency selective fading on OFDM subcarriers. Thus, the orthogonal circulant matrix transform (OCT) used to overcome frequency selective fading in [72] is also unnecessary. MC method generates random information bits to calculate coded symbols repeatedly and transmit them over the ILC system. The received signals are fed into the soft demodulator to generate LLR of each bit by calculating:

$$L(X_i) = \ln \left( \frac{\sum_{X \in \mathcal{C}^{[i,0]}} P(Y|X)}{\sum_{X \in \mathcal{C}^{[i,1]}} P(Y|X)} \right), \quad (8.1)$$

where  $X_i$  denotes the  $i$ -th bit of the transmitted symbol  $X$ ,  $\mathcal{C}^{[i,b]}$  is the set of symbols whose  $i$ -th bit is equal to  $b \in \{0,1\}$ .  $P(Y|X)$  is the conditional probability of receiving  $Y$  as symbol  $X$  is transmitted. For simplicity, we approximate the channel as an additive white Gaussian noise (AWGN) channel, whose signal-to-noise ratio (SNR) equals the average SNRs of all OFDM subcarriers. After obtaining the channel LLRs in (8.1), they are input into the SC decoder to make bit-by-bit estimation

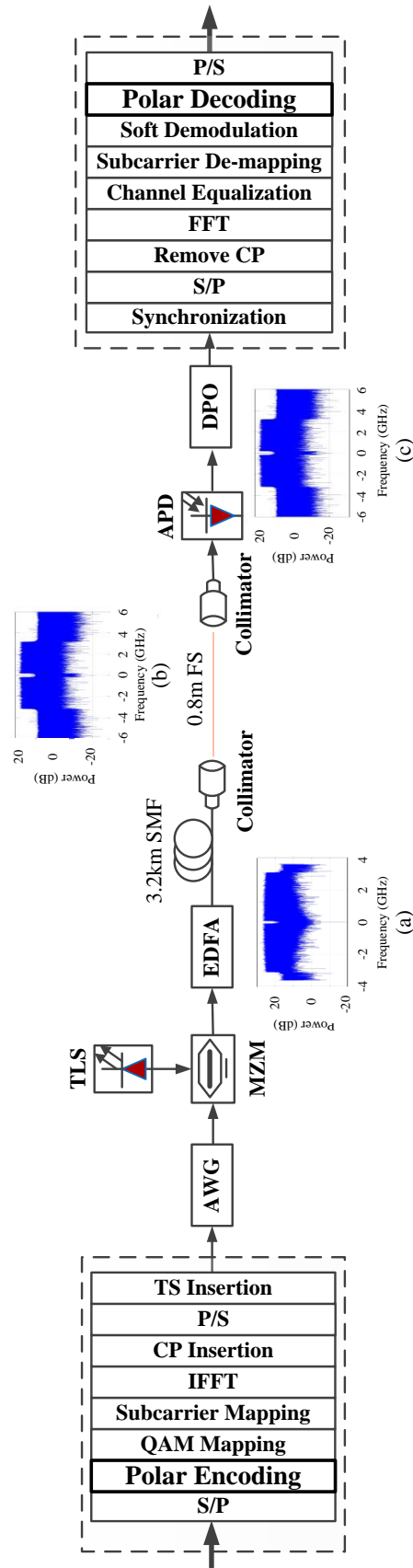


Figure 8.1 Block diagram of inter-frame polar coded modulation OFDM-ILC system. (S/P=serial to parallel conversion, IFFT=inverse fast Fourier transform, TS=training sequence, AWG=arbitrary waveform generator, TLS=tunable laser source, MZM=Mach-Zehnder modulator, EDFA=Erbium Doped Fiber Amplifier, SMF=single-mode fiber, FS=free-space, APD=avalanche photodiode, DPO=digital phosphor oscilloscope, P/S=parallel to serial conversion.) (a), Electrical spectrum after SMF. (b), Electrical spectrum after SMF+FS. (c), Electrical spectrum after SMF+FS.

assuming all previous bits are decoded correctly. The estimate is compared with the sent value for error detection of each bit. After transmitting enough coded symbols, the error probability of each polarized bit channel can be estimated with a ratio of the counted errors to the number of transmitted symbols. Consequently, the reliability of different bit indices can be sorted and the vector of relative reliabilities of bit indices  $\mathbf{v}$  can be generated. This joint design of polar code and modulation using MC method is called MC-constructed polar coded modulation, where classical polar coding is employed. If the classical polar coding is replaced by inter-frame polar coding, the MC-constructed inter-frame polar coded modulation scheme is proposed as a result.

### 8.3 Experimental setup and results

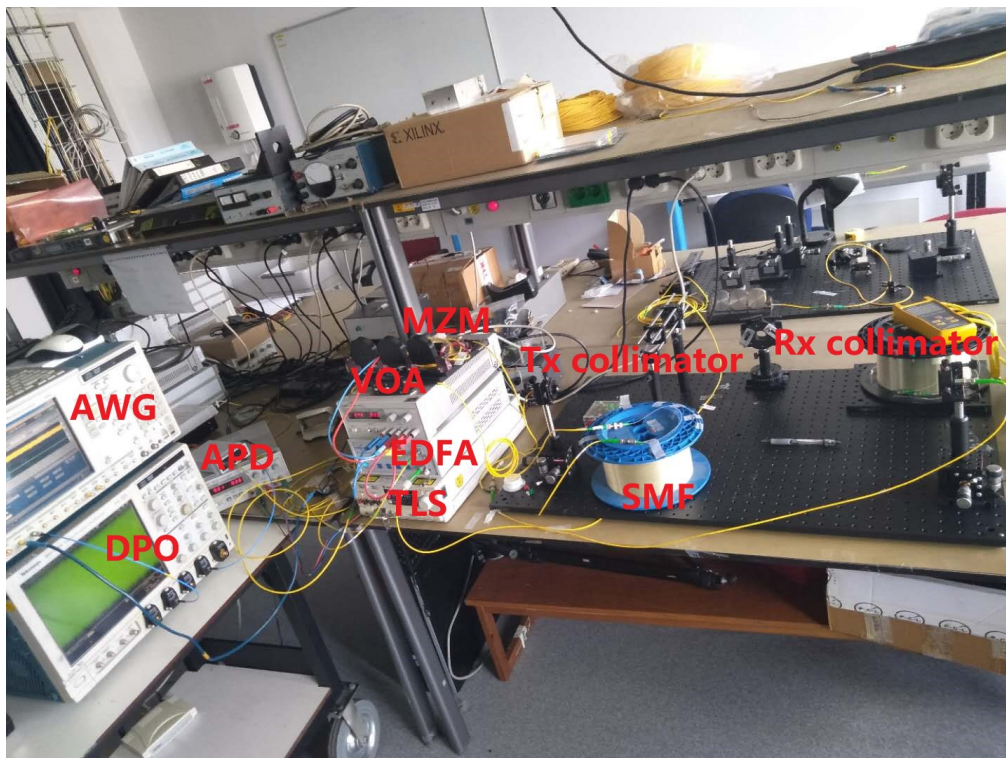


Figure 8.2 Photo of laboratory demonstrator setup. (VOA=variable optical attenuator.)

The block diagram of the experimental setup for the 9.6 Gbit/s polar coded 16QAM OFDM-ILC system is shown in Figure 8.1. The physical details of the setup are shown by the photo in Figure 8.2. Three kinds of coding schemes, namely, the classical MC-constructed polar coded modulation, the proposed MC-constructed inter-frame polar coded modulation and uncoded are investigated and compared in our experiments. Two code lengths of 512 and 1024 are considered in polar coding scheme. The used IFFT size is 512. Consider the Hermitian symmetry operation

for an intensity modulation direct detection (IM/DD) system, only half subcarriers can be used. The first 8 subcarriers are set to zero to avoid the direct current (DC) influence and a maximum of 248 subcarriers can be used. Due to the frequency selective fading from optical devices in ILC system, high frequency components in OFDM symbols are suffered from low SNR. To get a reasonable performance and also accommodate with the code length, we set the number of used subcarriers to the largest power of 2 less than 248, that is 128. In our system with 16QAM modulation and digital OFDM signal with 128 subcarriers loaded with data, one OFDM symbol carries 512 bits. Hence, the transmissions of 512 bits and 1024 bits codeword require 1 and 2 OFDM symbols, respectively.  $10^4$  codewords are generated offline for each code length. An arbitrary waveform generator (AWG) running at 12 GSa/s is used to produce the OFDM baseband signal, which then drives the Mach-Zehnder modulator (MZM). The effective bandwidth of the baseband OFDM signal is  $128/512 \times 12 = 3$  GHz. With 16QAM modulation, we achieve a data rate of 12 Gb/s. The optical carrier with 10 dBm optical power is generated from a tunable laser source (TLS) at 1550 nm wavelength. The modulated optical signal is firstly amplified by an Erbium Doped Fiber Amplifier (EDFA), transmitted over 3.2 km single mode fiber (SMF), and then launched into the free-space (FS) via a fiber collimator with 10 dBm optical power. After 0.8 m FS link, the optical signal is coupled into a short section of SMF with more than 0 dBm optical power via another fiber collimator. It should be noted here that the free-space length is limited by the size of our lab table, but longer lengths using collimated narrow beam within a typical indoor room size would not result in significant performance variation [154]. At the receiver, the 16QAM OFDM signal is converted into electrical domain by an avalanche photodiode (APD) module, which is then sampled by a digital phosphor oscilloscope (DPO) at 25 GSa/s for further offline signal processing. The electrical spectrum of the OFDM signals at back-to-back (BTB), after transmission through the SMF, after transmission through the SMF and FS are demonstrated in Fig. 8.1 (a), (b), (c), respectively.

The construction of the polar codes depends highly on the channel. In this work, an indoor optical wireless channel combined with a section of SMF is adopted in the experiment. The target SNR is designed to be 14 dB, which corresponds to the received optical power (ROP) of  $-19$  dBm in our experimental system. This SNR was targeted because the polar decoder requires a larger list size to achieve a reasonable performance for lower SNRs, while for the higher SNRs, more OFDM symbols are required to transmit to get an accurate bit error rate (BER) value. The code rates for the two code lengths are  $R = 0.8$ . Hence, the net data rate is 9.6 Gbit/s. It is shown in [10] that there exists a specific value of the parameter  $m$  with which the best error-correction performance can be achieved. The experiment sets  $m = 15$  and 30 for  $N = 512$  and 1024, respectively, which are firstly chosen through simulations and then by the experiments. For both the classical and proposed schemes,  $L = 1$  and 2 are considered in CA-SCL decoding and CRC16-CCITT is adopted.



