NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

AN EFFICIENT METHODOLOGY TO DE-ANONYMIZE THE 5G-NEW RADIO PHYSICAL DOWNLINK CONTROL CHANNEL

by

Benjamin Gardner

June 2020

Thesis Advisor:	John D. Roth
Second Reader:	John C. McEachen

Approved for public release. Distribution is unlimited.

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE	Form Approved OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington, DC 20503.

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE June 2020	3. REPORT TYPE AND DATES COVERED Master's thesis

4. TITLE AND SUBTITLE AN EFFICIENT METHODOLOGY TO DE-ANONYMIZE THE 5G-NEW RADIO PHYSICAL DOWNLINK CONTROL CHANNEL	5. FUNDING NUMBERS
6. AUTHOR(S) Benjamin Gardner	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A	10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release. Distribution is unlimited.	12b. DISTRIBUTION CODE A

13. ABSTRACT (maximum 200 words)

Over the last decade there has been tremendous growth in the number of mobile subscribers across the globe. Along with this growth in mobile device use there has been an accompanying increase in the amount of personally identifiable information that is transmitted and stored by various entities on a daily basis. This presents a significant privacy risk to users. This research paper presents a methodology by which a passive eavesdropper of a network can identify addressing information (i.e., cell-Radio Network Temporary Identifier [RNTI]) assigned to transmissions in the 5G physical downlink control channel. The ability for an eavesdropper to obtain this information degrades the degree to which the transmitted data is anonymized, thereby potentially negatively impacting user's privacy. The presented methodology determines this information by examining patterns in the frozen bit indices generated via the polar encoding and decoding processes and applied scrambling sequences. Simulations examined the effectiveness of the methodology under a variety of signal-to-noise ratio conditions and presented several optimization techniques. Through the presented methodology, this research paper aims to better inform industry and standards bodies of potential privacy vulnerabilities so that the existing mobile standards may be improved upon.

14. SUBJECT TERMS de-anonymization, new radio, 5G, physical downlink control channel, privacy, cell-Radio Network Temporary Identifier, RNTI	15. NUMBER OF PAGES 107
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UU

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release. Distribution is unlimited.

AN EFFICIENT METHODOLOGY TO DE-ANONYMIZE THE 5G-NEW RADIO PHYSICAL DOWNLINK CONTROL CHANNEL

Benjamin Gardner
Major, United States Marine Corps
BSME, University of Wisconsin-Madison, 2008
MS, Systems Engineering, Naval Postgraduate School, 2015

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
June 2020

Approved by: John D. Roth
Advisor

John C. McEachen
Second Reader

Douglas J. Fouts
Chair, Department of Electrical and Computer Engineering

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Over the last decade there has been tremendous growth in the number of mobile subscribers across the globe. Along with this growth in mobile device use there has been an accompanying increase in the amount of personally identifiable information that is transmitted and stored by various entities on a daily basis. This presents a significant privacy risk to users. This research paper presents a methodology by which a passive eavesdropper of a network can identify addressing information (i.e., cell-Radio Network Temporary Identifier [RNTI]) assigned to transmissions in the 5G physical downlink control channel. The ability for an eavesdropper to obtain this information degrades the degree to which the transmitted data is anonymized, thereby potentially negatively impacting user's privacy. The presented methodology determines this information by examining patterns in the frozen bit indices generated via the polar encoding and decoding processes and applied scrambling sequences. Simulations examined the effectiveness of the methodology under a variety of signal-to-noise ratio conditions and presented several optimization techniques. Through the presented methodology, this research paper aims to better inform industry and standards bodies of potential privacy vulnerabilities so that the existing mobile standards may be improved upon.

THIS PAGE INTENTIONALLY LEFT BLANK

Table of Contents

1 Introduction **1**
1.1 Privacy and Anonymization 1
1.2 The Physical Downlink Control Channel 2
1.3 Thesis Chapter Overview 4

2 Background **5**
2.1 PDCCH Encoding Process 5
2.2 PDCCH Decoding Process 19

3 Methodology **33**
3.1 Modified Polar Decoding Process 34
3.2 Error Patterns and Syndromes 37
3.3 Application to PDCCH 40
3.4 Methodology Optimization in Presence of Additive Noise 53

4 Simulation Results and Analysis **65**
4.1 Simulation Variables and Metrics 65
4.2 LCS and Hamming Distance 67
4.3 SNR Levels and Receiver Antenna Quantity 73
4.4 Rate-Matching and Frozen Bit Quantities 74
4.5 Type II Errors 78

5 Summary and Recommendations **81**
5.1 Summary 81
5.2 Recommend Related Future Research Areas 81

Appendix: Supplemental MATLAB Code **85**

List of References **87**

Initial Distribution List **89**

THIS PAGE INTENTIONALLY LEFT BLANK

List of Figures

Figure 1.1 Methodology Employment Concept Illustration. 3

Figure 2.1 PDCCH Encoding Process. 5

Figure 2.2 Relationship between Data and CRC Parity Bits. 9

Figure 2.3 Rate-Matching Schemes 13

Figure 2.4 Rate-Matching Implementation Relationships. 14

Figure 2.5 Bit Pair to Complex Symbol Mapping. 19

Figure 2.6 PDCCH Decoding Process. 20

Figure 2.7 Demodulation LLR Soft Outputs. 21

Figure 2.8 Polar Encoding Example. 26

Figure 2.9 Setup of Polar Decoding Example. 27

Figure 2.10 Polar Decoding Node Types and Computations. 28

Figure 2.11 Completed Polar Decoding Example. 30

Figure 2.12 Example Demodulated LLR Values and Corresponding Bit-Pairs. 31

Figure 3.1 Subset of Matching Syndromes $\mathcal{S}'$ and Respective Scrambling Sequences. 39

Figure 3.2 RNTI Determination Process Flow in Noiseless Channel. 41

Figure 3.3 Comparison of Error Pattern z_0^{N-1} and Matching Syndrome ${z'}_0^{N-1}$. 44

Figure 3.4 Relationship Between CRC Parity Bit p_8 and its Corresponding Data Bits. 47

Figure 3.5 RNTI Determination Process Flow in Presence of Additive Noise. 54

Figure 3.6 Demodulated LLR Values with 8 dB AWGN Present. 56

Figure 3.7 Comparison of Error Pattern z_0^{N-1} and Matching Syndrome $z_0'^{N-1}$ in Presence of Additive Noise. 57

Figure 3.8 Examination of Error Pattern Bits of z_0^{N-1}. 58

Figure 3.9 RNTI Determination Process Flow Incorporating Multiple Antennas in Presence of Additive Noise. 61

Figure 3.10 Multiple-Antenna Received LLR Value Plot. 62

Figure 3.11 Signal Summation LLR Value Plot. 63

Figure 4.1 Simulation Metric Process Flow. 66

Figure 4.2 $P_{Success}$ and $P_{Type\ I}$ as a Function of LCS Threshold for Varying SNR Levels. 68

Figure 4.3 $P_{Type\ I}$ as a Function of LCS Threshold for Varying SNR Levels. . 69

Figure 4.4 $P_{Success}$ and $P_{Type\ I}$ as a Function of Hamming Distance Threshold for Varying SNR Levels. 70

Figure 4.5 $P_{Type\ I}$ as a Function of Hamming Distance Threshold for Varying SNR Levels. 71

Figure 4.6 LCS and Hamming Determination Methodology Comparison. . . 73

Figure 4.7 $P_{Success}$ as a Function of SNR for Varying Antenna Quantities and Determination Methods. Adapted from [1]. 74

Figure 4.8 $P_{Success}$ and $P_{Type\ I}$ as a Function of LCS Threshold for Varying SNR Levels for Both No Rate-Matching and Puncturing Schemes. . . . 76

Figure 4.9 $P_{Success}$ and $P_{Type\ I}$ as a Function of LCS Threshold for Varying SNR Levels for Both No Rate-Matching and Shortening Schemes. . . . 77

Figure 4.10 Detailed Simulation Metric Process Flow. 79

Figure 5.1 Methodology Process Flow in the Presence of Additive Noise. . . 82

Figure 5.2 $P_{Success}$ and $P_{Type\ I}$ as a Function of LCS Threshold for Varying SNR Levels for Both No Rate-Matching and Puncturing Schemes. . . . 83

List of Tables

Table 2.1 Encoding Example Parameters. 6

Table 3.1 Effects of Rate-Matching Schemes on the Quantity of Usable Error Pattern Frozen Indices. 53

Table 4.1 Simulation Variables and Associated Bounds. 65

Table 4.2 Simulation Metrics. 67

Table 4.3 Comparison of LCS and Hamming Method Success Rates for Varying SNR Levels when $P_{\text{Type I}} \leq 0.0001$. 72

Table 4.4 Effects of Rate-Matching Schemes on the Quantity of Usable Error Pattern Frozen Indices. 75

Table 4.5 Comparison of Rate-Matching Scheme Success Rates for Varying SNR Levels when $P_{\text{Type I}} \leq 0.0001$. 78

Table 4.6 Summary of Type II Error Investigation. 80

THIS PAGE INTENTIONALLY LEFT BLANK

List of Acronyms and Abbreviations

3GPP 3rd Generation Partnership Project
AWGN additive white Gaussian noise
C-RNTI cell-radio network temporary identifier
CRC cyclic redundancy check
DCI downlink control information
GF Galois Field
gNB next generation NodeB
ICASSP International Conference on Acoustics, Speech and Signal Processing
IEEE Institute of Electrical and Electronics Engineers
LCS longest common substring
LDPC low density parity check
LLR logarithmic likelihood ratio
Log-SCS logarithmic SCS
NR new radio
PBCH physical broadcast channel
PDCCH physical downlink control channel
PDSCH physical downlink shared channel
PUCCH physical uplink control channel
PUSCH physical uplink shared channel
QPSK quadrature phase shift keying
RNTI radio network temporary identifier
SC successive-cancellation
SCL successive-cancellation list
SCS successive-cancellation stack
SNR signal-to-noise ratio
TA timing advance
UE user equipment

THIS PAGE INTENTIONALLY LEFT BLANK

Acknowledgments

I would like to first and foremost thank my wife, Danielle, and my three children, Jonathan, Jack, and Katherine, for their unyielding support in every endeavor that we embark upon. Your continued encouragement to take risks and relentlessly attack the problem at hand are a large driving force behind everything that we accomplish. Thank you.

Secondly, I would like to thank my advisor, Prof. John Roth. I recognize that you afforded me a great amount of freedom in choosing which path to pursue with this thesis, and I greatly appreciate that. Your uncommon level of patience, meticulous attention to detail, and unrelenting curiosity are admirable traits, and with any luck some of them rubbed off on me through all the time we spent together. Thank you.

Lastly, I would be remiss if I didn't thank the Naval Postgraduate School at large for enabling and facilitating the educational opportunities that have been afforded to me. I can't start to total the number of hours I have spent interacting with fellow students, the faculty, and staff first through the distance-learning program and now as a resident student. Greater than the degrees I have received from NPS, I will forever value the true transformation of myself into a life-long learner no matter which direction the future may take me. To all of the fellow students, faculty, and staff who assisted, guided, and inspired me along the way, thank you.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 1: Introduction

This chapter is adapted from [1], previously published for the 2020 International Conference on Acoustics, Speech and Signal Processing (ICASSP), © 2020 Institute of Electrical and Electronics Engineers (IEEE).

The last decade has witnessed tremendous growth in the quantity of mobile user equipment (UE) used on a daily basis by citizens across the globe. It is estimated that there were over 5.2 billion unique global mobile subscribers in 2019, with subscriber penetration rates ranging from 45% in Sub-Saharan Africa to 86% in Europe [2]. These percentages, and nearly every related metric, are projected to increase into the next decade and beyond. Common smartphones and other mobile devices are routinely used not only for communications but also for navigation, banking and mobile payments, internet browsing, streaming media, recording and sharing pictures and videos, and countless other applications. Users of many common applications do so with a basic expectation that personally identifiable information will remain private from those who do not have a legitimate need to know it. This thesis seeks to examine one of the 5G physical channels, the physical downlink control channel (PDCCH), identify channel processing mechanisms that afford users a certain measure of privacy through anonymization, and develop a methodology that a passive eavesdropper of a network can employ to gain insight into the information being transmitted. The objective of this research is to provide industry and standards organizations with a better understanding of a potential privacy shortcoming that may be present in order to help reduce vulnerabilities in future implementations of mobile technology.

1.1 Privacy and Anonymization

Before the PDCCH is examined, it is constructive to provide further background on the concepts of privacy and anonymization, and how they relate to mobile communications. The dictionary defines privacy as "the quality or state of being apart from company or observation" and "freedom from unauthorized intrusion" [3]. In the context of mobile communications, while there is an expectation that a service provider requires to know a user's location in order to provide reliable cellular service, and a navigation app requires a user's

location in order to provide service, this information is kept private from others. The dictionary further defines anonymize as "to remove identifying information from (something, such as computer data) so that the original source cannot be known" [4]. This has broad application in the context of mobile communications, where those entities who legitimately require personally identifiable information to provide their services are again expected to anonymize the data when practical to protect the users who the service is being provided to.

Along with the growth of global mobile subscribers has come a corresponding growth in the amount of personally identifiable information that is transmitted and collected by various means and entities on a daily basis [5]. While much of this user data is stored anonymously, there has been substantial research into de-anonymization attacks that propose techniques to affix identifying labels to data elements. These techniques encompass the use of location and mobility traces as discussed in [5]–[8]. Given a set of anonymous location traces, it is fairly simple to envision how one would go about identifying patterns in the data (e.g., same path throughout the work week from a residential neighborhood to a commercial sector in the morning, and the reverse path in the evening) and begin making inferences. Similarly, it has previously been shown that a control element quantity, called the timing advance (TA), can provide reasonably accurate user location data [9]–[16]. Limitations to these de-anonymization attack techniques include the need for large data sets (in the case of location and mobility traces) or a method to consistently associate location data with a fixed, albeit anonymous, label.

1.2 The Physical Downlink Control Channel

The PDCCH in 5G is a physical channel that, similar in concept to previous generations of mobile technology, directs UEs to the resources provided by other physical channels as required through the use of downlink control information (DCI) messages. As a basic example, if a UE requested to download a news broadcast, it would receive a DCI message from the base station (referred to as next generation NodeB (gNB)) over the PDCCH that would direct it to specific resources located in the physical downlink shared channel (PDSCH).

Messaging over the PDCCH is unencrypted, and adequate information (e.g., primary and secondary synchronization signals) is available to a passive eavesdropper of a network

through monitoring the physical broadcast channel (PBCH) in order to properly receive DCI messages. As PDCCH traffic is unencrypted, message addressing is implemented in a fashion that is efficient in terms of required resources, as well as preventing UEs other than which the message is intended for from processing the message. Unlike other communications protocols where message addressing is performed by prepending source and destination addresses to a message, PDCCH messaging is achieved by scrambling each transmission with a sequence initialized by a value unique to the destination UE it is being sent to. This unique value is termed a cell-radio network temporary identifier (C-RNTI), which for brevity purposes throughout this paper will be known simply as a radio network temporary identifier (RNTI). It is important to note that while the RNTI value is unique to a UE for a transmission, it is not a permanent value assigned to a UE like the international mobile subscriber identity is. Each gNB continually assigns, recycles, and reuses RNTI values to UEs as they join and leave the cell and during service. The RNTI does serve, however, as a label for a given moment in time that were it to be determined by an eavesdropper can contribute to de-anonymization and loss of privacy in PDCCH messaging.

To this end, this thesis presents the development of a methodology which enables a passive eavesdropper of a network to determine the scrambling sequence applied to a DCI message, and, by extension, the RNTI used to address and label the message. Figure 1.1 illustrates this concept, where each UE is able to decode only its respective PDCCH transmissions, while an eavesdropper is able to receive and decode all PDCCH transmissions intended for the UEs that are legitimate members of the cell.

Figure 1.1. Methodology Employment Concept Illustration.

1.3 Thesis Chapter Overview

The remainder of this thesis is structured as follows. Chapter 2 provides background on the steps of the PDCCH process beginning with the DCI message and working through polar encoding and quadrature phase shift keying (QPSK) modulation, as well as the demodulation and polar decoding process. Chapter 3 describes and details how the standard polar decoding process can be re-purposed in order to determine the scrambling sequence and RNTI associated with the DCI message, subsequently allowing for the accurate descrambling and decoding of the message. Chapter 4 presents simulation results for a variety of implementation cases over a broad range of signal-to-noise ratio (SNR) conditions, as well as presentation of methods which further develop the computational efficiency of the process. Finally, Chapter 5 reinforces the concepts presented throughout the paper with concluding remarks, as well as providing recommendations on potential future research areas.

CHAPTER 2:
Background

This chapter is adapted from [1], previously published for the 2020 ICASSP, © 2020 IEEE.

Before the methodology designed and implemented in Chapter 3 can be presented, an overview of the transmitter and receiver steps for processing a PDCCH message is provided. A thorough understanding of the process during these encoding and decoding steps will facilitate a clear and concise explanation of the modified steps that follow in Chapter 3.

2.1 PDCCH Encoding Process

The PDCCH encoding process is standardized by the 3rd Generation Partnership Project (3GPP) according to several technical specifications [17]–[19], among others. Figure 2.1 provides an illustration of the primary encoding steps.

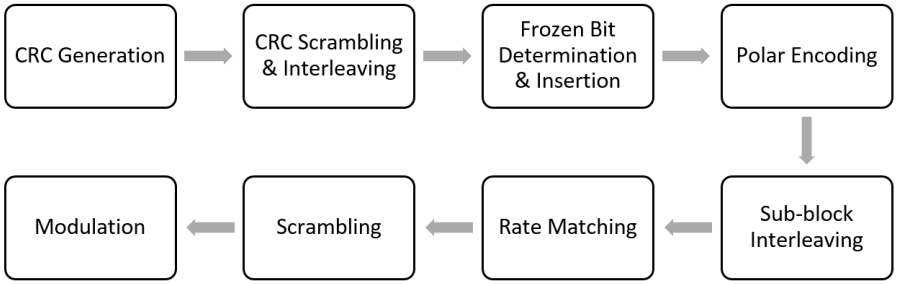


Figure 2.1. PDCCH Encoding Process.

The input to the encoding process is the DCI message to be transmitted to the UE, which can be 12 to 140 bits in length depending on desired functionality and options [18]. The DCI is denoted by a_0^{A-1} where A is the message length.

Throughout this chapter an arbitrarily generated bit sequence representing a DCI message will be processed by each one of the encoding and decoding steps to aid in describing the process. Table 2.1 summarizes additional parameters that will be used throughout the example.

A short message payload length $A = 12$ was selected in order to keep the message from becoming unnecessarily long and cumbersome during presentation of the processes. The RNTI = 100 value was randomly selected from the allowable RNTI values. The rate-matching codeword length $E = 128$ was selected so that the codeword is sufficiently long to effectively illustrate certain processes (e.g., sub-block interleaving) without becoming unnecessarily long. The cell scrambling ID parameter $\eta_{\text{ID}} = 10$, which is used to differentiate cells contained within the same resource grid from each other, was randomly selected from the allowable η_{ID} values.

Table 2.1. Encoding Example Parameters.

Description	Parameter	Value
Message Payload Length	A	12
Radio Network Temporary Identifier	RNTI	100
Rate-matching Codeword Length	E	128
Cell Scrambling ID	η_{ID}	10

The example 12-bit a_0^{A-1} DCI message used is highlighted in green.

1	1	0	1	0	1	1	0	0	1	1	1

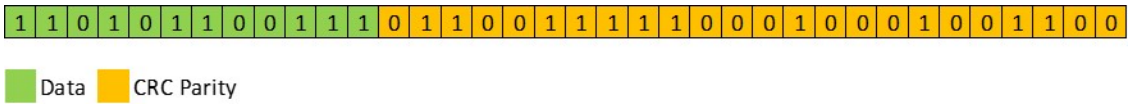
2.1.1 Cyclic Redundancy Check Generation

The cyclic redundancy check (CRC) parity bits provide for error detection for the transmitted DCI messages. The generator polynomial that is standardized for use in the encoding process (see Section 5.1 [18]) is

$$g_{24C} = x^{24} + x^{23} + x^{21} + x^{20} + x^{17} + x^{15} + x^{13} + x^{12} + x^{8} + x^{4} + x^{2} + x + 1. \tag{2.1}$$

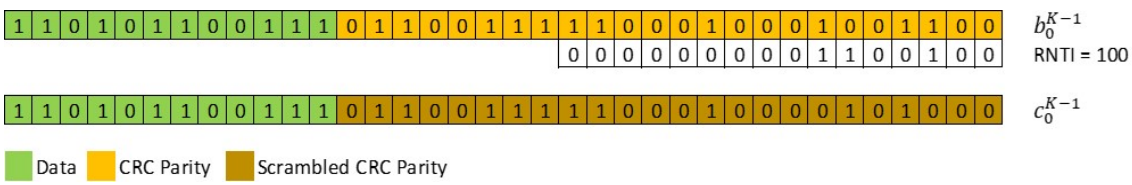
To determine the p_0^{L-1} CRC parity bits, where $L = 24$, g_{24C} is divided by a_0^{A-1} in Galois Field (GF)(2), also known as $\mathbb{Z}/2$. The p_0^{L-1} CRC parity bits (highlighted in orange below)

are then appended onto the end of the a_0^{A-1} data bits, and the combined bit strings are denoted as b_0^{K-1} where $K = A + L$. b_0^{K-1} for our example is shown below.

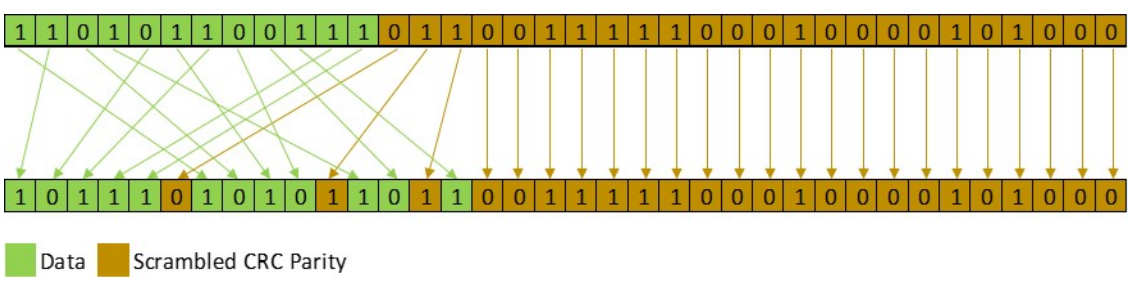


2.1.2 CRC Scrambling and Interleaving

The p_0^{L-1} CRC parity bits of b_0^{K-1} are next scrambled with the RNTI that corresponds to the UE that the DCI message is intended for. The scrambling is conducted such that the most significant bit of p_0^{L-1} is aligned with the least significant bit of the RNTI and an exclusive OR operation is applied. Scrambling using an arbitrary example RNTI of 100 results in p_{17}, p_{18}, and p_{21} switching parity. The scrambled p_0^{L-1} CRC parity bits are now denoted $p_0'^{L-1}$ (highlighted in brown), and the overall sequence is now denoted as c_0^{K-1}.



The data bits and the scrambled CRC parity bits are next interleaved. This step places the scrambled CRC parity bits behind the data bits upon which they correspond, facilitating early abortion of the decoding process when a bit error is detected. The mechanics behind how the polar decoding process can be aborted early using the scrambled CRC parity bits is explained at the end of Section 2.2.5. The interleaving pattern is structured so that it works for any valid-length DCI message (see Section 5.3.1.1 [18]). In our example, the relationship between c_0^{K-1} (upper row) and the interleaved sequence $c_0'^{K-1}$ (lower row) is shown below.