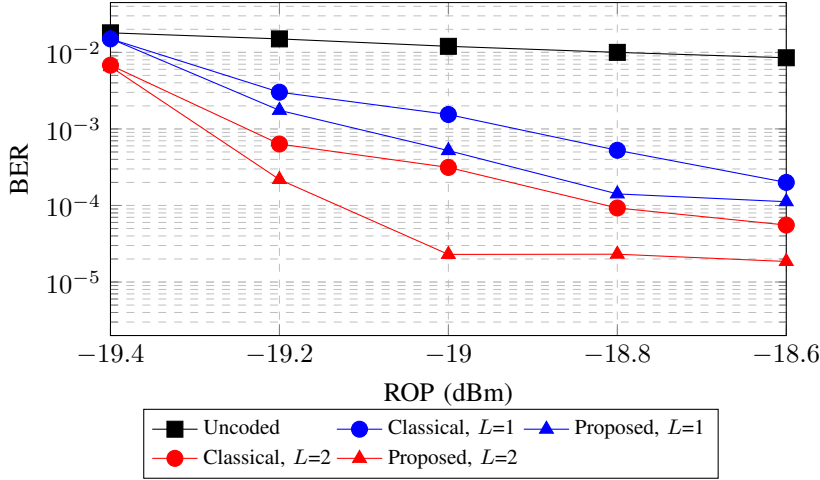


Figure 8.3 Error-correction performance of the proposed scheme with  $N = 512$ ,  $R = 0.8$ ,  $m = 15$  and designed ROP =  $-19$  dBm.

Fig. 8.3 shows the measured BER performance for different coding schemes as a function of the received optical power (ROP) at the code length of  $N = 512$ . The uncoded scheme retains a BER of about  $1.2 \times 10^{-2}$  around the designed operating point. As a comparison, the two schemes with polar coding show a significant performance improvement. It can be seen that the proposed scheme brings substantial performance gain in comparison with the classical scheme. At a ROP of  $-19$  dBm, the proposed scheme with CA-SCL ( $L = 2$ ) decoding achieves a BER of  $2.3 \times 10^{-5}$ , while that of its counterpart is  $3.1 \times 10^{-4}$ . As can be seen in the figure, this improvement decreases when the value of ROP deviates from  $-19$  dBm. As the vector  $\mathbf{v}$  changes with the deviation of ROP, the indices of bits in  $\mathcal{A}$ ,  $\mathcal{A}^V$ ,  $\mathcal{A}_c$ ,  $\mathcal{A}_c^\wedge$  selected for the designed operating point  $-19$  dBm are no longer optimal at a different ROP. This also explains why some curves do not keep dropping with the increase of ROP. In addition, it can be observed that a larger list size leads to a larger performance gap between the classical and proposed schemes. As illustrated in Section 5.2, the probability of successful decoding of **M2** will increase when using large list size, which leads to a higher probability of re-decoding for decoding failed frame. As a consequence, the overall average probability of decoding success will have a more pronounced improvement.

Fig. 8.4 displays the measured BER results at a code length of 1024 bits. Note that when measurements with no errors are observed, those are not included in the plot. Compared to Fig. 8.3, the performance gain brought by polar codes is higher at  $N = 1024$  for both the classical and proposed schemes. Specifically, for the proposed scheme using CA-SCL ( $L = 2$ ) decoding, measurements at a ROP of  $-19$  dBm resulted in no errors over the approximately  $10^7$  bits, whereas errors were observed for the classical scheme. Note that the performance can be improved by increasing the value of  $L$ , while at the cost of higher complexity. Fig. 8.4 also shows the BER results at  $N = 1024$  when codes optimized for each ROP are used. Significant improvement can be observed when comparing to those obtained using

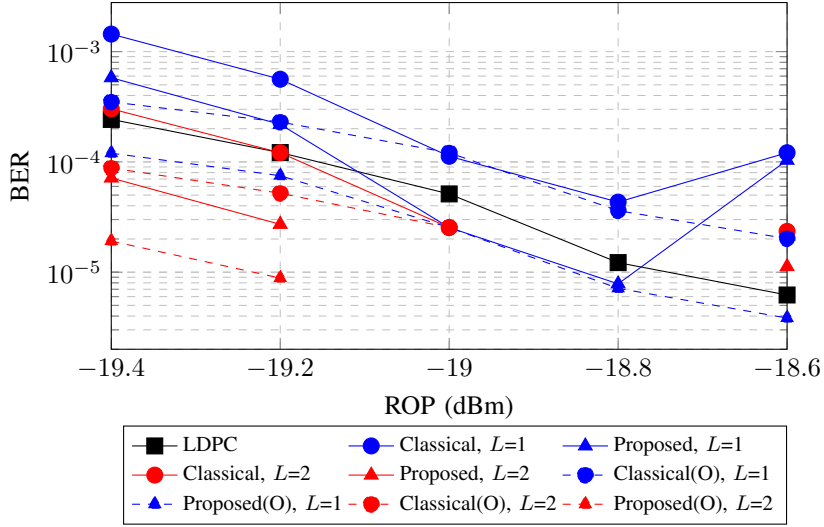


Figure 8.4 Error-correction performance of the proposed scheme with designed ROP = -19 dBm, with codes optimized for each ROP (O), and LDPC code with  $I = 2$ , when  $N = 1024$ ,  $R = 0.8$ ,  $m = 30$ .

fixed code designed for ROP= -19 dBm. The performance of LDPC code with layered decoding algorithm (LDA) [65] and number of iterations  $I = 2$  has been provided for comparison. We can find the performance of the proposed scheme outperforms that of LDPC when  $I = L = 2$ . Moreover, the performance advantage of the proposed scheme over the classical one can also be observed when 256QAM is adopted as shown in Fig. 8.5.

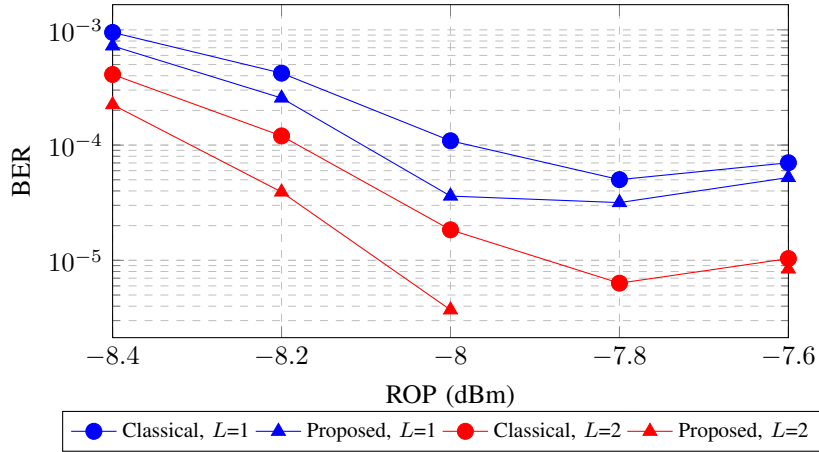


Figure 8.5 Error-correction performance of the proposed scheme using 256QAM with  $m = 30$ ,  $N = 1024$ ,  $R = 0.8$  and designed ROP = -8 dBm.

Table 8.1 reports the probability of re-decoding  $P_{re}$  and the normalized average and maximum computational complexity increment  $\eta_{aver}$  and  $\eta_{max}$  using CA-SCL ( $L = 2$ ).  $\eta_* = (C_{prop}^* - C_{clas}^{aver}) / C_{clas}^{aver}$ , where  $C_{prop}^{aver}$  and  $C_{prop}^{max}$  represents the average and maximum complexity of the proposed scheme, respectively,  $C_{clas}^{aver}$  denotes the average complexity of the classical scheme. The complexity is calculated in terms of

the number of arithmetic operations required for the decoding of one frame by each scheme with  $N = 512, 1024$  and CA-SCL ( $L = 2$ ) decoding. As shown in Table 8.1, while  $\eta_{\max}$  is about 100%,  $\eta_{\text{aver}}$  is only up to 2.4% and 0.1% when  $N = 512$  and 1024, respectively. It is worth mentioning that  $\eta_{\text{aver}}$  reduces as the code length or ROP increases. In terms of decoding latency, the maximum decoding latency of a frame is about twice as that of the classical scheme when re-decoding is executed. However, the increase in average latency is only up to 2.4% and 0.1% with respect to the classical counterpart when  $N = 512$  and 1024, respectively.

Table 8.1 Probability of re-decoding and normalized computational complexity increment comparison using CA-SCL ( $L = 2$ ).

		ROP (dBm)	-19.4	-19.2	-19	-18.8	-18.6
$N=512$	$P_{\text{re}}$		2.4%	0.3%	0.1%	0.06%	0.03%
	$\eta_{\text{aver}}$		2.4%	0.3%	0.1%	0.06%	0.03%
	$\eta_{\max}$		99.45%	99.45%	99.45%	99.45%	99.45%
$N=1024$	$P_{\text{re}}$		0.12%	0.06%	0.01%	0.005%	0.005%
	$\eta_{\text{aver}}$		0.12%	0.06%	0.01%	0.005%	0.005%
	$\eta_{\max}$		100%	100%	100%	100%	100%

## 8.4 Conclusion

In this Chapter, an inter-frame polar coded modulation scheme is proposed and experimentally demonstrated in a 16QAM OFDM-ILC system with a net data rate of 9.6 Gbit/s. The experimental measurements show that the error-correction performance of the ILC system with the proposed inter-frame polar coded modulation scheme outperforms that of the classical counterpart significantly. Specifically, the proposed scheme employing a polar code of length 1024 and CA-SCL ( $L = 2$ ) decoding resulted in no errors over  $10^7$  bits. In addition, it has an average decoding complexity which is only up to 2.4% higher than that of the classical scheme. Therefore, the proposed inter-frame polar coded modulation scheme is a promising candidate technology for indoor ILC systems.

SUMMARY AND FUTURE WORK

9.1 Summary of contributions

In this thesis, efforts have been made towards realizing high-reliability, high-speed and low-power indoor optical wireless communications from the perspective of channel codes, in particular polar codes, mainly by means of developing advanced decoders with low complexity, high error-correction performance and low latency as shown in figure below.