In order to better understand the effect that interleaving has on ${c_0'}^{K-1}$, the generator polynomial g_{24C} is now used to create a generator matrix P. This matrix P will establish direct relationships between which data bits correspond to each of the CRC bits. In order to create P, $A-1$ zeros are first appended to g_{24C} as shown below.

$$\left|\overbrace{1\ 1\ 0\ 1\ 1\ 0\ 0\ 1\ 0\ 1\ 0\ 1\ 1\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 1\ 0\ 1\ 1\ 1}^{g_{24C}}\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\right|$$

Next new rows are generated from the appended row by circular shifting the bits to the right one index until A rows have been generated.

$$\left|\begin{array}{cccccccccccccccccccccccccccccccccccccc}
1&1&0&1&1&0&0&1&0&1&0&1&1&0&0&0&1&0&0&0&1&0&1&1&1&0&0&0&0&0&0&0&0&0&0&0\\
0&1&1&0&1&1&0&0&1&0&1&0&1&1&0&0&0&1&0&0&0&1&0&1&1&1&0&0&0&0&0&0&0&0&0&0\\
0&0&1&1&0&1&1&0&0&1&0&1&0&1&1&0&0&0&1&0&0&0&1&0&1&1&1&0&0&0&0&0&0&0&0&0\\
0&0&0&1&1&0&1&1&0&0&1&0&1&0&1&1&0&0&0&1&0&0&0&1&0&1&1&1&0&0&0&0&0&0&0&0\\
0&0&0&0&1&1&0&1&1&0&0&1&0&1&0&1&1&0&0&0&1&0&0&0&1&0&1&1&1&0&0&0&0&0&0&0\\
0&0&0&0&0&1&1&0&1&1&0&0&1&0&1&0&1&1&0&0&0&1&0&0&0&1&0&1&1&1&0&0&0&0&0&0\\
0&0&0&0&0&0&1&1&0&1&1&0&0&1&0&1&0&1&1&0&0&0&1&0&0&0&1&0&1&1&1&0&0&0&0&0\\
0&0&0&0&0&0&0&1&1&0&1&1&0&0&1&0&1&0&1&1&0&0&0&1&0&0&0&1&0&1&1&1&0&0&0&0\\
0&0&0&0&0&0&0&0&1&1&0&1&1&0&0&1&0&1&0&1&1&0&0&0&1&0&0&0&1&0&1&1&1&0&0&0\\
0&0&0&0&0&0&0&0&0&1&1&0&1&1&0&0&1&0&1&0&1&1&0&0&0&1&0&0&0&1&0&1&1&1&0&0\\
0&0&0&0&0&0&0&0&0&0&1&1&0&1&1&0&0&1&0&1&0&1&1&0&0&0&1&0&0&0&1&0&1&1&1&0\\
0&0&0&0&0&0&0&0&0&0&0&1&1&0&1&1&0&0&1&0&1&0&1&1&0&0&0&1&0&0&0&1&0&1&1&1
\end{array}\right|$$

Finally the reduced row echelon form of the matrix is determined in $\mathbb{Z}/2$. This operation produces the matrix $[I|P]$, where I is the identity matrix and P is the generator matrix.

$$\left|\overbrace{\begin{array}{cccccccccccc}
1&0&0&0&0&0&0&0&0&0&0&0\\
0&1&0&0&0&0&0&0&0&0&0&0\\
0&0&1&0&0&0&0&0&0&0&0&0\\
0&0&0&1&0&0&0&0&0&0&0&0\\
0&0&0&0&1&0&0&0&0&0&0&0\\
0&0&0&0&0&1&0&0&0&0&0&0\\
0&0&0&0&0&0&1&0&0&0&0&0\\
0&0&0&0&0&0&0&1&0&0&0&0\\
0&0&0&0&0&0&0&0&1&0&0&0\\
0&0&0&0&0&0&0&0&0&1&0&0\\
0&0&0&0&0&0&0&0&0&0&1&0\\
0&0&0&0&0&0&0&0&0&0&0&1
\end{array}}^{I}\overbrace{\begin{array}{cccccccccccc}
0&1&0&1&1&1&1&1&1&1&0&1\\
1&1&1&1&0&1&1&0&1&0&1&1\\
0&1&1&1&1&0&1&1&0&1&0&1\\
0&0&1&1&1&1&0&1&1&0&1&0\\
0&0&0&1&1&1&1&0&1&1&0&1\\
1&1&0&0&0&1&1&1&1&0&0&0\\
0&1&1&1&0&1&0&0&0&1&0&1\\
1&1&1&0&1&0&0&0&1&0&1&1\\
0&1&1&1&0&1&0&0&0&1&0&1\\
0&1&1&1&0&1&0&0&0&1&0&1\\
1&1&0&1&1&0&0&1&0&1&0&1\\
1&1&0&1&1&0&0&1&0&1&0&1
\end{array}}^{P}\right|$$

Each of the columns of P represent a CRC parity bit, and each of the rows of P represent the

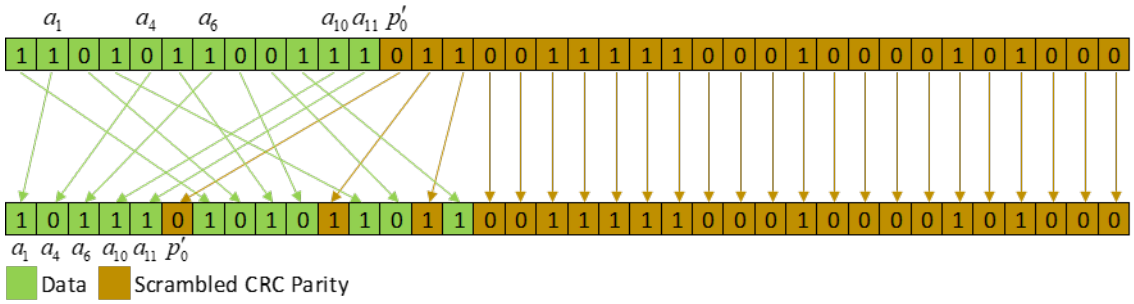
data bits that correspond to the respective CRC parity bit. Figure 2.2 presents the information contained within P in a manner that explicitly depicts the relationships between data and CRC parity bits.

p_0	p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8	p_9	p_{10}	p_{11}	p_{12}	p_{13}	p_{14}	p_{15}	p_{16}	p_{17}	p_{18}	p_{19}	p_{20}	p_{21}	p_{22}	p_{23}
-	a_0	-	a_0	a_0	a_0	a_0	a_0	a_0	a_0	-	a_0	-	-	-	a_0	-	a_0	-	-	a_0	a_0	a_0	a_0
a_1	a_1	a_1	a_1	-	a_1	a_1	-	a_1	-	a_1	a_1	-	-	-	-	-	-	a_1	-	a_1	a_1	-	-
-	a_2	a_2	a_2	a_2	-	a_2	a_2	-	a_2	-	a_2	a_2	-	-	-	-	-	-	a_2	-	a_2	a_2	-
-	-	a_3	a_3	a_3	a_3	-	a_3	a_3	-	a_3	-	a_3	a_3	-	-	-	-	-	-	a_3	-	a_3	a_3
a_4	a_4	-	-	-	a_4	a_4	a_4	a_4	-	-	-	a_4	a_4	a_4	-	a_4	-	-	-	a_4	a_4	a_4	-
-	a_5	a_5	-	-	-	a_5	a_5	a_5	a_5	-	-	-	a_5	a_5	a_5	-	a_5	-	-	-	a_5	a_5	a_5
a_6	a_6	a_6	-	a_6	-	-	-	a_6	-	a_6	a_6	a_6	-	a_6	a_6	-	-	a_6	-	a_6	-	-	-
-	a_7	a_7	a_7	-	a_7	-	-	-	a_7	-	a_7	a_7	a_7	-	a_7	a_7	-	-	a_7	-	a_7	-	-
-	-	a_8	a_8	a_8	-	a_8	-	-	-	a_8	-	a_8	a_8	a_8	-	a_8	a_8	-	-	a_8	-	a_8	-
-	-	-	a_9	a_9	a_9	-	a_9	-	-	-	a_9	-	a_9	a_9	a_9	-	a_9	a_9	-	-	a_9	-	a_9
a_{10}	a_{10}	-	a_{10}	-	a_{10}	a_{10}	a_{10}	a_{10}	a_{10}	-	a_{10}	-	-	a_{10}	a_{10}	-	-	a_{10}	a_{10}	a_{10}	-	-	a_{10}
a_{11}	-	a_{11}	a_{11}	-	-	a_{11}	-	a_{11}	-	a_{11}	a_{11}	-	-	-	a_{11}	-	-	-	a_{11}	-	a_{11}	a_{11}	a_{11}

Figure 2.2. Relationship between Data and CRC Parity Bits.

The first column of Figure 2.2 is interpreted such that the parity of p_0 equals the sum of the parities of a_1, a_4, a_6, a_{10}, and a_{11}. Due to p_0^{L-1} being scrambled by the RNTI, however, in practice the parities of the respective data bits are compared against that of the scrambled bit p'_i, not p_i. Therefore, during the decoding process detailed in Section 2.2.5, Figure 2.2 is implemented in a manner such that if $p_i = p'_i$ then the parity of p'_i should equal that of its respective data bits. Conversely, if scrambling with the RNTI causes $p_i \neq p'_i$, then the parity of p'_i should not equal that of its respective data bits.

Re-examining the interleaved sequence ${c'}_0^{K-1}$ (bottom row) from earlier, it is evident that interleaving placed p'_0 directly behind its respective data bits, and since $p_0 = p'_0$, the parity of p'_0 equals that of its respective data bits.



2.1.3 Frozen Bit Determination and Insertion

A brief background on the theory of polar coding is presented at this point in order to assist with the understanding of the concept of frozen bits. Polar codes are a class of capacity achieving codes first presented in 2009 [20]. Polar codes are built upon the polarization effect of the matrix $G_2 = \left(\begin{smallmatrix} 1 & 0 \\ 1 & 1 \end{smallmatrix}\right)$. The polarization matrix is recursively expanded to $G_N = G_2^{\otimes n}$ where N is the code length and is defined as $N = 2^n$, and $\otimes$ is the Kronecker product. To define the Kronecker product, let $\mathbf{Y}$ be an $m \times n$ matrix and $\mathbf{Z}$ be a $p \times q$ matrix. The Kronecker product $\mathbf{Y} \otimes \mathbf{Z}$ is then the $pm \times qn$ block matrix

$$\mathbf{Y} \otimes \mathbf{Z} = \begin{bmatrix} y_{11}\mathbf{Z} & \dots & y_{1n}\mathbf{Z} \\ \vdots & \ddots & \vdots \\ y_{m1}\mathbf{Z} & \dots & y_{mn}\mathbf{Z} \end{bmatrix}. \tag{2.2}$$

The effect of the polarization matrix is to establish N virtual sub-channels, where each sub-channel has a varying degree of reliability and is used to transmit a single bit u_i. As N trends towards infinity, the sub-channels trend to becoming either completely noiseless or completely noisy [20]. Put a different way, in a given transmission of infinite length that the polarization matrix has been applied to, certain bit positions within the transmission (i.e., sub-channels) become completely unaffected by noise, and the other bit positions (i.e., sub-channels) become completely affected by noise. For coding purposes, it is desirable to place message bits in the sub-channels that are noiseless, and place frozen bits (set to zero) in the noisy sub-channels. For codes of defined length, the sub-channels have varying degrees of reliability. For the PDCCH the sub-channels are arranged in reliability order in a standardized fashion based off of allowable code lengths (see Section 5.3.1.2 [18]).

The first step in determining the quantity and location of frozen bits to be inserted into ${c'_0}^{K-1}$ is to determine n, which by extension will allow us to determine N. The value n is dependent upon the ratio of the number of message bits to be encoded K to the rate-matching codeword length E, as well as the length of E itself. (see Section 5.3.1 [18]). For the PDCCH, allowable values of n are 5, 6, 7, 8, and 9, corresponding to N values of 32, 64, 128, 256, and 512, respectively. For our example where $E = 128$, it is determined that $n = 7$ and $N = 128$. Using the standardized sub-channel reliability positions, $N - K$ frozen bits (highlighted in blue) are inserted into ${c'_0}^{K-1}$ message bits to form u_0^{N-1}.

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	0	0	0	0	0	0	0	1	0	1	0	0
1	1	1	1	0	1	0	0	0	1	0	0	0	0	1	0	1	0	0	0																

Reliability Freezing Data Scrambled CRC Parity

A disadvantage of implementing polar codes in their native form is that the code length N must be a factor of two. In order to allow implementations of other code lengths different rate-matching schemes can be utilized. These include repetition, puncturing, and shortening. Considerations of implementing puncturing or shortening are discussed in further detail in Section 2.1.6, but is mentioned here as it does have an effect on the selection of frozen bits.

2.1.4 Polar Encoding

After the frozen bits have been inserted to form u_0^{N-1}, the codeword d_0^{N-1} is formed by

$$d_0^{N-1} = u_0^{N-1} G_N \tag{2.3}$$

where the multiplication is performed in GF(2). For our example, d_0^{N-1} is shown below.

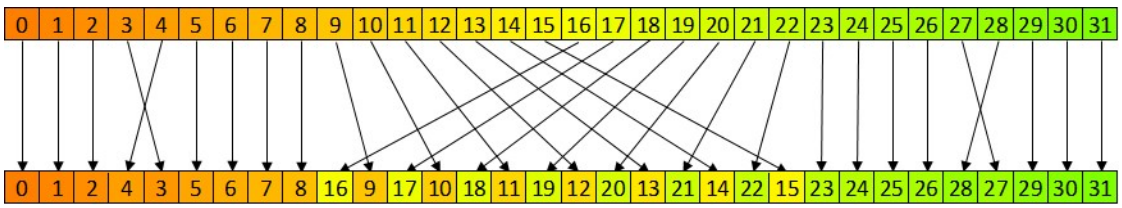
0	0	0	0	1	0	1	0	1	1	1	1	1	0	0	1	0	0	1	0	0	1	0	0	0	0	1	0	1	0	0	0	0	1	1	0
0	0	1	1	1	0	0	1	0	0	0	0	1	0	1	1	0	0	1	0	1	0	1	1	1	1	1	0	0	1	1	0	0	0	1	1
1	0	0	1	0	0	0	0	1	0	1	1	0	0	1	0	1	0	1	1	1	1	1	0	0	0	0	0	1	0	1	0	1	1	1	1
1	0	0	1	0	0	1	0	0	1	0	0	0	0	1	0	1	0	0	0																

Of note, due to multiplying our message u_0^{N-1} by the polarization matrix G_N, we are no longer able to directly discern what the individual bits in d_0^{N-1} represent (e.g., data, CRC

parity, frozen) like we were up to this point.