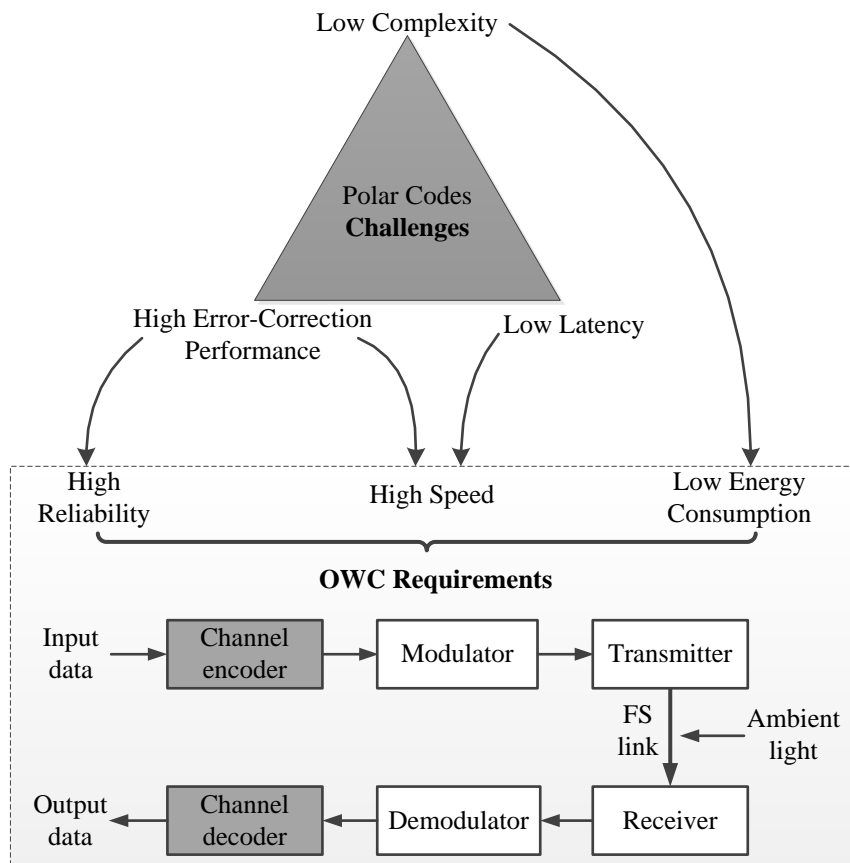


Figure 9.1 Challenges of Polar Codes to meet the requirements of OWC.

In **Chapter 1**, first, the optical wireless communication (OWC) and the role of channel codes in OWC were briefly introduced. Literature reviews on the applications of various channel codes including the latest polar codes in OWC were then elaborated. Finally, the three main challenges in using polar codes to attain high-reliability, high-speed and low-power indoor optical wireless communication were investigated as motivations of our research. **Chapter 2 and 3** further gave in-depth introductions of polar codes and OWC to familiarize the reader with their main principles, construction and way of working.

Chapter 4, 5, 6 gave solutions to the three challenges: low complexity, high error-correction performance and low latency, respectively, from algorithm level and theoretical perspective. More specifically, **Chapter 4** proposed the enhanced list-aided successive cancellation stack (ELSCS) decoding algorithm with a new decoding path searching principle which combines the advantages of both successive cancellation list (SCL) and successive cancellation stack (SCS) decoding. This multimode decoder offers a flexible tradeoff between time complexity and computational complexity by adjusting the working mode. The adjustable complexity performance varies among maximum time/minimum computational complexity (similar to SCS) and minimum time/maximum computational complexity (similar to SCL) and some intermediate complexity states freely, while ensuring a stable error-correction performance. In addition, the introduction of a log-likelihood ratio (LLR)-threshold based path extension scheme reduces the required storage space of ELSCS by 70%. Making use of its property, we can choose different modes of ELSCS algorithm to meet different application requirements at a low computational complexity.

In **Chapter 5**, an inter-frame related polar coding scheme with carefully designed encoding and decoding algorithms was presented. The proposed scheme achieves significant improvement in error-correction performance with negligible increase in memory and average computational complexity compared to conventional polar coding scheme. Moreover, it does not need a long code length or incur rate loss like the concatenation scheme and spatially coupled codes analyzed in **Chapter 1**. Simulation results show that for a polar code of length 1024, rate 1/2 and list size 16, the proposed inter-frame polar coding scheme can provide a performance gain of 0.28 dB over the classical scheme at the block error rate of  $10^{-4}$  with 5% and 2% increase in memory and computational complexity, respectively.

This thesis further proposed two solutions for latency reduction in decoding in **Chapter 6**. The first was a fast parallel decoder based on sequence repetition (SR) node. SR node gives a unified description of most existing fast decoding node types. We proved an important property of the SR node which enables the design of an efficient fast decoder with lower latency than the state-of-the-art. The second was a new threshold-based hard-decision-aided (TA) scheme, which speeds up the decoding of general nodes outside the class of SR nodes, especially when the noise is low. Compared to existing schemes, the proposed one has more latency reduction and its threshold value can be pre-calculated for a desired error-correction performance.

For a polar code of length 1024 and rate 1/2, 57% reduction in the average decoding latency can be achieved with the help of the TA scheme at  $E_b/N_0 = 5$  dB .

**Chapter 7** reported the first hardware implementation of the SR-node-based fast SC decoder. A dedicated architecture was designed for the SR node processor and specified for a 5G polar code with code length 1024, three code rates 1/4, 1/2, 3/4, and 64 processing elements. A 17.9% improvement in throughput can be observed over the previous work at rate 1/2 in the FPGA implementation result.

In **Chapter 8**, an inter-frame polar coded modulation scheme was proposed, and experimentally demonstrated in a 16-QAM OFDM-ILC system with a net data rate of 9.6 Gbit/s. This is the first demonstration of infrared light communication system employing polar codes. The proposed scheme with a polar code of length 1024 and CA-SCL ( $L = 2$ ) decoding resulted in no errors over  $10^7$  bits.

## 9.2 Future work

This thesis focused on three major challenges in terms of complexity, error-correction performance and latency that hinder the widespread use of polar codes. Based on the insights we have gained in the thesis, here we sketch several potential research directions which may lead to further interesting results in these three fields. We also raise some interesting open questions for other challenges that were not covered in this dissertation.

### 9.2.1 Complexity

The proposed ELSCS decoding algorithm uses a new decoding path searching principle, however, at each path extension stage, the length of decoding path will only increase by 1. An interesting research direction is to introduce the idea of fast SC decoding, that is, employing the special fast decoding nodes. In this way, decoding path will increase by the length of corresponding special node each time, thus the minimal time complexity can be further reduced and we will get a larger adjustable complexity range. Moreover, since ELSCS decoding has variable runtime similar to SCS decoding, its hardware implementation is a challenging problem.

### 9.2.2 Error-correction performance

In the proposed inter-frame related polar coding, consecutive frames share some mutual information. If the decoding of a frame failed, there is still chance to get the hard-valued estimates of these mutual information from adjacent frame to perform a re-decoding. A possible improvement is that the failed decoding frame uses the LLRs instead of hard-valued estimates of these shared bits to perform re-decoding. Soft messages carry more information than hard values and thereby improve the successful probability of re-decoding. The original inter-frame related polar coding establishes connection between two consecutive frames. Another interesting research

direction is to expand the connection to more than two frames, which means one frame will not only share information with its adjacent frames, but also non-adjacent frames. This expansion may give rise to more powerful and complex decoding algorithms providing greater performance gain.

### 9.2.3 Latency

Our next step in this research area is the design of a fast SCL decoder using SR nodes which can be of great interest since SCL decoder is more attractive in practical applications. Another improvement is to optimize the design of the hardware architecture. As shown in Chapter 7, the proposed decoder requires an increase in terms of the used look up tables (LUTs) and the memory size compared to the previous work. This is mainly due to a non-optimal design of memory module and partial sum network module. Moreover, the 505 Mbps throughput of the proposed decoder can not satisfy the demand of high-speed OWC system. A fully unrolled, deeply pipelined decoder architecture like what has been demonstrated in [155], but employing the proposed SR node, is an interesting open direction.

### 9.2.4 Experiment

We have achieved 9.6 Gbit/s with 16-QAM OFDM ILC system. The achievable data rate for other modulation formats such as OOK, PAM-4 should be further investigated. Moreover, the link reach was only 0.8 m in the experiment. How to increase the reach, to say  $> 3$  m, is a potential research direction.

### 9.2.5 Polar codes using MIMO and NOMA techniques

Multiple Input Multiple Output (MIMO)-OFDM and non-orthogonal multiple access (NOMA) are two key techniques to address the high throughput requirements for the fifth generation (5G) wireless networks. In order to further improve the system performance, the channel coding techniques have been introduced into the MIMO [156, 157] and NOMA [158–160] systems. Polar codes using MIMO and NOMA have been systematically studied in [161] and [162], respectively. They all follow the same idea that MIMO and NOMA transmissions are recognized from the novel perspective of channel polarization, thus enabling the joint design between the polar coding and the MIMO or NOMA transmission. Polar code (PC)-MIMO system has been proved to be capacity achieving for the MIMO channel. Compare to the counterparts using Turbo and LDPC codes, PC-MIMO and PC-NOMA provide better system performance with comparable complexity. In addition, parallel processing schemes were proposed to effectively reduce the processing latency with some loss in performance for both PC-MIMO and PC-NOMA. However, the proposed PC-MIMO and PC-NOMA schemes are only appropriate for the additive white Gaussian noise (AWGN) channel and the Rayleigh fading channel. For other channel models and changing channel condition, the construction of polar codes is still an open problem.

### 9.2.6 Other challenges

Consider the blockage and shadowing problems in an OWC system. If the impact on link appears instantly, such as the interruptions induced by human movement, a handover of the user's terminal to other transmitters is unwise in this case since it will switch back soon. An adaptive and robust transmission techniques is thereby required. One efficient solution is a channel coding scheme that enables incremental retransmissions at different rates in order to adapt to channel conditions. To this end, hybrid automatic repeat request (HARQ) schemes based on rate-compatible polar (RCP) codes were proposed in [163–165]. It is an interesting topic to explore their effectiveness in OWC systems.

This thesis is about the application of novel polar coding methods on OWC, in particular beam-steered OWC in case of only LoS. The multipath effects are neglected so that we assume memoryless channel throughout the thesis. If wide-beam OWC (like VLC) or Non-LoS links are considered, multipath effects are inevitable, leading to channel with memory. Although our proposed polar coding methods are still workable in this case, they are not optimal. New solutions have to be investigated to adapt to channels with memory. Some pioneering works have been done in [166–168].