2.1.5 Sub-block Interleaving

After polar encoding, the next step in the process is that of sub-block interleaving. This step divides the codeword d_0^{N-1} into 32 sub-blocks that are interleaved in order to reduce the likelihood that there is a long continual sequence of ones or zeros in the codeword prior to modulation. The sub-block interleaving pattern is standardized as shown below,



where the top row is the 32 d_i sub-blocks prior to interleaving, and the bottom row is the 32 y_i sub-blocks after interleaving (see Section 5.4.1.1 [18]). For our example, the 128-bit codeword d_0^{N-1} is interleaved into 32 sub-blocks of four bits each per the interleaving pattern to form y_0^{N-1}.

0	0	0	0	1	0	1	0	1	1	1	1	0	0	1	0	1	0	0	1	0	1	0	0	0	0	1	0	1	0	0	0	0	1	1	0
0	1	1	0	0	0	1	1	0	0	1	1	1	0	0	1	1	0	0	1	0	0	0	0	0	0	0	0	1	0	1	1	1	0	1	1
0	0	1	0	0	0	1	0	1	0	1	1	1	0	1	1	1	1	1	0	1	1	1	0	0	0	0	0	1	0	1	0	1	1	1	1
0	0	1	0	1	0	0	1	0	1	0	0	0	0	1	0	1	0	0	0																

2.1.6 Rate-Matching

In our example up to this point $E = N$, and as such there is no need for a rate-matching scheme to be implemented. In these specific instances where there is no rate-matching scheme implemented, the interleaved codeword y_0^{N-1} is equal to the rate-adjusted codeword e_0^{E-1}. For those cases where $E \neq N$, a suitable rate-matching technique is implemented in order for the length of the codeword to be adjusted prior to modulation. In order to maintain low bit error rates, the rate-matching scheme to be implemented directly influences the quantity and location of frozen bits that are inserted into the message $c_0'^{K-1}$. The three rate-matching schemes that can be implemented are repetition (codeword is repeated onto itself in a circular manner), puncturing (bits removed from beginning of codeword), and shortening (bits removed from end of codeword). Figure 2.3 provides a graphic representation of the

different rate-matching schemes.

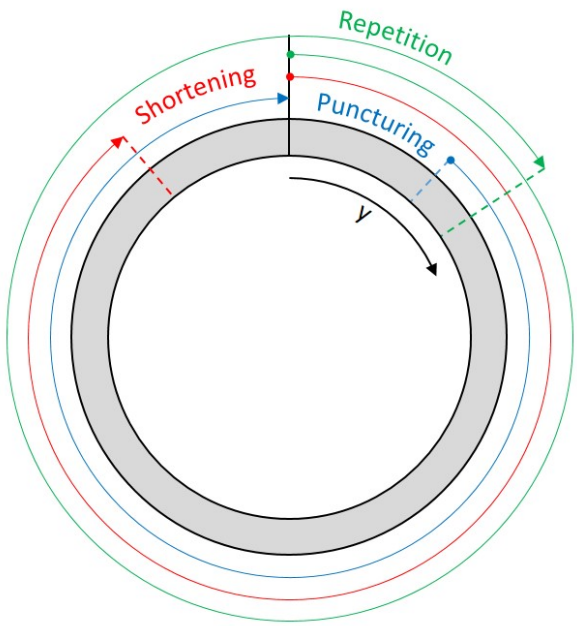


Figure 2.3. Rate-Matching Schemes

Figure 2.4 illustrates the relationship between the data payload length A, the rate-matching codeword length E, and the different rate-matching schemes. Also called out on Figure 2.4 are the three values of E that will be used throughout chapters 2 and 3 to demonstrate examples of repetition, puncturing, and shortening rate-matching schemes.

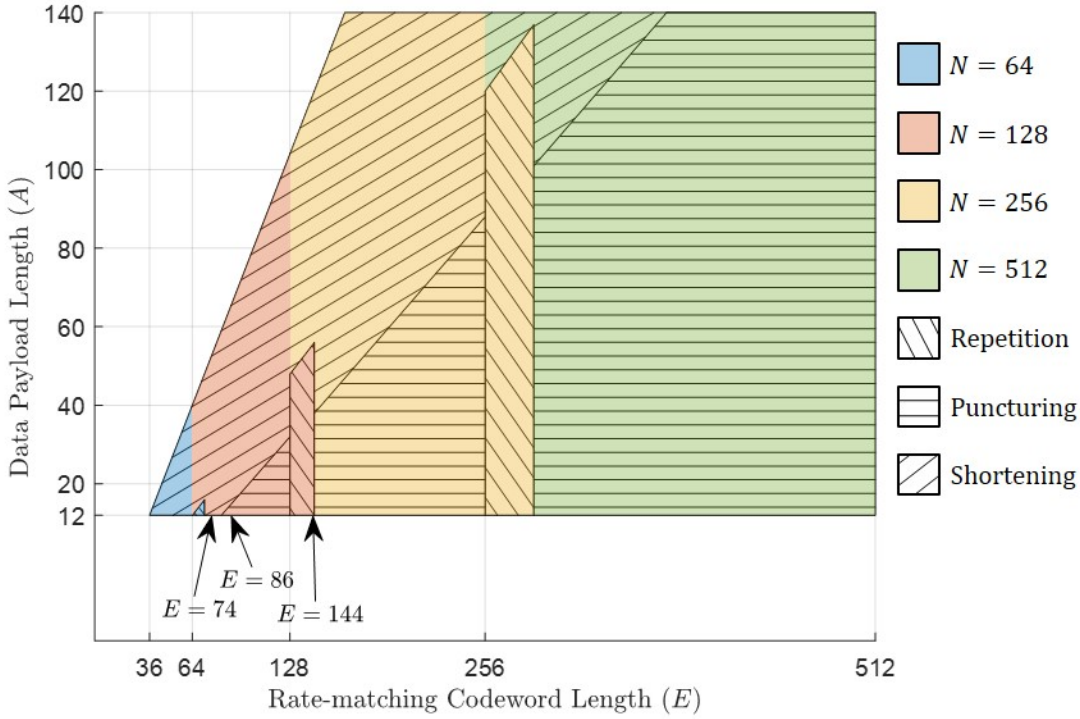


Figure 2.4. Rate-Matching Implementation Relationships.

Repetition

Repetition occurs when the rate-matching codeword length (E) is greater than the codeword length that was encoded (N). This can occur when there are relatively few message bits compared to the quantity of physical resources available. For repetition, codeword bits y_0^{E-N-1} are appended onto the end of y_0^{N-1} in a circular manner such that the new codeword e_0^{E-1} is formed. Use of repetition does not have any effect on the quantity or location of frozen bits that are inserted into ${c'}_0^{K-1}$. As an example, if the value of E from Table 2.1 is changed from 128 to 144 in order to implement repetition, e_0^{E-1} is formed as shown below,

0	0	0	0	1	0	1	0	1	1	1	1	0	0	1	0	1	0	0	1	0	1	0	0	0	0	1	0	1	0	0	0	0	1	1	0
0	1	1	0	0	0	1	1	0	0	1	1	1	0	0	1	1	0	0	1	0	0	0	0	0	0	0	0	1	0	1	1	1	0	1	1
0	0	1	0	0	0	1	0	1	0	1	1	1	0	1	1	1	1	1	0	1	1	1	0	0	0	0	0	1	0	1	0	1	1	1	1
0	0	1	0	1	0	0	1	0	1	0	0	0	0	1	0	1	0	0	0	0	0	0	0	1	0	1	0	1	1	1	1	0	0	1	0

Repeated Bits

where the first $E - N$ bits from y_0^{N-1} have been copied and appended to the end of y_0^{N-1}.

Puncturing

Puncturing occurs when $E < N$ and $K/E \leq 7/16$. Referring back to Figure 2.3, puncturing removes the first $N - E$ bits from y_0^{N-1} in order to form e_0^{E-1}. So that data loss does not occur when puncturing is implemented, several additional steps are required during the determination of the frozen bit set to be inserted into ${c'}_0^{K-1}$. First, frozen bits are inserted into indices that correspond with those bits to be punctured. The effect of sub-block interleaving must be considered, so the bit indices that will be punctured are not the first $N - E$ from d_0^{N-1} but rather from y_0^{N-1}. Next, dependent upon the ratio of E and N, additional frozen bits may need to be inserted after the indices to be punctured. This extra freezing step (see Section 5.4.1.1 [18]) is necessary to enable successful decoding of the transmitted codeword. After the frozen bits corresponding to the punctured bits and extra frozen bits have been determined, frozen bits are further inserted into ${c'}_0^{K-1}$ per the standardized sub-channel reliability positioning to form u_0^{N-1}. As an example, if the value of E from Table 2.1 is changed from 128 to 86 in order to implement puncturing, u_0^{N-1} is

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	1	1	0	0	0	0	0	0	0	1	0	1	0	0
1	1	1	1	0	1	0	0	0	1	0	0	0	0	1	0	1	0	0	0																

Punctured Freezing Reliability Freezing Extra Freezing Data Scrambled CRC Parity

where light blue highlighting denotes frozen bits to be punctured, blue highlighting represents reliability freezing, and dark blue highlighting denotes extra freezing. As mentioned earlier, the bits to be punctured (light blue in color) are determined based off of the post sub-interleaving order, and as such they are not the first $N - E$ bits of u_0^{N-1}.

Now that u_0^{N-1} has been formed, the polar encoding and sub-block interleaving steps are identical to those outlined in sections 2.1.4 and 2.1.5. The interleaved codeword y_0^{N-1} that is formed is shown below.

0	1	0	0	1	1	1	0	1	0	1	1	0	1	1	0	1	1	0	1	0	0	0	0	0	1	1	0	1	1	0	0	0	1	1	0
0	0	1	0	1	1	0	0	1	0	0	0	0	1	1	0	0	0	1	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0
0	0	1	0	0	1	1	0	1	0	1	1	1	1	1	1	0	0	0	1	0	1	0	1	0	0	0	0	1	0	1	0	1	1	1	1
0	0	1	0	1	0	0	1	0	1	0	0	0	0	1	0	1	0	0	0																

Finally, the first $N - E$ bits are removed from y_0^{N-1} to form e_0^{E-1}.

						0	0	1	0	0	0	0	1	1	0	0	0	1	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0
0	0	1	0	0	1	1	0	1	0	1	1	1	1	1	1	0	0	0	1	0	1	0	1	0	0	0	0	1	0	1	0	1	1	1	1
0	0	1	0	1	0	0	1	0	1	0	0	0	0	1	0	1	0	0	0																

Of note, there were both zero and one values present for the bits that are punctured. This is due to the properties of the polarization matrix G_N, and will influence the manner that the rate-recovery step is performed in Section 2.2.3.

Shortening

Shortening occurs when $E < N$ and $K/E > 7/16$. Referring back to Figure 2.3, shortening removes the last $N - E$ bits from y_0^{N-1} in order to form e_0^{E-1}. The process of shortening is simpler then puncturing, in that there is no extra freezing step. Frozen bits are inserted into the last $N - E$ indices that are to be punctured, then frozen bits are further inserted into ${c'}_0^{K-1}$ per the standardized sub-channel reliability positioning to form u_0^{N-1}. As an example, if the value of E from Table 2.1 is changed from 128 to 74 in order to implement shortening, u_0^{N-1} is

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	1	1	0	1	0	0	0	0	0
0	0	1	0	0	1	1	0	1	1	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																

Shortened Freezing | Reliability Freezing | Data | Scrambled CRC Parity

where light blue highlighting denotes frozen bits to be shortened and blue highlighting denotes reliability freezing. y_0^{N-1}, similar to puncturing, is formed as outlined in sections 2.1.4 and 2.1.5 and is shown below.

0	1	0	0	0	1	0	0	0	0	1	1	1	1	1	1	1	0	1	0	1	0	0	1	1	1	0	1	0	0	1	0	0	1	0	1
1	1	0	0	1	0	1	0	1	1	0	0	0	0	1	0	0	0	1	1	0	1	0	0	0	0	1	1	1	1	0	1	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																

Finally, the last $N - E$ bits from y_0^{N-1} are removed to form e_0^{E-1}.

0	1	0	0	0	1	0	0	0	0	1	1	1	1	1	1	1	0	1	0	1	0	0	1	1	1	0	1	0	0	1	0	0	1	0	1
1	1	0	0	1	0	1	0	1	1	0	0	0	0	1	0	0	0	1	1	0	1	0	0	0	0	1	1	1	1	0	1	0	0	0	0
0	1																																		

In contrast to puncturing, the bits that are removed during the shortening process are in this example, and will always be for any implementation of shortening, all zero valued. Where the punctured frozen indices positioned at the beginning of u_0^{N-1} became zero and one valued after polar encoding, the shortened frozen indices, positioned at the end of u_0^{N-1} remain zero valued due to the properties of the polarization matrix. This property will influence the manner that rate-recovery is performed in Section 2.2.3.