From the experimental results in Chapter 8, we can find polar codes are highly channel dependent. A laboriously constructed polar code only works well within a small SNR range. Thus, how to retain the good performance of polar codes while reducing its dependence on channel condition is an open problem worth pursuing. Works [169, 170] have shed some light on this problem.

The application of polar codes in some special communication scenarios is also worthy of further study. For example, the bidirectional OWC link in machine-to-machine communications, where low latency and high reliability are very important so polar codes can play a key role. The construction and encoding/decoding of polar codes will be more flexible and intelligent in bidirectional link as in one direction you may learn from the other direction.





---

## REFERENCES

---

- [1] A. G. Bell, "Upon the production and reproduction of sound by light," *Journal of the Society of Telegraph Engineers*, vol. 9, no. 34, pp. 404–426, 1880.
- [2] A. Al-Kinani, C. Wang, L. Zhou, and W. Zhang, "Optical wireless communication channel measurements and models," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 3, pp. 1939–1962, 2018.
- [3] "Electromagnetic spectrum." [Online]. Available: <https://www.mpoweruk.com/radio.htm>.
- [4] T. H. Maiman, "Stimulated optical radiation in ruby," *Nature*, vol. 187, no. 4736, pp. 493–494, 1960.
- [5] N. Zheludev, "The life and times of the LED—a 100-year history," *Nature Photonics*, vol. 1, no. 4, pp. 189–192, 2007.
- [6] N. Holonyak Jr and S. Bevacqua, "Coherent (visible) light emission from Ga(As<sub>1-x</sub>P<sub>x</sub>) junctions," *Applied Physics Letters*, vol. 1, no. 4, pp. 82–83, 1962.
- [7] J. M. Kahn and J. R. Barry, "Wireless infrared communications," *Proceedings of the IEEE*, vol. 85, no. 2, pp. 265–298, 1997.
- [8] F. E. Goodwin, "A review of operational laser communication systems," *Proceedings of the IEEE*, vol. 58, no. 10, pp. 1746–1752, 1970.
- [9] T. Koonen, F. Gomez-Agis, F. Huijskens, K. A. Mekonnen, Z. Cao, and E. Tangdiongga, "High-capacity optical wireless communication using two-dimensional IR beam steering," *Journal of Lightwave Technology*, vol. 36, no. 19, pp. 4486–4493, Oct. 2018.
- [10] G. Pang, T. Kwan, H. Liu, and C.-H. Chan, "Optical wireless based on high brightness visible LEDs," in *IEEE Industry Applications Conference*, vol. 3, 1999, pp. 1693–1699.
- [11] Y. Tanaka, T. Komine, S. Haruyama, and M. Nakagawa, "Indoor visible communication utilizing plural white leds as lighting," in *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications. PIMRC 2001. Proceedings*, vol. 2, 2001, pp. F–F.
- [12] T. Komine and M. Nakagawa, "Fundamental analysis for visible-light communication system using LED lights," *IEEE Transactions on Consumer Electronics*, vol. 50, no. 1, pp. 100–107, 2004.
- [13] F.-M. Wu, C.-T. Lin, C.-C. Wei, C.-W. Chen, Z.-Y. Chen, and H.-T. Huang, "3.22-Gb/s WDM visible light communication of a single RGB LED employing carrier-less amplitude and phase modulation," in *IEEE Optical Fiber Communication Conference and Exposition and the National Fiber Optic Engineers Conference (OFC/NFOEC)*, 2013, pp. 1–3.
- [14] H. Chun, S. Rajbhandari, G. Faulkner, D. Tsonev, E. Xie, J. J. D. McKendry, E. Gu, M. D. Dawson, D. C. O'Brien, and H. Haas, "LED based wavelength division multiplexed 10 Gb/s visible light communications," *Journal of Lightwave Technology*, vol. 34, no. 13, pp. 3047–3052, 2016.
- [15] D. Tsonev, H. Chun, S. Rajbhandari, J. J. McKendry, S. Videv, E. Gu, M. Haji, S. Watson, A. E. Kelly, G. Faulkner *et al.*, "A 3-Gb/s single-LED OFDM-based wireless VLC link using a gallium nitride  $\mu$ LED," *IEEE Photonics Technology Letters*, vol. 26, no. 7, pp. 637–640, 2014.

- [16] G. Cossu, W. Ali, R. Corsini, and E. Ciaramella, "Gigabit-class optical wireless communication system at indoor distances (1.5–4 m)," *Optics Express*, vol. 23, no. 12, pp. 15 700–15 705, 2015.
- [17] F. Gfeller, H. Muller, and P. Vettiger, "Infrared communication for in-house applications," in *IEEE COMPCON*, vol. 78, 1978, pp. 132–138.
- [18] H. Le Minh, D. O'Brien, G. Faulkner, O. Bouchet, M. Wolf, L. Grobe, and J. Li, "A 1.25-Gb/s indoor cellular optical wireless communications demonstrator," *IEEE Photonics Technology Letters*, vol. 22, no. 21, pp. 1598–1600, 2010.
- [19] A. Koonen, C. Oh, and E. Tangdiongga, "Reconfigurable free-space optical indoor network using multiple pencil beam steering," in *IEEE OptoElectronics and Communication Conference and Australian Conference on Optical Fibre Technology*, 2014, pp. 204–206.
- [20] C. Oh, F. Huijskens, Z. Cao, E. Tangdiongga, and A. Koonen, "Toward multi-Gbps indoor optical wireless multicasting system employing passive diffractive optics," *Optics Letters*, vol. 39, no. 9, pp. 2622–2625, 2014.
- [21] C. Oh, E. Tangdiongga, and A. Koonen, "Steerable pencil beams for multi-Gbps indoor optical wireless communication," *Optics Letters*, vol. 39, no. 18, pp. 5427–5430, Sep. 2014.
- [22] —, "42.8 Gbit/s indoor optical wireless communication with 2-dimensional optical beam-steering," in *IEEE Optical Fiber Communications Conference and Exhibition (OFC)*, 2015, pp. 1–3.
- [23] T. Koonen, J. Oh, K. Mekonnen, Z. Cao, and E. Tangdiongga, "Ultra-high capacity indoor optical wireless communication using 2D-steered pencil beams," *Journal of Lightwave Technology*, vol. 34, no. 20, pp. 4802–4809, 2016.
- [24] T. Koonen, K. A. Mekonnen, F. Huijskens, N.-Q. Pham, Z. Cao, and E. Tangdiongga, "Fully passive user localization for beam-steered high-capacity optical wireless communication system," *Journal of Lightwave Technology*, vol. 38, no. 10, pp. 2842–2848, 2020.
- [25] T. Koonen, "Indoor optical wireless systems: technology, trends, and applications," *Journal of Lightwave Technology*, vol. 36, no. 8, pp. 1459–1467, 2017.
- [26] C. E. Shannon, "A mathematical theory of communication," *The Bell System Technical Journal*, vol. 27, no. 4, pp. 623–656, 1948.
- [27] D. J. Costello and G. D. Forney, "Channel coding: The road to channel capacity," *Proceedings of the IEEE*, vol. 95, no. 6, pp. 1150–1177, 2007.
- [28] E. Berlekamp, *Algebraic coding theory*. World Scientific, 1968.
- [29] A. Viterbi, "Convolutional codes and their performance in communication systems," *IEEE Transactions on Communication Technology*, vol. 19, no. 5, pp. 751–772, 1971.
- [30] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near shannon limit error-correcting coding and decoding: Turbo-codes. 1," in *IEEE International Conference on Communications*, vol. 2, 1993, pp. 1064–1070.
- [31] C. Berrou and A. Glavieux, "Near optimum error correcting coding and decoding: Turbo-codes," *IEEE Transactions on Communications*, vol. 44, no. 10, pp. 1261–1271, 1996.
- [32] R. Gallager, "Low-density parity-check codes," *IRE Transactions on Information Theory*, vol. 8, no. 1, pp. 21–28, 1962.

- [33] D. J. MacKay and R. M. Neal, "Near Shannon limit performance of low density parity check codes," *Electronics Letters*, vol. 33, no. 6, pp. 457–458, 1997.
- [34] Z. Si, R. Thobaben, and M. Skoglund, "Rate-compatible LDPC convolutional codes achieving the capacity of the BEC," *IEEE Transactions on Information Theory*, vol. 58, no. 6, pp. 4021–4029, 2012.
- [35] 3GPP, "Final report of 3GPP TSG RAN WG1 87 v1.0.0," Reno, USA, 2016. [Online]. Available: [http://www.3gpp.org/ftp/tsg\\_ran/WG1\\_RL1/TSGR1\\_87/Report/Final\\_Minutes\\_report\\_RAN1%2387\\_v100.zip](http://www.3gpp.org/ftp/tsg_ran/WG1_RL1/TSGR1_87/Report/Final_Minutes_report_RAN1%2387_v100.zip).
- [36] E. Arikan, "Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels," *IEEE Transactions on Information Theory*, vol. 55, no. 7, pp. 3051–3073, 2009.
- [37] I. S. Association *et al.*, "IEEE standard for local and metropolitan area networks-part 15.7: short-range wireless optical communication using visible light," *IEEE: Piscataway, NJ, USA*, pp. 1–309, 2011.
- [38] S. Rajagopal, R. D. Roberts, and S.-K. Lim, "IEEE 802.15. 7 visible light communication: modulation schemes and dimming support," *IEEE Communications Magazine*, vol. 50, no. 3, pp. 72–82, 2012.
- [39] T. A. Tran, N. Chehrazi, and T. Le-Ngoc, "Adaptive Reed-Solomon coding scheme for OFDM systems over frequency-selective fading channels," in *IEEE Conference on Vehicular Technology*, 2006, pp. 1–5.
- [40] V. J. Stulpman, J. Terry, and G. C. Orsak, "Approaches to adaptive Reed-Solomon coding for OFDM systems," in *IEEE Conference on Vehicular Technology*, vol. 62, no. 1, 2005, p. 453.
- [41] T. Ozeki and M. Nakagawa, "Adaptive Reed Solomon code allocation for wireless LAN," in *IEEE International Conference on Communication Systems*, vol. 1, 2002, pp. 309–312.
- [42] Y. Wei, J. He, R. Deng, J. Shi, S. Chen, and L. Chen, "An approach enabling adaptive FEC for OFDM in fiber-VLLC system," *Optics Communications*, vol. 405, pp. 329–333, 2017.
- [43] E. Pisek, S. Rajagopal, and S. Abu-Surra, "Gigabit rate mobile connectivity through visible light communication," in *IEEE International Conference on Communications (ICC)*, 2012, pp. 3122–3127.
- [44] J. Amano, T. Wada, and K. Mukumoto, "An evaluation of a parallel transmission visible light communication system employing an LDPC code," in *IEEE International Symposium on Information Theory and Its Applications*, 2012, pp. 566–570.
- [45] C. Tang, M. Jiang, H. Shen, and C. Zhao, "Analysis and optimization of P-LDPC coded RGB-LED-based VLC systems," *IEEE Photonics Journal*, vol. 7, no. 6, pp. 1–13, 2015.
- [46] K. S. Immink, "Runlength-limited sequences," *Proceedings of the IEEE*, vol. 78, no. 11, pp. 1745–1759, 1990.
- [47] X. Lu and J. Li, "New Miller codes for run-length control in visible light communications," *IEEE Transactions on Wireless Communications*, vol. 17, no. 3, pp. 1798–1810, 2017.
- [48] Z. Li, H. Yu, B. Shan, D. Zou, and S. Li, "New run-length limited codes in On–Off keying visible light communication systems," *IEEE Wireless Communications Letters*, vol. 9, no. 2, pp. 148–151, 2019.

- [49] S. Kim and S.-Y. Jung, "Novel FEC coding scheme for dimmable visible light communication based on the modified Reed–Muller codes," *IEEE Photonics Technology Letters*, vol. 23, no. 20, pp. 1514–1516, 2011.
- [50] —, "Modified Reed–Muller coding scheme made from the bent function for dimmable visible light communications," *IEEE Photonics Technology Letters*, vol. 25, no. 1, pp. 11–13, 2012.
- [51] J. Kim and H. Park, "A coding scheme for visible light communication with wide dimming range," *IEEE Photonics Technology Letters*, vol. 26, no. 5, pp. 465–468, 2014.
- [52] S. H. Lee and J. K. Kwon, "Turbo code-based error correction scheme for dimmable visible light communication systems," *IEEE Photonics Technology Letters*, vol. 24, no. 17, pp. 1463–1465, 2012.
- [53] S. Kim, "Adaptive FEC codes suitable for variable dimming values in visible light communication," *IEEE Photonics Technology Letters*, vol. 27, no. 9, pp. 967–969, 2014.
- [54] H. Wang and S. Kim, "New RLL decoding algorithm for multiple candidates in visible light communication," *IEEE Photonics Technology Letters*, vol. 27, no. 1, pp. 15–17, 2014.
- [55] X. Lu and J. L. Tiffany, "Achieving FEC and RLL for VLC: A concatenated convolutional–Miller coding mechanism," *IEEE Photonics Technology Letters*, vol. 28, no. 9, pp. 1030–1033, 2016.
- [56] S. Zhao, "A serial concatenation-based coding scheme for dimmable visible light communication systems," *IEEE Communications Letters*, vol. 20, no. 10, pp. 1951–1954, 2016.
- [57] J. Hagenauer, "Rate-compatible punctured convolutional codes (RCPC codes) and their applications," *IEEE Transactions on Communications*, vol. 36, no. 4, pp. 389–400, 1988.
- [58] T. Ohtsuki, "Rate adaptive indoor infrared wireless communication systems using repeated and punctured convolutional codes," in *IEEE International Conference on Communications*, vol. 1, 1999, pp. 609–613.
- [59] L. Diana and J. M. Kahn, "Rate-adaptive modulation techniques for infrared wireless communications," in *IEEE International Conference on Communications*, vol. 1, 1999, pp. 597–603.
- [60] T. OHTSUKI, "Rate-adaptive indoor infrared wireless communication systems using repeated and punctured convolutional codes," *IEEE Communications Letters*, vol. 4, no. 2, pp. 56–58, 2000.
- [61] J. M. Garrido-Balsells, A. Jurado-Navas, M. Castillo-Vázquez, and A. Puerta-Notario, "Improving RCPC codes in rate-adaptive optical wireless communications systems," *Wireless Personal Communications*, vol. 69, no. 2, pp. 879–889, Mar. 2013.
- [62] N. Yamamoto and T. Ohtsuki, "Iterative MAP decoding of turbo coded OOK and turbo coded BPPM," in *IEEE Global Telecommunications Conference*, vol. 3, 2001, pp. 1913–1917.
- [63] J. Y. Kim, "Delay-throughput performance of a DS/CDMA packet radio network in an indoor infrared wireless channel," in *IEEE Vehicular Technology Conference*, vol. 7, 2004, pp. 4938–4941.
- [64] I. B. Djordjevic and H. G. Batshon, "LDPC-coded OFDM for heterogeneous access optical networks," *IEEE Photonics Journal*, vol. 2, no. 4, pp. 611–619, 2010.
- [65] T. Koike-Akino, Y. Wang, S. C. Draper, K. Sugihara, and W. Matsumoto, "Bit-interleaved polar-coded OFDM for low-latency M2M wireless communications," in *IEEE International Conference on Communications (ICC)*, May. 2017, pp. 1–7.

- [66] J. Fang, Z. Che, Z. L. Jiang, X. Yu, S.-M. Yiu, K. Ren, X. Tan, and Z. Chen, "An efficient flicker-free FEC coding scheme for dimmable visible light communication based on polar codes," *IEEE Photonics Journal*, vol. 9, no. 3, pp. 1–10, 2017.
- [67] H. Wang and S. Kim, "Dimming control systems with polar codes in visible light communication," *IEEE Photonics Technology Letters*, vol. 29, no. 19, pp. 1651–1654, 2017.
- [68] —, "Design of polar codes for run-length limited codes in visible light communications," *IEEE Photonics Technology Letters*, vol. 31, no. 1, pp. 27–30, 2018.
- [69] —, "Adaptive puncturing method for dimming in visible light communication with polar codes," *IEEE Photonics Technology Letters*, vol. 30, no. 20, pp. 1780–1783, 2018.
- [70] —, "Decoding of polar codes for intersymbol interference in visible-light communication," *IEEE Photonics Technology Letters*, vol. 30, no. 12, pp. 1111–1114, 2018.
- [71] E. N. Mambou, T. Tonnellier, S. A. Hashemi, and W. J. Gross, "Efficient flicker-free FEC codes using knuth's balancing algorithm for vlc," in *IEEE Global Communications Conference (GLOBECOM)*, 2019, pp. 1–6.
- [72] K. Wu, J. He, J. Ma, and Y. Wei, "A BIPCM scheme based on OCT precoding for a 256-QAM OFDM-VLC system," *IEEE Photonics Technology Letters*, vol. 30, no. 21, pp. 1866–1869, Nov. 2018.
- [73] X. Yan, J. He, Y. Liu, J. Shi, Z. Zhou, J. Ma, and Q. Tang, "A polar-coded MIMO-OFDM scheme for VLC system," in *Asia Communications and Photonics Conference*. Optical Society of America, 2019, pp. M4A–55.
- [74] Z. Gao, Y. Wang, X. Liu, and F. Zhou, "Using polar codes in NOMA-enabled visible light communication systems," *IEEE Sensors Letters*, vol. 3, no. 5, pp. 1–4, 2019.
- [75] Z. Che, J. Fang, Z. L. Jiang, J. Li, S. Zhao, Y. Zhong, and Z. Chen, "A physical-layer secure coding scheme for indoor visible light communication based on polar codes," *IEEE Photonics Journal*, vol. 10, no. 5, pp. 1–13, 2018.
- [76] R. Mori and T. Tanaka, "Performance and construction of polar codes on symmetric binary-input memoryless channels," in *IEEE International Symposium on Information Theory*, 2009, pp. 1496–1500.
- [77] —, "Performance of polar codes with the construction using density evolution," *IEEE Communications Letters*, vol. 13, no. 7, pp. 519–521, 2009.
- [78] I. Tal and A. Vardy, "How to construct polar codes," *IEEE Transactions on Information Theory*, vol. 59, no. 10, pp. 6562–6582, 2013.
- [79] P. Trifonov, "Efficient design and decoding of polar codes," *IEEE Transactions on Communications*, vol. 60, no. 11, pp. 3221–3227, 2012.
- [80] D. Wu, Y. Li, and Y. Sun, "Construction and block error rate analysis of polar codes over AWGN channel based on Gaussian approximation," *IEEE Communications Letters*, vol. 18, no. 7, pp. 1099–1102, 2014.
- [81] J. Dai, K. Niu, Z. Si, C. Dong, and J. Lin, "Does Gaussian approximation work well for the long-length polar code construction?" *IEEE Access*, vol. 5, pp. 7950–7963, 2017.

- [82] E. Arıkan, "Systematic polar coding," *IEEE Communications Letters*, vol. 15, no. 8, pp. 860–862, 2011.
- [83] H. Vangala, Y. Hong, and E. Viterbo, "Efficient algorithms for systematic polar encoding," *IEEE Communications Letters*, vol. 20, no. 1, pp. 17–20, 2015.
- [84] I. Tal and A. Vardy, "List decoding of polar codes," *IEEE Transactions on Information Theory*, vol. 61, no. 5, pp. 2213–2226, 2015.
- [85] K. Niu and K. Chen, "CRC-aided decoding of polar codes," *IEEE Communications Letters*, vol. 16, no. 10, pp. 1668–1671, 2012.
- [86] —, "Stack decoding of polar codes," *Electronics Letters*, vol. 48, no. 12, pp. 695–697, 2012.
- [87] O. Afisiadis, A. Balatsoukas-Stimming, and A. Burg, "A low-complexity improved successive cancellation decoder for polar codes," in *IEEE Asilomar Conference on Signals, Systems and Computers*, 2014, pp. 2116–2120.
- [88] G. Sarkis, P. Giard, A. Vardy, C. Thibeault, and W. J. Gross, "Fast polar decoders: Algorithm and implementation," *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 5, pp. 946–957, 2014.
- [89] M. Hanif and M. Ardakani, "Fast successive-cancellation decoding of polar codes: Identification and decoding of new nodes," *IEEE Communications Letters*, vol. 21, no. 11, pp. 2360–2363, 2017.
- [90] C. Condo, V. Bioglio, and I. Land, "Generalized fast decoding of polar codes," in *IEEE Global Communications Conference (GLOBECOM)*, 2018, pp. 1–6.
- [91] P. Giard, A. Balatsoukas-Stimming, G. Sarkis, C. Thibeault, and W. J. Gross, "Fast low-complexity decoders for low-rate polar codes," *Journal of Signal Processing Systems*, vol. 90, no. 5, pp. 675–685, 2018.
- [92] F. Ercan, C. Condo, and W. J. Gross, "Reduced-memory high-throughput fast-SSC polar code decoder architecture," in *IEEE International Workshop on Signal Processing Systems (SiPS)*, 2017, pp. 1–6.
- [93] F. Ercan, T. Tonnellier, C. Condo, and W. J. Gross, "Operation merging for hardware implementations of fast polar decoders," *Journal of Signal Processing Systems*, vol. 91, no. 9, pp. 995–1007, 2019.
- [94] E. Arıkan, "A performance comparison of polar codes and Reed-Muller codes," *IEEE Communications Letters*, vol. 12, no. 6, pp. 447–449, 2008.
- [95] A. Elkelesh, M. Ebada, S. Cammerer, and S. Ten Brink, "Belief propagation list decoding of polar codes," *IEEE Communications Letters*, vol. 22, no. 8, pp. 1536–1539, 2018.
- [96] K. Niu, K. Chen, and J. Lin, "Low-complexity sphere decoding of polar codes based on optimum path metric," *IEEE Communications Letters*, vol. 18, no. 2, pp. 332–335, 2014.
- [97] S. A. Hashemi, C. Condo, and W. J. Gross, "List sphere decoding of polar codes," in *IEEE Asilomar Conference on Signals, Systems and Computers*, 2015, pp. 1346–1350.
- [98] D. Wu, Y. Li, X. Guo, and Y. Sun, "Ordered statistic decoding for short polar codes," *IEEE Communications Letters*, vol. 20, no. 6, pp. 1064–1067, 2016.