2.1.7 Scrambling

After the message y_0^{N-1} has had its length adjusted per the rate-matching scheme in use to generate e_0^{E-1}, the next step in the encoding process is that of scrambling. The purpose of scrambling is both to facilitate message addressing and provided data anonymization by making the message intelligible only to the UE to which it is intended. This is achieved through the use of a length-31 Gold sequence whose generator is initialized with the RNTI and η_{ID} for a given UE. The initialization value (see Section 7.3.2.3 [17]) is determined by

$$c_{init} = (\text{RNTI} \times 2^{16} + \eta_{\text{ID}}) \mod 2^{31}. \tag{2.4}$$

Of particular interest is the mod 2^{31} operator that has the effect of discarding the 16th (i.e., most significant) bit from the RNTI value. Therefore, the size of the set of possible unique scrambling sequences is halved, as RNTIs r and $r + 2^{15}$ will yield identical initialization values, and thus identical scrambling sequences. In Chapter 3, we will see that this is beneficial in that it reduces the number of potential syndromes that need to be examined, but also introduces additional considerations in how to determine whether a given message has been scrambled with RNTI r or $r + 2^{15}$. For our example parameters defined in 2.1, the scrambling sequence initialization value c_{init} is shown below.

$$\overbrace{0000000011001000}^{\text{RNTI}}\overbrace{0000000000001010}^{\eta_{\text{ID}}}$$

The remaining steps for generating the scrambling sequence are omitted for brevity (see Section 5.2.1 [17]). For our example, the scrambling sequence s_0^{E-1} is shown below.

0	0	0	1	0	1	1	0	0	1	1	0	1	0	0	0	1	0	0	1	0	0	0	1	1	1	1	1	1	1	0	1	1	0	1	1
1	0	0	0	0	0	0	0	0	1	1	0	0	0	1	0	1	1	1	0	0	0	0	0	0	0	1	0	0	1	1	0	1	0	1	0
1	0	1	1	1	0	1	1	0	1	1	0	0	0	1	1	1	1	0	1	0	0	0	0	0	1	0	1	1	0	1	1	1	1	0	0
1	1	1	1	0	0	1	1	1	0	1	0	0	0	1	1	1	0	1	0																				

After the scrambling sequence s_0^{E-1} is generated, the message x_0^{E-1} to be modulated is determined by $x_0^{E-1} = e_0^{E-1} \oplus s_0^{E-1}$. For our example, the message to be modulated x_0^{E-1} is shown below.

0	0	0	1	1	1	0	0	1	0	0	1	1	0	1	0	0	0	0	0	0	0	1	0	1	1	1	0	1	0	1	0	1	1	1	0	1
1	1	1	0	0	0	1	1	0	1	0	1	1	0	1	1	0	1	1	1	0	0	0	0	0	0	1	0	1	1	0	1	0	0	0	1
1	0	0	1	1	0	0	1	1	1	0	1	1	0	0	0	0	0	1	1	1	1	1	0	0	1	0	1	0	0	0	1	0	0	1	1
1	1	0	1	1	0	1	0	1	1	1	0	0	0	0	1	0	0	1	0																				

2.1.8 Modulation

The PDCCH uses QPSK modulation. The complex-valued modulation symbols e_0^{M-1} are given by

$$e(i) = \frac{1}{\sqrt{2}}[(1 - 2x(2i)) + j(1 - 2x(2i + 1))] \tag{2.5}$$

where $i = 0, 1, \ldots, E/2$ (see Section 5.1.3 [17]). Figure 2.5 illustrates a constellation plot that maps the different possibilities of bit pairs to each of the complex-valued symbols.

After the message bits have been modulated, the encoding process is complete and the complex-valued symbols are mapped to physical resources for transmission.

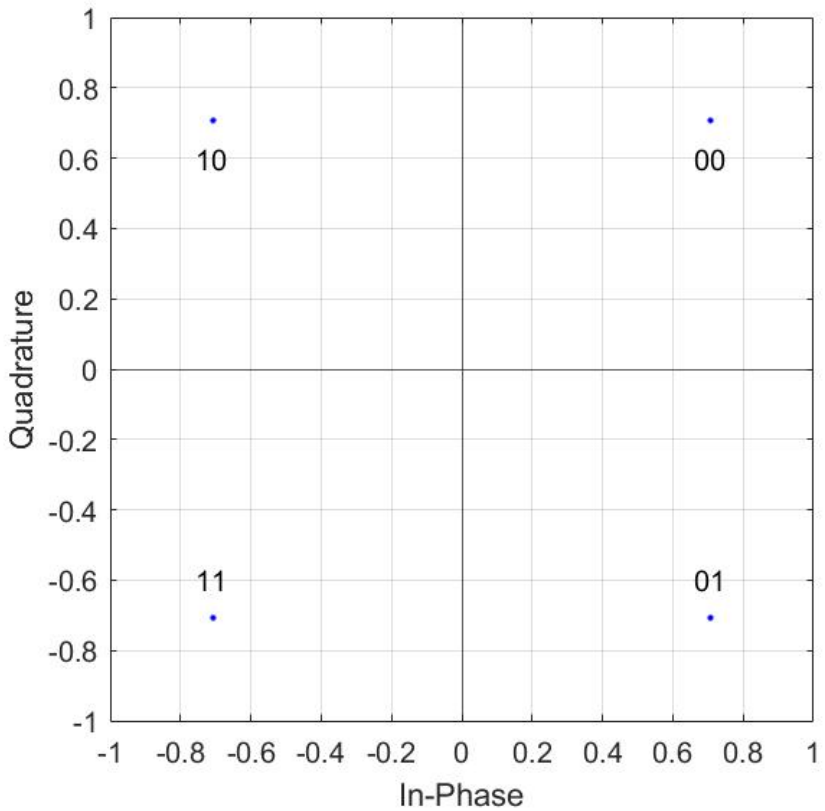


Figure 2.5. Bit Pair to Complex Symbol Mapping.

2.2 PDCCH Decoding Process

The PDCCH decoding process recovers the transmitted codeword through providing error detection and correction, so that the DCI message may be processed by the intended UE. The process is referred to as a "blind" decoding process, in that each UE monitoring a given cell does not have *a priori* knowledge if a received message is intended for it or not. Additionally, each UE can be configured to receive up to three different DCI message lengths (see Section 7.3.1.0 [18]) in a given cell. Each of the different message lengths influence the rate-matching scheme in use, and subsequently affect the manner in which the message is decoded. This means that in practice, each UE attempts to decode each received message up to three times, processing the message payload if no errors are detected, and discarding the message when errors are present. For simplicity purposes in the following

example, illustrated in Figure 2.6, it is assumed that the message length is known.

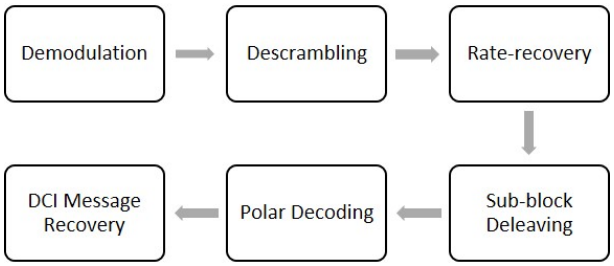


Figure 2.6. PDCCH Decoding Process.

2.2.1 Demodulation

The first step in the decoding process is that of demodulating the received QPSK complex-valued symbols. Let the received symbols be f_0^{M-1} where $M = E/2$. Using the same constellations as during the modulation step (see Section 5.1.3 [17]), a soft output of logarithmic likelihood ratio (LLR) values can be determined for each bit in the codeword. The complex-valued symbols demodulate to the LLR values illustrated in Figure 2.7.

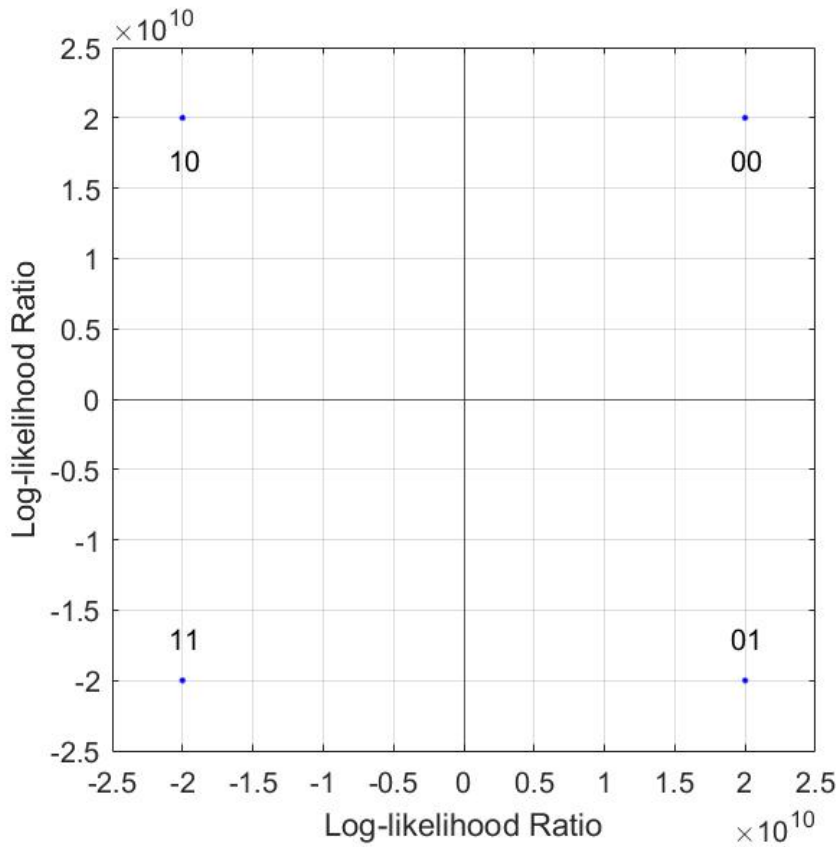


Figure 2.7. Demodulation LLR Soft Outputs.

The bit pairs that map to each corresponding pair of LLR values are also annotated for reference on Figure 2.7, though caution must be exercised in interpreting the bit pairs. In a transmission channel void of additive white Gaussian noise (AWGN), a bit value of zero maps to a positive LLR value and a bit value of one maps to a negative LLR value. In order to apply polar decoding to the received complex symbols, however, the demodulated values must remain as LLR values and not be mapped back to bit values until the appropriate step in the decoding process. Prematurely mapping LLR values to bit values will remove the error-correcting properties of polar encoding and decoding.

In our example, the received symbols f_0^{M-1} demodulate to form h_0^{E-1} where each value h_i is listed without the exponent ($h_i \times 10^{10}$) for brevity.

2	2	2	-2	-2	-2	2	2	-2	2	2	-2	-2	2	-2	2	2	2	2	2	2	-2	2	-2	-2	-2	2	-2	2	-2	2	-2	-2	-2	2	-2
-2	-2	-2	2	2	2	-2	-2	2	-2	2	-2	-2	2	-2	-2	2	-2	-2	-2	2	2	2	2	2	2	-2	2	-2	-2	2	-2	2	2	2	-2
-2	2	2	-2	-2	2	2	-2	-2	-2	2	-2	-2	2	2	2	2	2	-2	-2	-2	-2	-2	2	2	-2	2	-2	2	2	2	-2	2	2	-2	-2
-2	-2	2	-2	-2	2	-2	2	-2	-2	-2	2	2	2	2	-2	2	2	-2	2																

2.2.2 Descrambling

After demodulation, the effects of scrambling are removed from the received LLR values. This is achieved through generating a scrambling sequence identical to the steps outlined in Section 2.1.7 with one exception. Instead of the scrambling sequence being mapped in binary fashion, the scrambling sequence is mapped as positive and negative values. This signed mapping outputs a positive one for bit values of zero and a negative one for bit values of one. Applying the signed scrambling sequence to h_0^{E-1} forms the LLR values j_0^{E-1} shown below.

2	2	2	2	-2	2	-2	2	-2	-2	-2	-2	2	2	-2	2	-2	2	2	-2	2	-2	2	2	2	2	-2	2	-2	2	2	2	2	-2	-2	2
2	-2	-2	2	2	2	-2	-2	2	2	-2	-2	-2	2	2	-2	-2	2	2	-2	2	2	2	2	2	2	2	2	-2	2	-2	-2	-2	2	-2	-2
2	2	-2	2	2	2	-2	2	-2	2	-2	-2	-2	2	-2	-2	-2	-2	-2	2	-2	-2	-2	2	2	2	2	2	-2	2	-2	2	-2	-2	-2	-2
2	2	-2	2	-2	2	2	-2	2	-2	2	2	2	2	-2	2	-2	2	2	2																

2.2.3 Rate-recovery

Prior to polar decoding, the received LLR values must be of a length of a power of 2. For our example, where $E = N$, there are no rate-recovery considerations and the descrambled values j_0^{E-1} are equal to the rate-recovered values k_0^{N-1}. For the cases where $E \neq N$, the rate-matching scheme that was implemented during encoding process (e.g., repetition, puncturing, shortening) now must be undone.