- [99] U. U. Fayyaz and J. R. Barry, "Low-complexity soft-output decoding of polar codes," *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 5, pp. 958–966, 2014.
- [100] M. Seidl and J. B. Huber, "Improving successive cancellation decoding of polar codes by usage of inner block codes," in *IEEE International Symposium on Turbo Codes & Iterative Information Processing*, 2010, pp. 103–106.
- [101] J. Guo, M. Qin, A. G. i Fabregas, and P. H. Siegel, "Enhanced belief propagation decoding of polar codes through concatenation," in *IEEE International Symposium on Information Theory*, 2014, pp. 2987–2991.
- [102] Y. Wang, K. R. Narayanan, and Y.-C. Huang, "Interleaved concatenations of polar codes with BCH and convolutional codes," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 2, pp. 267–277, 2015.
- [103] 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 Transactions on Information Theory*, vol. 57, no. 2, pp. 803–834, 2011.
- [104] S. Kudekar, T. Richardson, and R. L. Urbanke, "Spatially coupled ensembles universally achieve capacity under belief propagation," *IEEE Transactions on Information Theory*, vol. 59, no. 12, pp. 7761–7813, 2013.
- [105] Y. Xie, L. Yang, P. Kang, and J. Yuan, "Euclidean geometry-based spatially coupled LDPC codes for storage," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 9, pp. 2498–2509, 2016.
- [106] A. R. Iyengar, P. H. Siegel, R. L. Urbanke, and J. K. Wolf, "Windowed decoding of spatially coupled codes," *IEEE Transactions on Information Theory*, vol. 59, no. 4, pp. 2277–2292, 2012.
- [107] P. Kang, Y. Xie, L. Yang, and J. Yuan, "Reliability-based windowed decoding for spatially coupled LDPC codes," *IEEE Communications Letters*, vol. 22, no. 7, pp. 1322–1325, 2018.
- [108] X. Wu, L. Yang, and J. Yuan, "Information coupled polar codes," in *IEEE International Symposium on Information Theory (ISIT)*, 2018, pp. 861–865.
- [109] X. Wu, L. Yang, Y. Xie, and J. Yuan, "Partially information coupled polar codes," *IEEE Access*, vol. 6, pp. 63 689–63 702, Sep. 2018.
- [110] R. G. Gallager, *Information theory and reliable communication*. Springer, 1968, vol. 2.
- [111] Z. Liu, K. Chen, K. Niu, and Z. He, "Distance spectrum analysis of polar codes," in *IEEE Wireless Communications and Networking Conference (WCNC)*, 2014, pp. 490–495.
- [112] A. Balatsoukas-Stimming, M. B. Parizi, and A. Burg, "LLR-based successive cancellation list decoding of polar codes," *IEEE Transactions on Signal Processing*, vol. 63, no. 19, pp. 5165–5179, 2015.
- [113] C. Leroux, I. Tal, A. Vardy, and W. J. Gross, "Hardware architectures for successive cancellation decoding of polar codes," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2011, pp. 1665–1668.
- [114] T. I. 802.16e LDPC Encoder/Decoder Core. [Online]. Available: [http://www.turbobest.com/tb\\_ldpc80216e.htm](http://www.turbobest.com/tb_ldpc80216e.htm).



- [115] A. Alamdar-Yazdi and F. R. Kschischang, "A simplified successive-cancellation decoder for polar codes," *IEEE Communications Letters*, vol. 15, no. 12, pp. 1378–1380, 2011.
- [116] 2017. [Online]. Available: [https://en.wikipedia.org/wiki/Laser\\_safety](https://en.wikipedia.org/wiki/Laser_safety).
- [117] E. A. Lee and D. G. Messerschmitt, *Digital communication*. Springer Science & Business Media, 2012.
- [118] T. Koike-Akino, Y. Wang, S. C. Draper, K. Sugihara, W. Matsumoto, D. S. Millar, K. Parsons, and K. Kojima, "Bit-interleaved polar-coded modulation for low-latency short-block transmission," in *Optical Fiber Communication Conference*. Optical Society of America, 2017, pp. W1J–6.
- [119] G. Sarkis and W. J. Gross, "Increasing the throughput of polar decoders," *IEEE Communications Letters*, vol. 17, no. 4, pp. 725–728, 2013.
- [120] G. Sarkis, P. Giard, A. Vardy, C. Thibeault, and W. J. Gross, "Increasing the speed of polar list decoders," in *IEEE Workshop on Signal Processing Systems (SiPS)*, October 2014, pp. 1–6.
- [121] ———, "Fast list decoders for polar codes," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 2, pp. 318–328, 2016.
- [122] M. El-Khamy, H. Mahdavifar, G. Feygin, J. Lee, and I. Kang, "Relaxed polar codes," *IEEE Transactions on Information Theory*, vol. 63, no. 4, pp. 1986–2000, 2017.
- [123] T. Koike-Akino, C. Cao, Y. Wang, S. C. Draper, D. S. Millar, K. Kojima, K. Parsons, L. Galdino, D. J. Elson, D. Lavery *et al.*, "Irregular polar coding for complexity-constrained lightwave systems," *Journal of Lightwave Technology*, vol. 36, no. 11, pp. 2248–2258, 2018.
- [124] K. Chen, B. Li, H. Shen, J. Jin, and D. Tse, "Reduce the complexity of list decoding of polar codes by tree-pruning," *IEEE Communications Letters*, vol. 20, no. 2, pp. 204–207, 2016.
- [125] K. Chen, K. Niu, and J. Lin, "Improved successive cancellation decoding of polar codes," *IEEE Transactions on Communications*, vol. 61, no. 8, pp. 3100–3107, 2013.
- [126] J. Chen, Y. Fan, C. Xia, C.-Y. Tsui, J. Jin, K. Chen, and B. Li, "Low-complexity list successive-cancellation decoding of polar codes using list pruning," in *IEEE Global Communications Conference (GLOBECOM)*, December 2016, pp. 1–6.
- [127] B. Li, H. Shen, and D. Tse, "An adaptive successive cancellation list decoder for polar codes with cyclic redundancy check," *IEEE Communications Letters*, vol. 16, no. 12, pp. 2044–2047, 2012.
- [128] P. Trifonov, "Reduced complexity decoding of polar codes with reed-solomon kernel," in *Information Theory and Applications Workshop (ITA)*. San Diego, CA, Feb Feb 2018, pp. 1–9.
- [129] H. Aurora, C. Condo, and W. J. Gross, "Low-complexity software stack decoding of polar codes," in *IEEE International Symposium on Circuits and Systems (ISCAS)*. Florence, Italy, May May 2018, pp. 1–5.
- [130] J. Guo, Z. Shi, Z. Liu, Z. Zhang, and Q. Liu, "Multi-CRC polar codes and their applications," *IEEE Communications Letters*, vol. 20, no. 2, pp. 212–215, 2016.
- [131] Z. Zhang and L. Zhang, "A split-reduced successive cancellation list decoder for polar codes," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 2, pp. 292–302, Feb. 2016.

- [132] S. A. Hashemi, C. Condo, and W. J. Gross, "A fast polar code list decoder architecture based on sphere decoding," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 63, no. 12, pp. 2368–2380, Dec. 2016.
- [133] P. Trifonov and V. Miloslavskaya, "Polar subcodes," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 2, pp. 254–266, Feb. 2016.
- [134] P. Trifonov, "Star polar subcodes," in *IEEE Wireless Communication and Networking Conference Workshops*, Mar. 2017, pp. 1–6.
- [135] M. El-Khamy, J. Lee, and I. Kang, "Detection analysis of CRC-assisted decoding," *IEEE Communication Letter*, vol. 19, no. 3, pp. 483–486, Mar. 2015.
- [136] S. A. Hashemi, C. Condo, F. Ercan, and W. J. Gross, "Memory-efficient polar decoders," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 7, no. 4, pp. 604–615, Dec. 2017.
- [137] "3GPP TS RAN 38.212 v1.2.1," Dec. 2017. [Online]. Available: [http://www.3gpp.org/ftp/Specs/archive/38\\_series/38.212/38212-f30.zip](http://www.3gpp.org/ftp/Specs/archive/38_series/38.212/38212-f30.zip).
- [138] 3GPP, "Final report of 3GPP TSG RAN WG1 #87 v1.0.0," Reno, USA, Nov. 2016. [Online]. Available: [http://www.3gpp.org/ftp/tsg\\_ran/WG1\\_RL1/TSGR1\\_87](http://www.3gpp.org/ftp/tsg_ran/WG1_RL1/TSGR1_87).
- [139] K. Niu, K. Chen, J. Lin, and Q. Zhang, "Polar codes: Primary concepts and practical decoding algorithms," *IEEE Communications Magazine*, vol. 52, no. 7, pp. 192–203, 2014.
- [140] C. Zhang, B. Yuan, and K. K. Parhi, "Reduced-latency SC polar decoder architectures," in *IEEE International Conference on Communications (ICC)*, Jun. 2012, pp. 3471–3475.
- [141] B. Yuan and K. K. Parhi, "Low-latency successive-cancellation polar decoder architectures using 2-Bit decoding," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 61, no. 4, pp. 1241–1254, April. 2014.
- [142] A. Mishra, A. J. Raymond, L. G. Amaru, G. Sarkis, C. Leroux, P. Meinerzhagen, A. Burg, and W. J. Gross, "A successive cancellation decoder ASIC for a 1024-bit polar code in 180nm CMOS," in *IEEE Asian Solid-State Circuits Conference*, Nov. 2012, pp. 205–208.
- [143] B. Yuan and K. K. Parhi, "Reduced-latency LLR-based SC list decoder for polar codes," in *the 25th edition of the ACM Great Lakes Symposium on VLSI*, May. 2015, pp. 107–110.
- [144] —, "Low-latency successive-cancellation list decoders for polar codes with multibit decision," *IEEE Transactions on Very Large Scale Integration*, vol. 23, no. 10, pp. 2268–2280, Oct. 2015.
- [145] C. Husmann, P. C. Nikolaou, and K. Nikitopoulos, "Reduced latency ML polar decoding via multiple sphere-decoding tree searches," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 2, pp. 1835–1839, Feb. 2017.
- [146] S. Li, Y. Deng, L. Lu, J. Liu, and T. Huang, "A low-latency simplified successive cancellation decoder for polar codes based on node error probability," *IEEE Communications Letters*, vol. 22, no. 12, pp. 2439–2442, Dec. 2018.