During the rate-recovery step is where the receiver must work through a portion of the blind decoding process mentioned at the beginning of Section 2.2. During normal operations, an UE can be configured to monitor for data payloads of up to three different lengths. The receiver uses the length of the received LLR values E, combined with one of the data payload lengths A it is configured for, to determine both the rate-matching scheme implemented and the polar decoding length N in a reverse implementation of Section 2.1.6. If the receiver is unable to correctly decode the received codeword into an intelligible data payload, it repeats the process using a different data payload length it is configured for. Throughout this paper, it is assumed that the receiver uses the correct data payload length during rate-recovery and

decoding.

Repetition

Repetition occurs when $E > N$. Recalling from Section 2.1.6, the process of repetition takes bits y_0^{E-N-1} and appends them onto the end of y_0^{N-1}. The simplest approach to recovering repetition would be to simply remove the LLR values j_N^{E-1} from j_0^{E-1} to form k_0^{N-1}. A better approach is to leverage the properties of the LLR values, specifically the ability to sum values that represent the same transmitted bits. The circular nature of repetition provides that LLR values j_i and j_{i+N} correlate to the same transmitted bit. Therefore, summing each j_i and j_{i+N} value will yield an LLR value greater in magnitude, which effectively increases the processing gain of that bit and helps to mitigate any potential effects of additive noise during transmission. This can be illustrated by referring back to Figure 2.7. Given that positive LLR values map to zero bit values, and negative LLR values map to one bit values, the greater the magnitude (farther away from the horizontal or vertical axis) of the LLR value, the greater confidence we will have during the polar decoding step of what the bit value is. Using $E = 144$ as in the repetition example from Section 2.1.6, j_0^{E-1} is formed from the demodulated, descrambled LLR values, where the exponent (10^{10}) for each value j_i is omitted for brevity.

2	2	2	2	-2	2	-2	2	-2	-2	-2	-2	2	2	-2	2	-2	2	2	-2	2	-2	2	2	2	2	-2	2	-2	2	2	2	2	-2	-2	2
2	-2	-2	2	2	2	-2	-2	2	2	-2	-2	-2	2	2	-2	-2	2	2	-2	2	2	2	2	2	2	2	2	-2	2	-2	-2	-2	2	-2	-2
2	2	-2	2	2	2	-2	2	-2	2	-2	-2	-2	2	-2	-2	-2	-2	-2	2	-2	-2	-2	2	2	2	2	2	-2	2	-2	2	-2	-2	-2	-2
2	2	-2	2	-2	2	2	-2	2	-2	2	2	2	2	-2	2	-2	2	2	2	2	2	2	2	-2	2	-2	2	-2	-2	-2	-2	2	2	-2	2

Repeated Bits

In order to form the rate-recovered output k_0^{N-1}, the repeated indices are summed with the first $N - E$ indices of j_0^{E-1}.

4	4	4	4	-4	4	-4	4	-4	-4	-4	-4	4	4	-4	4	-2	2	2	-2	2	-2	2	2	2	2	-2	2	-2	2	2	2	2	-2	-2	2
2	-2	-2	2	2	2	-2	-2	2	2	-2	-2	-2	2	2	-2	-2	2	2	-2	2	2	2	2	2	2	2	2	-2	2	-2	-2	-2	2	-2	-2
2	2	-2	2	2	2	-2	2	-2	2	-2	-2	-2	2	-2	-2	-2	-2	-2	2	-2	-2	-2	2	2	2	2	2	-2	2	-2	2	-2	-2	-2	-2
2	2	-2	2	-2	2	2	-2	2	-2	2	2	2	2	-2	2	-2	2	2	2																

In order to form the rate-recovered output k_0^{N-1}, the repeated indices are summed with the first $N - E$ indices of j_0^{E-1}.

Puncturing

Puncturing occurs when $E < N$ and $K/E \leq 7/16$. Recalling from Section 2.1.6, the process of puncturing removed the first $N - E$ bits from y_0^{N-1} to form e_0^{E-1}. Furthermore, the $N - E$

bits that were punctured were both zero and one valued. Therefore, we do not have any certainty as to what value should be assigned to these punctured indices during the rate-recovery process. Due to this fact, LLR values of zero are assigned to each of the punctured bit indices k_0^{N-E-1} and the LLR values j_0^{E-1} are inserted into k_{N-E}^{N-1} to form k_0^{N-1}. Using $E = 86$ as in the puncturing example from Section 2.1.6, rate-recovery would output k_0^{N-1}, where the exponent (10^{10}) for each value k_i is omitted for brevity.

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	2	2	-2	2	2	2	2	-2	-2	2	2	2	-2	2	2	2	2	2	2	-2	2	2	2	-2	2	2	2	2	2	2
2	2	-2	2	2	-2	-2	2	-2	2	-2	-2	-2	-2	-2	-2	2	2	2	-2	2	-2	2	-2	2	2	2	2	-2	2	-2	2	-2	-2	-2	-2
2	2	-2	2	-2	2	2	-2	2	-2	2	2	2	2	-2	2	-2	2	2	2																

Shortening

Shortening occurs when $E < N$ and $K/E > 7/16$. Recalling from Section 2.1.6, the process of shortening removed the last $N - E$ bits from y_0^{N-1} in order to form e_0^{E-1}. Additionally, based on the positioning of the shortened frozen indices in u_0^{N-1}, the properties of the polarization matrix, and the structure of the sub-block interleaving pattern, the $N - E$ bits that were removed due to shortening were, and always will be, all zero valued. As we know with absolute certainty that the bit values for the shortened indices are zero valued, it follows that the LLR values for those indices are infinitely valued. In practice finite values are required for computations during the polar decoding step. Using LLR values of 9×10^{20} for the shortened indices generally provides a sufficient degree of confidence of the values of the indices without being computationally prohibitive. Using $E = 74$ as in the shortening example from Section 2.1.6, rate-recovery would output k_0^{N-1}, where the exponent (10^{10}) for indices k_0^{E-1} and the exponent (10^{20}) for indices k_E^{N-1} is omitted for brevity.

2	-2	2	2	2	-2	2	2	2	2	-2	-2	-2	-2	-2	-2	-2	2	-2	2	-2	2	2	-2	-2	-2	2	-2	2	2	-2	2	2	-2	2	-2
-2	-2	2	2	-2	2	-2	2	-2	-2	2	2	2	2	-2	2	2	2	-2	-2	2	-2	2	2	2	2	-2	-2	-2	-2	2	-2	2	2	2	2
2	-2	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9
9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9																

2.2.4 Sub-block Deleaving

Now that rate-recovery examples have been provided for the different rate-matching schemes, the remainder of Chapter 2 returns to the parameters specified in Table 2.1, specifically $E = 128$. Moving forward in the process steps, sub-block deleaving is per-

formed in the reverse manner as the steps outlined in Section 2.1.5. Sub-block deleaving outputs m_0^{N-1}, as shown below.

2	2	2	2	-2	2	-2	2	-2	-2	-2	-2	-2	2	2	-2	2	2	-2	2	2	-2	2	2	2	2	-2	2	-2	2	2	2	2	-2	-2	2
2	2	-2	-2	-2	2	2	-2	2	2	2	2	-2	2	-2	-2	2	2	-2	2	-2	2	-2	-2	-2	-2	-2	2	2	-2	-2	2	2	2	-2	-2
-2	2	2	-2	2	2	2	2	-2	2	-2	-2	2	2	-2	2	-2	2	-2	-2	-2	-2	-2	2	2	2	2	2	-2	2	-2	2	-2	-2	-2	-2
-2	2	2	-2	2	2	-2	2	2	-2	2	2	2	2	-2	2	-2	2	2	2																

2.2.5 Polar Decoding

The process of taking the soft LLR values and making hard bit determinations from them occurs during polar decoding. The underlying theory behind the process is presented in [20]. The decoding approach presented throughout this paper is referred to as LLR-based successive-cancellation (SC) decoding, in that the LLR values determined through demodulation are used and hard bit estimates are successively determined for the beginning of the message and sequentially working towards the end. Other decoding techniques, that generally have improved error-correcting performance at the cost of varying degrees of increased complexity, include successive-cancellation list (SCL) [21] and successive-cancellation stack (SCS) [22].

Up to this point in our example, we have let $E = N = 128$ (aside from the supplemental rate-matching examples). Having a code length this long is beneficial in understanding the relationship between the data and CRC parity bits, and the effects that the several interleaving steps have. In order to discuss the mechanics of polar decoding we will significantly reduce the code length to $N = 8$ bits, four of which are data and four are frozen bits. The data sequence used will be 1101, with frozen bits inserted into indices u_0, u_1, u_2, and u_4 (as per the standardized reliability positioning discussed in Section 2.1.3) to form $u_0^7 = 00010101$. The message u_0^7 is next encoded via multiplication by the polarization matrix G_8 in GF(2) to form $d_0^7 = 11000011$. This encoding step is illustrated in Figure 2.8, which will aid in the decoding steps to follow. The polar code graph depicted in Figure 2.8 represents the polarization matrix.

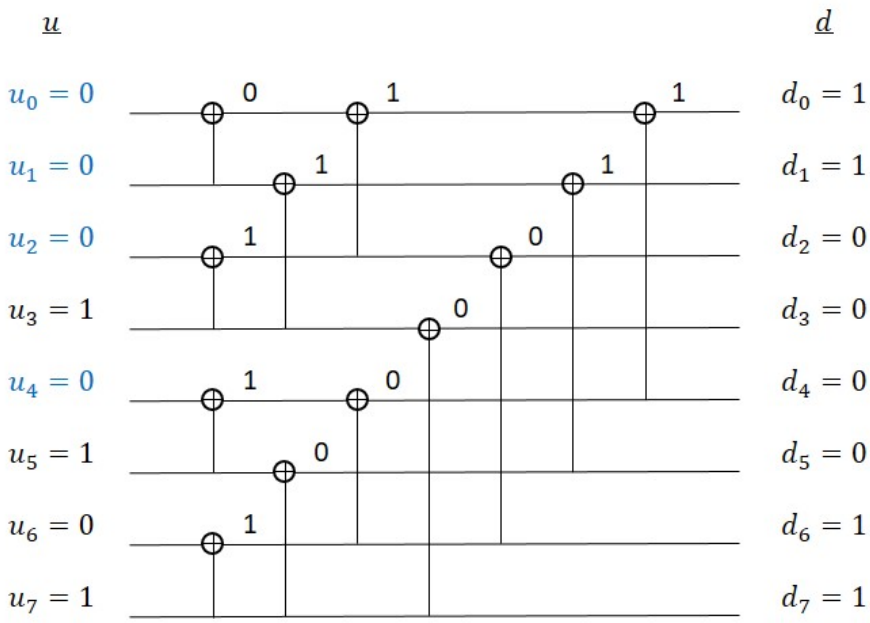


Figure 2.8. Polar Encoding Example.

QPSK modulation occurs next, and then transmission where AWGN is introduced to the signal. The received, demodulated LLR values are listed on the right-hand side of Figure 2.9 (exponent 10^{10} omitted for brevity). As the index locations of the frozen bits are standardized for given code lengths, we are able to insert zeros in the appropriate frozen bit locations on the left side of Figure 2.9 as well.

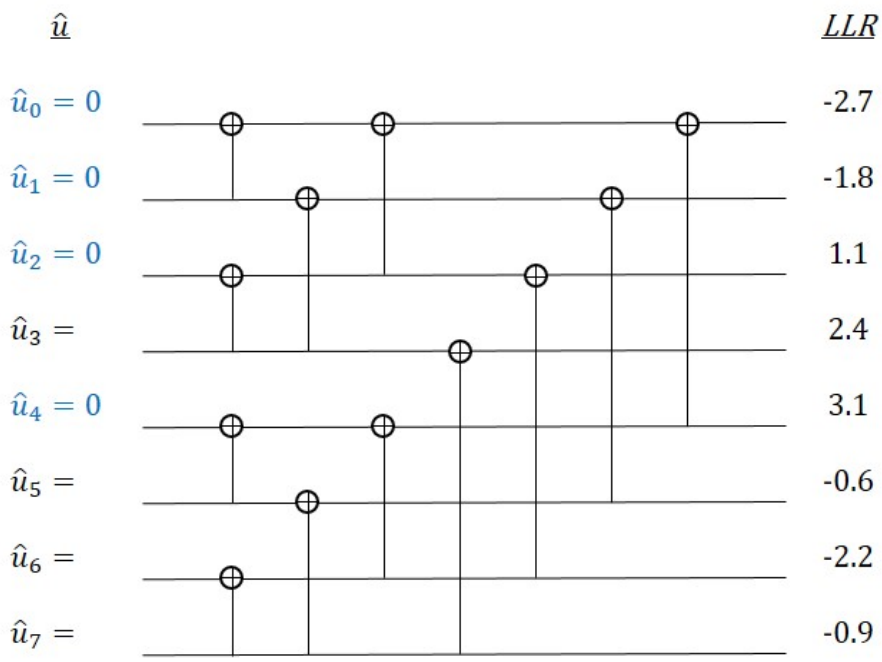


Figure 2.9. Setup of Polar Decoding Example.

Examining the LLR values on the right-hand side, it is evident that additive noise is present as they would all have a magnitude of two in a noiseless channel. Furthermore, a bit error has been introduced to the fifth LLR value (-0.6), which is represented in the value being assigned the incorrect sign during demodulation. In order to make bit estimates for message bits $\hat{u}_3, \hat{u}_5, \hat{u}_6$, and $\hat{u}_7$ the polar code graph is traversed generally from right to left, top to bottom, with computations performed according to the node type outlined in Figure 2.10.

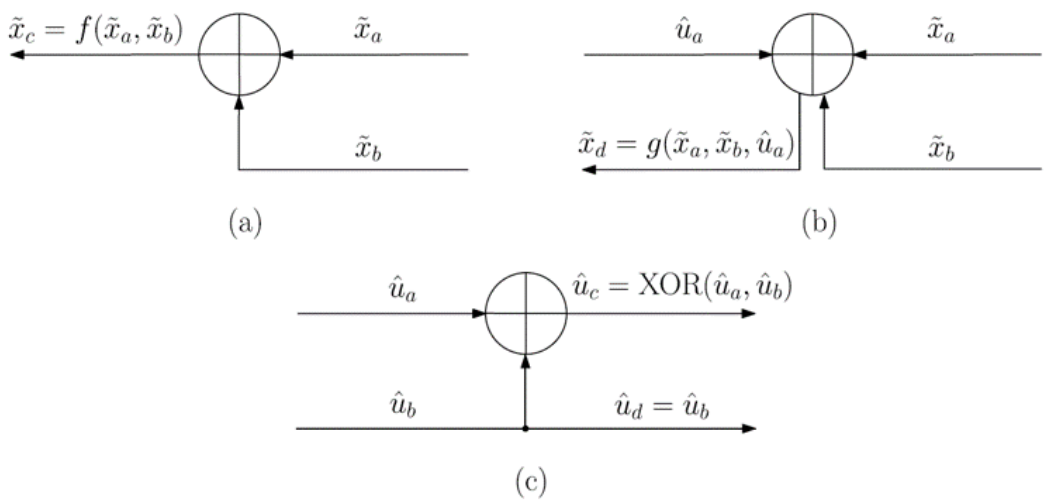


Figure 2.10. Polar Decoding Node Types and Computations. Source: [23].

The first type of computation that can be performed during decoding is the f function, depicted in Figure 2.10(a). The f function is used when the LLR of two values right of a node $\tilde{x}_a$ and $\tilde{x}_b$ are known, the desired output is the LLR left of a node $\tilde{x}_c$, and is defined as

$$\begin{aligned}
\tilde{x}_c &= f(\tilde{x}_a, \tilde{x}_b) \\
&= 2\tanh^{-1}\left(\tanh\left(\frac{\tilde{x}_a}{2}\right)\tanh\left(\frac{\tilde{x}_b}{2}\right)\right) && (2.6a) \\
&= \tilde{f}(\tilde{x}_a, \tilde{x}_b) + \log(1 + \exp(-|\tilde{x}_a + \tilde{x}_b|)) - \log(1 + \exp(-|\tilde{x}_a - \tilde{x}_b|)) && (2.6b) \\
&\approx \tilde{f}(\tilde{x}_a, \tilde{x}_b), && (2.6c)
\end{aligned}$$

where

$$\tilde{f}(\tilde{x}_a, \tilde{x}_b) \triangleq \mathrm{sgn}(\tilde{x}_a)\mathrm{sgn}(\tilde{x}_b)\min(|\tilde{x}_a|, |\tilde{x}_b|). \tag{2.7}$$

Equation 2.6b is a numerically stable calculation of 2.6a, which is then further simplified to Equation 2.6c [23]. With regard to Equation 2.7,

$$\operatorname{sgn}(x) = \begin{cases} -1 & \text{if } x < 0 \\ 0 & \text{if } x = 0 \\ 1 & \text{if } x > 0. \end{cases} \tag{2.8}$$

The second type of computation that can be performed during decoding is the g function, depicted in Figure 2.10(b). The g function is used when the LLR values right of a node $\tilde{x}_a$ and $\tilde{x}_b$ and the bit estimate left of a node $\hat{u}_a$ are known, the desired output is the LLR value left of a node $\tilde{x}_d$, and is defined as

$$\tilde{x}_d = g(\tilde{x}_a, \tilde{x}_b, \hat{u}_a) = (-1)^{\hat{u}_a} \tilde{x}_a + \tilde{x}_b. \tag{2.9}$$

Both f and g functions are used while traversing the polar code graph from right to left. Once the left side of the polar code graph has been reached, and provided the index $\hat{u}_i$ is a data index, the LLR value computed is used to make an estimate of $\hat{u}_i$ according to

$$\begin{aligned} \tilde{x}_i \geq 0 &\rightarrow \hat{u}_i = 0 \\ \tilde{x}_i < 0 &\rightarrow \hat{u}_i = 1. \end{aligned} \tag{2.10}$$

As mentioned at the start of the decoding process, if $\hat{u}_i$ is a frozen index, the value of $\hat{u}_i$ is set to zero regardless of what the computed LLR value of $\tilde{x}_i$ is.

The third type of computation that can be performed during decoding is the partial sum calculation, depicted in Figure 2.10(c). The partial sum calculation is used when the bit estimates left of a node $\hat{u}_a$ and $\hat{u}_b$ are known, the desired output are the bit estimates right of a node $\hat{u}_c$ and $\hat{u}_d$, and is defined as

$$\hat{u}_c = \hat{u}_a \oplus \hat{u}_b$$
$$\hat{u}_d = \hat{u}_b. \tag{2.11}$$

Recursively applying the three computations to nodes in the polar code graph will successively make estimates for each bit $\hat{u}_i$. A completed polar code graph, with values displayed for each of the relevant nodes where a computation was required, is illustrated in Figure 2.11 where f functions produce outputs highlighted in black, g in green, and partial sum in red.

Figure 2.11. Completed Polar Decoding Example.

Examining the decoded message $\hat{u}_0^7$, and comparing it to our example message (u_0^7 = 00010101), we see that it has been correctly decoded. In order to understand the effect of the additive noise in the decoding process, of particular interest in Figure 2.11 is the first message bit $\hat{u}_0$ and the corresponding LLR value immediately to the right (-0.6). $\hat{u}_0$ is a frozen bit, so $\hat{u}_0$ is assigned a value of zero and the bit estimate rules given in Equation

2.11 are not followed. Forcing all frozen bits to zero is a key component to correcting errors in the decoded message. In a noiseless channel each of the message bits, to include frozen bits, would correctly decode to their respective values based off of the bit estimate rules. The fact that we had to force a frozen bit to be zero-valued is an indication that one or more bit errors were present in the received symbols.

Further examining and comparing the encoded message d_0^7 and the received demodulated LLR values, we can determine that the error is in the sixth LLR value due to it having the incorrect sign. This error is illustrated through Figure 2.12. The third bit pair transmitted ($d_4 d_5$) had a value of 00, which in turn should correlate to received demodulated LLR values that map out to the first quadrant, not the fourth.

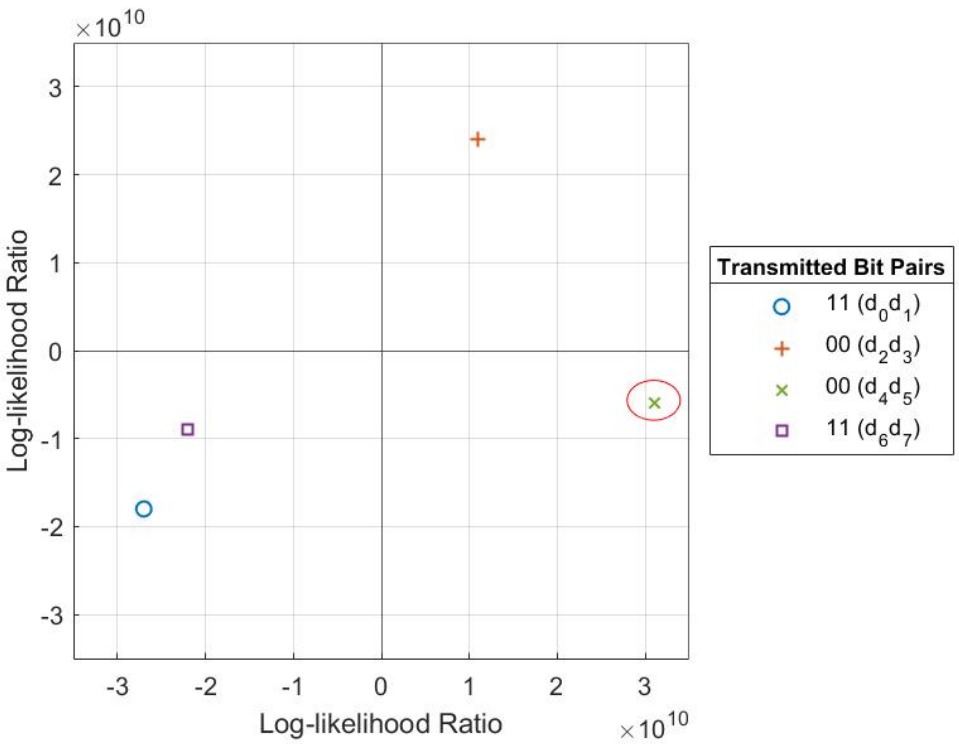


Figure 2.12. Example Demodulated LLR Values and Corresponding Bit-Pairs.

Now that the mechanics of polar decoding have been presented, it is beneficial to recall the interleaving of the CRC parity bits p_0^{L-1} (unscrambled) and ${p'}_0^{L-1}$ (scrambled with RNTI) discussed in Section 2.1.2. As the polar decoding is performed in a successive manner from

the top of the polar code graph to the bottom, the interleaving of the scrambled CRC parity bits ${p'_0}^{L-1}$ allows the decoding process to be aborted at the moment an error is detected. As a bit-value determination is made for each bit $\hat{u}_i$ that corresponds to a scrambled CRC parity bit $\hat{p}'_i$, the parity of $\hat{p}'_i$ should match that of the sum of parity of the respective data bits if $\hat{p}'_i = \hat{p}_i$. Were the RNTI used in scrambling p_i to cause $p_i \neq p'_i$, then the parity of $\hat{p}'_i$ should not match that of the sum of the parity of the respective data bits.

2.2.6 DCI Message Recovery

Provided that no errors have been detected during the decoding process, the DCI message is now extracted for processing. Frozen bits are removed, and the data and CRC parity bits are deleaved through the reverse application of the interleaving process discussed in Section 2.1.2. After the message bits are deleaved, the CRC parity bits are removed, and the DCI message is now ready to be processed by the UE.

Throughout Chapter 2 the underlying theory and concepts pertaining to the PDCCH encoding and decoding processes were presented, to include working through a realistic example using an arbitrary 12-bit data payload. Each of the steps in the process serves a specific purpose, whether it is error detection in the case of CRC parity bits or error correction in the case of polar encoding and decoding, with the overall endstate of enabling the successful transmission of a DCI message from a gNB to a UE. Chapter 3 will examine how specific steps in the process can be modified to achieve a different endstate, namely to allow an eavesdropper to determine the RNTI associated with a given DCI message.

CHAPTER 3: Methodology

This chapter is adapted from [1], previously published for the 2020 ICASSP, © 2020 IEEE.

The objective of Chapter 3 is to provide an efficient methodology that allows both the RNTI and DCI data bits of a given PDCCH transmission to be determined. A "brute force" approach to determining this information could be taken, where a receiver applies the scrambling sequence associated with the lowest legal RNTI value (1) to a received transmission and attempts to decode the message. If the message is able to be decoded without error, the correct RNTI is that which the applied scrambling sequence was initialized with. If the message is unable to be decoded, the scrambling sequence is initialized with the next sequential RNTI value and decoding is attempted again. While this approach will eventually yield the RNTI associated with the transmitted message, it is inefficient in the sense that each incorrect decoding attempt requires working through the polar decoding process until a CRC parity bit does not match its associated data bits (see Section 2.2.5). A more computationally-efficient approach, which this chapter establishes, is to use proprieties of the polarization matrix to determine error patterns in the frozen bit indices that when compared against a generated syndrome table serve to identify the RNTI associated with a transmission.