- [147] H. Sun, R. Liu, and C. Gao, "A simplified decoding method of polar codes based on hypothesis testing," *IEEE Communications Letters*, pp. 1–1, Jan. 2020.
- [148] R. Silverman and M. Balser, "Coding for constant-data-rate systems," *IEEE Transactions of the IRE Professional Group on Information Theory*, vol. 4, no. 4, pp. 50–63, 1954.
- [149] C. Leroux, A. J. Raymond, G. Sarkis, and W. J. Gross, "A semi-parallel successive-cancellation decoder for polar codes," *IEEE Transactions on Signal Processing*, vol. 61, no. 2, pp. 289–299, 2012.
- [150] H. Zheng, A. Balatsoukas-Stimming, Z. Cao, and A. M. J. Koonen, "Implementation of a high-throughput Fast-SSC polar decoder with sequence repetition node," *arXiv:2007.11394*, to appear in IEEE International Workshop on Signal Processing Systems, Oct. 2020 (SIPS2020).
- [151] H. Zheng, S. A. Hashemi, A. Balatsoukas-Stimming, Z. Cao, A. M. J. Koonen, J. Cioffi, and A. Goldsmith, "Threshold-based fast successive-cancellation decoding of polar codes," *arXiv:2005.04394*, 2020.
- [152] J. Ma, J. He, J. Shi, K. Wu, M. Chen, Z. Zhou, and Y. Xiao, "Performance enhanced 256-QAM BIPCM-DMT system enabled by CAZAC precoding," *Journal of Lightwave Technology*, vol. 38, no. 3, pp. 557–563, Feb. 2020.
- [153] M. Seidl, A. Schenk, C. Stierstorfer, and J. B. Huber, "Polar-coded modulation," *IEEE Transactions on Communications*, vol. 61, no. 10, pp. 4108–4119, Oct. 2013.
- [154] K. A. Mekonnen, J. H. van Zantvoort, N. Calabretta, N. Tessema, E. Tangdiongga, and T. Koonen, "High-capacity dynamic indoor network employing optical-wireless and 60-ghz radio techniques," *J. Lightw. Technol.*, vol. 36, no. 10, pp. 1851–1861, May. 2018.
- [155] P. Giard, G. Sarkis, C. Thibeault, and W. J. Gross, "237 gbit/s unrolled hardware polar decoder," *Electronics Letters*, vol. 51, no. 10, pp. 762–763, 2015.
- [156] Y.-L. Ueng, C.-J. Yeh, M.-C. Lin, and C.-L. Wang, "Turbo coded multiple-antenna systems for near-capacity performance," *IEEE Journal on Selected Areas in Communications*, vol. 27, no. 6, pp. 954–964, 2009.
- [157] S. Haykin, M. Sellathurai, Y. De Jong, and T. Willink, "Turbo-MIMO for wireless communications," *IEEE Communications Magazine*, vol. 42, no. 10, pp. 48–53, 2004.
- [158] H. Nikopour and H. Baligh, "Sparse code multiple access," in *IEEE International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, 2013, pp. 332–336.
- [159] X. Wang and H. V. Poor, "Iterative (turbo) soft interference cancellation and decoding for coded CDMA," *IEEE Transactions on communications*, vol. 47, no. 7, pp. 1046–1061, 1999.
- [160] Y. Wu, S. Zhang, and Y. Chen, "Iterative multiuser receiver in sparse code multiple access systems," in *IEEE International Conference on Communications (ICC)*, 2015, pp. 2918–2923.
- [161] J. Dai, K. Niu, and J. Lin, "Polar-coded MIMO systems," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 7, pp. 6170–6184, 2018.
- [162] J. Dai, K. Niu, Z. Si, C. Dong, and J. Lin, "Polar-coded non-orthogonal multiple access," *IEEE Transactions on Signal Processing*, vol. 66, no. 5, pp. 1374–1389, 2017.
- [163] K. Chen, K. Niu, and J. Lin, "A hybrid ARQ scheme based on polar codes," *IEEE Communications Letters*, vol. 17, no. 10, pp. 1996–1999, 2013.

- 
- [164] S.-N. Hong, D. Hui, and I. Marić, "Capacity-achieving rate-compatible polar codes," *IEEE Transactions on Information Theory*, vol. 63, no. 12, pp. 7620–7632, 2017.
- [165] S.-N. Hong and M.-O. Jeong, "An efficient construction of rate-compatible punctured polar (RCPP) codes using hierarchical puncturing," *IEEE Transactions on Communications*, vol. 66, no. 11, pp. 5041–5052, 2018.
- [166] R. Wang, J. Honda, H. Yamamoto, R. Liu, and Y. Hou, "Construction of polar codes for channels with memory," in *IEEE Information Theory Workshop-Fall (ITW)*, 2015, pp. 187–191.
- [167] B. Bourassa, M. Tremblay, and D. Poulin, "Convolutional polar codes on channels with memory," *arXiv preprint arXiv:1805.09378 [cs, math]*, 2018.
- [168] B. Shuval and I. Tal, "Fast polarization for processes with memory," *IEEE Transactions on Information Theory*, vol. 65, no. 4, pp. 2004–2020, 2018.
- [169] E. Şaşoğlu and L. Wang, "Universal polarization," *IEEE Transactions on Information Theory*, vol. 62, no. 6, pp. 2937–2946, 2016.
- [170] B. Shuval and I. Tal, "List decoding of universal polar codes," *arXiv preprint arXiv:2001.03784*, 2020.



---

## APPENDIX

---

### Proof of Proposition 6.3.1

Let  $I_k$  denote the  $k \times k$  identity matrix for  $k \geq 1$ . Since the source node is the rightmost node in an SR node, the  $g$  function calculation in (2.44) can be used as

$$\begin{aligned} \alpha_r^E [1 : 2^r] &= \alpha_j^i [1 : 2^j] \times \left( I_{2^{j-1}} \otimes ((-1)^{\eta_{j-1}}, 1)^T \right) \times \left( I_{2^{j-2}} \otimes ((-1)^{\eta_{j-2}}, 1)^T \right) \\ &\quad \times \cdots \times \left( I_{2^r} \otimes ((-1)^{\eta_r}, 1)^T \right). \end{aligned}$$

Using the identity  $(A \otimes B) \times (C \otimes D) = (A \times C) \otimes (B \times D)$  with  $A = I_{2^{j-2}}$ ,  $B = I_2 \otimes ((-1)^{\eta_{j-1}}, 1)^T$ ,  $C = I_{2^{j-2}}$ , and  $D = ((-1)^{\eta_{j-2}}, 1)^T$  results in

$$\begin{aligned} \alpha_r^E [1 : 2^r] &= \alpha_j^i [1 : 2^j] \times \left[ (I_{2^{j-2}} \times I_{2^{j-2}}) \otimes \left( I_2 \otimes ((-1)^{\eta_{j-1}}, 1)^T \times ((-1)^{\eta_{j-2}}, 1)^T \right) \right] \\ &\quad \times \left( I_{2^{j-3}} \otimes ((-1)^{\eta_{j-3}}, 1)^T \right) \times \cdots \times \left( I_{2^r} \otimes ((-1)^{\eta_r}, 1)^T \right), \end{aligned}$$

which can be written as

$$\begin{aligned} \alpha_r^E [1 : 2^r] &= \alpha_j^i [1 : 2^j] \times I_{2^{j-2}} \otimes \left( ((-1)^{\eta_{j-2}}, 1)^T \otimes ((-1)^{\eta_{j-1}}, 1)^T \right) \\ &\quad \times \left( I_{2^{j-3}} \otimes ((-1)^{\eta_{j-3}}, 1)^T \right) \times \cdots \times \left( I_{2^r} \otimes ((-1)^{\eta_r}, 1)^T \right), \end{aligned}$$

where the identity  $I_2 \otimes (a_1, \dots, a_k)^T \times (b_1, b_2)^T = (b_1, b_2)^T \otimes (a_1, \dots, a_k)^T$  is used. Repeating the above procedures results in

$$\begin{aligned} \alpha_r^E [1 : 2^r] &= \alpha_j^i [1 : 2^j] \times I_{2^r} \otimes \left( ((-1)^{\eta_r}, 1)^T \otimes \cdots \otimes ((-1)^{\eta_{j-1}}, 1)^T \right) \\ &= \alpha_j^i [1 : 2^j] \times I_{2^r} \otimes \left( (-1)^{s_l[1]}, (-1)^{s_l[2]}, \dots, (-1)^{s_l[2^{j-r}]} \right)^T. \end{aligned}$$

Thus for  $k \in \{1, \dots, 2^r\}$ ,

$$\alpha_{r_l}^E [k] = \sum_{m=1}^{2^{j-r}} \alpha_j^i \left[ (k-1)2^{j-r} + m \right] (-1)^{s_l[m]}.$$

## Explanation of Observation 1

To explain Observation 1, a lemma is first introduced as follows.

**Lemma 1.** *Under the Gaussian approximation assumption in (6.11), for any node  $\mathcal{N}_j^i$  whose  $2^j$  bits undergo a hard decision in (6.15), assuming all prior bits are decoded correctly, the probability of correct decoding can be calculated as  $\left(\frac{\tilde{P}_c}{\tilde{P}_e + \tilde{P}_c}\right)^{2^j}$ .*

*Proof.* In accordance with Fig. 6.5, for any node  $\mathcal{N}_j^i$ , considering all the previous bits are decoded correctly, the probability that the  $k$ -th bit ( $1 \leq k \leq 2^j$ ) in the node undergoes a hard decision is  $\tilde{P}_c + \tilde{P}_e$ . Moreover, The probability of a correct hard decision for the  $k$ -th bit in the node is  $\tilde{P}_c$ , regardless of the value of  $\beta_j^i[k]$ . Thus, the conditional probability that a hard decision on the  $k$ -th bit is correct given that the  $k$ -th bit undergoes a hard decision is  $\frac{\tilde{P}_c}{\tilde{P}_e + \tilde{P}_c}$ . Since the LLR values of bits in a node are independent of each other, the conditional probability that hard decisions on all the  $2^j$  bits of node  $\mathcal{N}_j^i$  are correct given that all its  $2^j$  bits undergo hard decisions can be calculated as  $\left(\frac{\tilde{P}_c}{\tilde{P}_e + \tilde{P}_c}\right)^{2^j}$ .  $\square$

To have a probability of correct decoding of at least  $\varepsilon$  for all the nodes that undergo hard decision in a polar code of length  $2^n$ , any such node  $\mathcal{N}_j^i$  is assumed to have the probability of correct decoding of at least  $2^{(n-j)}\sqrt{\varepsilon}$ . Therefore and by using the result in Lemma 1, we have

$$\left(\frac{\tilde{P}_c}{\tilde{P}_e + \tilde{P}_c}\right)^{2^j} \geq 2^{(n-j)}\sqrt{\varepsilon}, \quad (1)$$

which is equivalent to

$$\frac{\tilde{P}_c}{\tilde{P}_e + \tilde{P}_c} \geq 2^n\sqrt{\varepsilon}. \quad (2)$$

If  $m_j^i \leq 2c^2$ , then  $T = -m_j^i + c\sqrt{2m_j^i}$ ,  $\tilde{P}_c = Q\left(c - \sqrt{2m_j^i}\right)$ , and  $\tilde{P}_e = Q(c)$ . When  $0.5 < \varepsilon < 1$ , (2) can be written as

$$\frac{1}{2} \left[ c - Q^{-1} \left( \frac{Q(c)}{\frac{1}{2^n\sqrt{\varepsilon}} - 1} \right) \right]^2 \leq m_j^i \leq 2c^2, \quad (3)$$

which requires

$$Q(c) \leq 1 - 2^n\sqrt{\varepsilon}. \quad (4)$$



If  $m_j^i \geq 2c^2$ , then  $T = m_j^i - c\sqrt{2m_j^i}$ ,  $\tilde{P}_c = Q(-c)$ , and  $\tilde{P}_e = Q(\sqrt{2m_j^i} - c)$ . Thus (2) can be written as

$$m_j^i \geq \max \left\{ 2c^2, \frac{1}{2} \left[ c + Q^{-1} \left( \left( \frac{1}{\sqrt[2^n]{\varepsilon}} - 1 \right) Q(-c) \right) \right]^2 \right\}. \quad (5)$$

If (4) holds and by using the fact that  $Q(-c) = 1 - Q(c)$ , then

$$2c^2 \geq \frac{1}{2} \left[ c + Q^{-1} \left( \left( \frac{1}{\sqrt[2^n]{\varepsilon}} - 1 \right) Q(-c) \right) \right]^2. \quad (6)$$

Thus  $m_j^i \geq 2c^2$ , which always holds based on the initial assumption. Therefore, it suffices that

$$m_j^i \geq \frac{1}{2} \left[ c - Q^{-1} \left( \frac{Q(c)}{\frac{1}{\sqrt[2^n]{\varepsilon}} - 1} \right) \right]^2, \quad (7)$$

and (4) to ensure (2). In other words, assuming all previous bits are decoded correctly, the probability that any node  $\mathcal{N}_j^i$  that undergo hard decision (6.15) in the decoding process is decoded correctly is lower bounded by  $\sqrt[2^{(n-j)}]{\varepsilon}$  if (4) and (7) are satisfied.

---

## LIST OF PUBLICATIONS

---

1. **H. Zheng**, B. Chen, L. F. Abanto-Leon, Z. Cao and T. Koonen, "Complexity-adjustable SC decoding of polar codes for energy consumption reduction," in *IET Communications*, vol. 13, no. 14, pp. 2088-2096, Aug. 2019.
2. **H. Zheng**, S. A. Hashemi, B. Chen, Z. Cao and A. M. J. Koonen, "Inter-Frame Polar Coding with Dynamic Frozen Bits," in *IEEE Communications Letters*, vol. 23, no. 9, pp. 1462-1465, Sept. 2019.
3. **H. Zheng**, S. A. Hashemi, Z. Cao, A. M. J. Koonen, J. Cioffi, A. Goldsmith, "Threshold-Based Successive-Cancellation Decoding of Polar Codes", *IEEE International Conference on Communications (ICC)*, Dublin, Ireland, Jun. 2020.
4. **H. Zheng**, A. Balatsoukas-Stimming, Z. Cao, A. M. J. Koonen, "Implementation of a High-Throughput Fast-SSC Polar Decoder with Sequence Repetition Node", *IEEE International Workshop on Signal Processing Systems (SIPS)*, Portugal, Oct. 2020.
5. **H. Zheng**, S. A. Hashemi, A. Balatsoukas-Stimming, Z. Cao, A. M. J. Koonen, J. Cioffi, A. Goldsmith, "Fast Successive-Cancellation Decoding of Polar Codes", *IEEE Transactions on Communications*, major revision.
6. **H. Zheng**, A. Balatsoukas-Stimming, Z. Cao, A. M. J. Koonen, "Experimental demonstration of 9.6 Gbit/s polar coded infrared wireless communication system", *IEEE Photonics Technology Letter*, accepted.



---

## CURRICULUM VITAE

---

Haotian Zheng was born on 14-03-1990 in Xiangyang, China. In 2012, he received a B.S. degree in Information and Communication Engineering from Nanjing University of Science and Technology, Nanjing, China. In 2015, he earned the M.E. degree in Communication and Information Systems at National Digital Switching System Engineering and Technological Research Center (NDSC), Zhengzhou, China. From 2015 to 2016, he worked in NDSC as a Research Engineer. In October 2016, he started working as a PhD student in the Electro-Optical Communications (ECO) group of the Department of Electrical and Engineering, Eindhoven University of Technology. The most important results of his research during the PhD period are described in this thesis.



---

## ACKNOWLEDGMENTS

---

I started my Ph.D. at Eindhoven University of Technology on 29th October 2016. Herewith, I would like to express my greatest appreciation to all of them who shared with this journey.

First of all, I would like to thank my supervisors, especially to my first promoter Prof. Ton Koonen, for the opportunity to do a Ph.D. in ECO group. I adore you deeply for your academic achievements and personal charisma.

I would like to express my deep gratitude to my daily supervisor, Dr. Zizheng Cao. You are a person who can see the essence of things and express them clearly. I have learned a lot from you in both research and life. Your precious guidance, comments and encouragement helped me become a real scientific researcher and I will remember for my lifetime.

I would like to thank my co-promotor, Dr. Alexios Balatsoukas Stimming. I am honored that I am the first Ph.D. under your supervision in TU/e. You gave me the example of precise, meticulous and professional scholar which I admire to be. You are so smart that I only said 50% and you have already understood 100% in most cases when we discussed. I enjoy discussing and working with you. Your encouragement always gave me confidence and helped me through the most difficult period of my research.

I would like to thank Dr. Seyyed Ali Hashemi. Our intersection started when I first read your paper. I thought this guy is a genius and sent the first email to discuss something that have confused me for a long time. Your warm and detailed reply soon made it clear to me. Then I was lucky to have more deeper cooperation with you. You are always patient in answering my questions and your insightful suggestions always enlightened me. I can't get where I am now without your help. A great pity that the epidemic hindered us in our first meeting. Wish you a great success in Stanford.

I would like to thank Dr. Bin Chen. You led me to the study of polar codes. Your wealth of knowledge in this area always benefit and inspire me. Our fruitful discussions gave birth to many interesting ideas. I have learned a lot from you both in life and research. Thanks, Bin!

I would like to give my sincere gratitude to the committee of this thesis, Prof.dr. Magnus Karlsson (Chalmers University of Technology), Prof.dr.ir. Frans Willems, Dr. Chigo M. Okonkwo, Dr. Seyyed Ali Hashemi, the reserved member Dr. Alex Alvarado and the chairman Prof.dr.ing. Guus Pemen for being part of my doctorate committee and for the insightful comments of the thesis.

Special thanks go to other experts including Nicola, Oded, Chigo, Patty, Sonia, Georgios and Eduward in ECO group for your support and help, and our dear secretary José Hakkens, for your warm assistance on all the administrative matters in my four-year Ph.D. work.

I would like to thank my colleagues - Xiong, Xuebing, Chao, Luis, Mahir, Ailee, Mehedi, Fu, Bruno, Konstantinou, Yi, Lei, Jianou, Yu (Lei), Ye, Mingyang, Simon, Aref, Joanne, Asterios, Liuyan, Javier, Ketema, Netsanet, Simone, Carina, Ngoc Quan, Alvaro, Sjoerd, and Kristif for your kind help and encouragement.

I would like to thank my friends who gave me an unforgettable four years life in TU/e and make me never feel homesick – Qiang, Yingzhe, Xuwei, Hanglong, Da, Huan, Yixin(Xiaobao), Wang, Shuli, Kexin(Keke), Hui, Lu, Ruoyu(Dudu), Jianzhi, Yanan, Yachen, Tianzong, Teng, Pan, Xinran, Yu (Wang), Bin (Shi), Junyu, Yuchen, Yu (Zhao), Wenjing, Yuwen, Xin, Weibo, Jing, Ping, She, Dan, Bitao, Xiaotao, Shaojuan, Weigang and Fulong.

Last but not least, the thanks come to my parents and girlfriend. I am deeply grateful to your warming love, unconditional support and continuous encouragement. This thesis is dedicated to you all.