In Chapter 2 an overview of the PDCCH encoding and decoding processes was presented. Section 2.2.5 specifically discussed the mechanics of polar decoding and presented an example where the transmitted message was successfully recovered despite a bit error caused by the presence of additive noise. Chapter 3 extends the example from Section 2.2.5 to a scenario where an eavesdropper attempts to determine the scrambling sequence for a received PDCCH transmission using a modified polar decoding process. To facilitate this process, the creation of a syndrome table will be outlined, which maps the decoded error patterns to possible scrambling sequences. The syndrome concepts and methodology will then be extended from the 8-bit example message to a realistic PDCCH example using the parameters provided in Table 2.1. Finally, techniques for optimizing the syndrome methodology in the presence of unfavorable signal-to-noise ratio (SNR) environments will be discussed.

3.1 Modified Polar Decoding Process

Central to the methodology outlined in this chapter is the use of a modified polar decoding process in place of standard polar decoding techniques. At the cost of losing the error-correcting properties of standard polar decoding, the modified polar decoding process will provide a means to gain insight into the scrambling sequence applied to the codeword prior to transmission. Throughout the presentation of the modified polar decoding process, it is important to understand the two types of noise that are considered, and what their respective impacts are.

First, the scrambling sequence applied to the codeword prior to modulation will itself be viewed as a type of non-random noise. It is non-random in the sense that a specific scrambling sequence will always result in specific bit parity changes occurring at specific bit indices. Interpreting the scrambling sequence as non-random noise will provide for the identification of error patterns in frozen index locations, allowing for mapping back to pre-determined tables in order to determine the scrambling sequence that was applied.

Second, due to the implementation of the modified decoding process that lacks error-correcting properties, the ability of the methodology to correctly identify the RNTI associated with a transmission becomes sensitive to additive channel noise levels that result in bit errors. Throughout sections 3.1 — 3.3 it will be assumed that transmission occurs in a noiseless channel in order to prevent the occurrence of bit errors. Section 3.4 will discuss several techniques while in the presence of additive channel noise levels that result in bit errors during transmission.

To begin the discussion on the modified polar decoding process, recall the simplified eight-bit example from Section 2.2.5, where a given data sequence is 1101 and frozen bits are inserted into u_0, u_1, u_2, and u_4 to form $u_0^7 = 00010101$. Polar encoding occurs next as illustrated in Equation 3.1

$$d_0^7 = u_0^7 G_8 = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \tag{3.1}$$

$$d_0^7 = 11000011.$$

In order to understand the effect of a given scrambling sequence, let $s_0^7 = 00000100$. The scrambling sequence is applied to the codeword d_0^7 to form x_0^7 per

$$x_0^7 = d_0^7 \oplus s_0^7 = 1100000011 \oplus 00000100 = 11000111. \tag{3.2}$$

x_0^7 is subsequently modulated to QPSK symbols and transmitted over a noiseless channel. Assuming that no bit errors occur during transmission allows for the complex symbols to be demodulated and mapped directly back to the bit values $\hat{x}_0^7 = 11000111$, which is equivalent to x_0^7. At this point, a descrambling sequence is not applied to $\hat{x}_0^7$, but rather simply let $\hat{d}_0^7 = \hat{x}_0^7$. By not applying a descrambling sequence to $\hat{x}_0^7$, and by assuming a noiseless channel, any bit value differences between d_0^7 and $\hat{d}_0^7$ are directly attributed to the scrambling sequence s_0^7 that was applied to d_0^7.

d_0^7	1	1	0	0	0	0	1	1
$\hat{d}_0^7$	1	1	0	0	0	1	1	1

To an eavesdropper who would not have *a priori* information of d_0^7, however, there is no constructive information to be gained solely from $\hat{d}_0^7$. This is because individual indices

of $\hat{d}_0^7$ do not represent meaningful information by themselves but are rather simply part of the larger codeword. Therefore, it is desirable to transform the codeword $\hat{d}_0^7$ into a message by undoing the effects of the polarization matrix so that individual indices have specific meaning (e.g., frozen and data indices). Let this new message be z_0^7, which will come to be known as an error pattern. Substituting d_0^7 with $\hat{d}_0^7$ and replacing u_0^7 with z_0^7 in Equation 3.1 yields the transformation from $\hat{d}_0^7$ to z_0^7

$$\begin{aligned} \hat{d}_0^7 &= z_0^7 G_8 \\ \hat{d}_0^7 G_8^{-1} &= z_0^7 G_8 G_8^{-1} \\ \hat{d}_0^7 G_8 &= z_0^7 \end{aligned} \tag{3.3}$$

provided that the polarization matrix G_8 is involutory. This property is readily shown by examining $G_2 = \left(\begin{smallmatrix} 1 & 0 \\ 1 & 1 \end{smallmatrix}\right)$ in Equation 3.4

$$G_2^{-1} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}^{-1} = \frac{1}{(1)(1) - (0)(1)} \begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix}, \tag{3.4}$$

and by noting that the additive inverse of 1 in $\mathbb{Z}/2$ is itself produces

$$G_2^{-1} = \begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} = G_2. \tag{3.5}$$

When the polarization matrix is recursively expanded depending upon the polar encoding length N, according to $G_N = G_2^{\otimes n}$ where $n = \log_2 N$, it can be shown that each matrix G_N is also involutory.

Using Equation 3.3 the error pattern z_0^7 can be directly determined as

$$z_0^7 = \hat{d}_0^7 G_8 = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \tag{3.6}$$

$$z_0^7 = 11011001.$$

Thus far the modified decoding process has taken a received PDCCH transmission as its input and output an error pattern z_0^7 that, due to the noiseless channel assumption, any bit value discrepancies between it and the original message u_0^7 are directly attributable to the scrambling sequence s_0^7. Interpreting and leveraging the error pattern z_0^7 in order to determine the scrambling sequence applied is discussed in the next section.

3.2 Error Patterns and Syndromes

It is important at this point to note the differences between the message u_0^7 and the error pattern z_0^7, where blue highlighting denotes frozen bit indices as shown below.

u_0^7	0	0	0	1	0	1	0	1
z_0^7	1	1	0	1	1	0	0	1

The involutory nature of G_N provides that the frozen and data index locations for both u_0^7 and z_0^7 remain the same throughout the process. Due to the application of the scrambling sequence s_0^7 after polar encoding, and then the omission of a descrambling sequence prior to determining z_0^7, several of both of the frozen and data indices in z_0^7 reflect different values than u_0^7. A definitive determination can be made on the effect that s_0^7 had on the frozen indices, namely the parity of bits z_0, z_1, and z_4 were swapped and the parity of z_2 remained

the same. This determination is possible, because as these indices are the frozen indices, they should all represent bit values of zero. No determination can be made for the data indices z_3, z_5, z_6, and z_7 at this time, however, as a receiver would not normally have *a priori* knowledge of the value of the data bits being transmitted, and as such has no reference point for what their values should reflect.

In order to interpret what the bit values of the frozen indices in the error pattern z_0^7 represent, the concept of a syndrome is now introduced. For the purposes of this paper, a syndrome ${z'_0}^7$ is defined as an error pattern that is solely dependent upon a corresponding scrambling sequence. Being solely dependent upon a scrambling sequence means that any effect that data bit values may have had on the syndrome during generation have been removed. The simplest manner to generate a syndrome, which is valid for the current eight-bit example, is to multiply a given scrambling sequence by the polarization matrix. In this manner Equation 3.7 determines the syndrome ${z'_0}^7$ that corresponds with the scrambling sequence $s_0^7 = 00000100$

$$
\begin{aligned}
{z'_0}^7 &= s_0^7 G_8 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \\
{z'_0}^7 &= 11001100.
\end{aligned}
\tag{3.7}
$$

The interpretation of ${z'_0}^7$ is that an error pattern that reflects the zeroth, first, fourth, and fifth bits having swapped parity and the second, third, sixth, and seventh bits having unchanged parity determines that that the scrambling sequence $s_0^7 = 00000100$ was applied to the codeword prior to modulation.

Now that a single syndrome has been derived, the next step is to generate a table that

maps each possible scrambling sequence to its respective syndrome. For the given example, define the scrambling sequence as all combinations of ones or zeros that are eight bits in length, less the trivial sequences of all ones or all zeros. The number of possible scrambling sequences, and consequentially the size of the set of syndromes is

$$|\mathcal{S}| = \sum_{1 \leq k \leq 7} \binom{8}{k} = 254 \tag{3.8}$$

where $\mathcal{S}$ represents the set of all syndromes. Therefore, the set of syndromes $\mathcal{S}$ will be N rows by $|\mathcal{S}|$ columns in size. Each column in the table is generated by taking a possible scrambling sequence and applying Equation 3.7. Note that the only requirement to construct the syndrome table $\mathcal{S}$ for this simplified example is knowledge of the message length N, similar to how an UE is configured to process DCI messages, and what the set of possible scrambling sequences are.

With the syndrome set $\mathcal{S}$ determined, recall the error pattern z_0^7 that was previously determined. The next step is to compare the reference points of the error pattern (i.e., frozen indices) to each syndrome $z_0'^7$ that comprise the syndrome table $\mathcal{S}$. Figure 3.1 illustrates the matching results of this comparison.

Error Pattern z_0^7									Scrambling Sequence
1	1	0	-	1	-	-	-	→	?

Matching Syndromes z'^7_0									Scrambling Sequences							
1	1	0	0	1	1	0	0	→	0	0	0	0	0	1	0	0
1	1	0	0	1	0	0	0	→	1	1	0	0	1	0	0	0
1	1	0	1	1	1	1	1	→	1	0	1	0	0	0	0	1
1	1	0	1	1	0	1	0	→	1	0	0	1	0	0	1	0
1	1	0	0	1	0	1	0	→	0	1	1	0	0	0	1	0
1	1	0	0	1	1	1	1	→	0	1	0	1	0	0	0	1
1	1	0	1	1	0	0	0	→	0	0	1	1	1	0	0	0
1	1	0	1	1	1	0	1	→	0	0	0	0	1	0	1	1
1	1	0	1	1	1	0	0	→	1	1	1	1	0	1	0	0
1	1	0	1	1	0	0	1	→	1	1	0	0	0	1	1	1
1	1	0	0	1	1	1	0	→	1	0	1	0	1	1	1	0
1	1	0	0	1	0	1	1	→	1	0	0	1	1	1	0	1
1	1	0	1	1	0	1	1	→	0	1	1	0	1	1	0	1
1	1	0	1	1	1	1	0	→	0	1	0	1	1	1	1	0
1	1	0	0	1	0	0	1	→	0	0	1	1	0	1	1	1
1	1	0	0	1	1	0	1	→	1	1	1	1	1	0	1	1

Figure 3.1. Subset of Matching Syndromes $\mathcal{S}'$ and Respective Scrambling Sequences.

Figure 3.1 displays the 16 matching syndromes ${z'_0}^7$ whose frozen index values are equivalent to that of the error pattern. Note that data indices are not able to be used in the comparison between the error pattern and the syndromes. The only indices that can be used in comparison are the frozen indices. This is because frozen indices, by definition, should have a bit value of zero. Anytime that an error pattern contains a frozen index with a value of one it is evidence that the scrambling sequence changed that specific index. Furthermore, any time that an error pattern contains a frozen index with a value of zero it is evidence that the scrambling sequence did not change that respective index, which is also helpful in determining a match. Data indices can be set to either zero or one, and as such cannot serve as a reference point like the frozen indices can. Refining the possible scrambling sequences to 16 out of 254 represents progress in determining the specific sequence used prior to modulation. The simplicity of this example compared to the PDCCH encoding process, specifically the manner in that the set of possible scrambling sequences is defined and the absence of CRC parity bits, prevents any further refining of the 16 candidates to determine the actual scrambling sequence used. Section 3.3 details how further refinement is possible when examining PDCCH messages.

Finally, recall from Equation 3.7 that each syndrome ${z'_0}^7$ was generated directly by multiplying the corresponding scrambling sequence s_0^7 with the polarization matrix G_8. While this approach works for the simple eight-bit example presented, it does not work for PDCCH applications. This is due to the sub-block interleaving and rate-matching steps that occur in the PDCCH process that have to be taken into consideration. The Query Syndrome Table step in Section 3.3.1 outlines the additional steps required to generate the syndrome table $\mathcal{S}$ for the PDCCH.

3.3 Application to PDCCH

The discussed methodology for determining the scrambling sequence applied to a received codeword is now extended to an example where the RNTI associated with a PDCCH message is determined, while maintaining the noiseless channel assumption. The process will first be presented in its entirety using the parameters specified in Table 2.1, notably where $E = 128$ (no rate-matching) and RNTI = 100. Next, additional considerations for each of the possible rate-matching schemes will be discussed. Figure 3.2 illustrates the process flow for applying the discussed methodology in a noiseless channel.

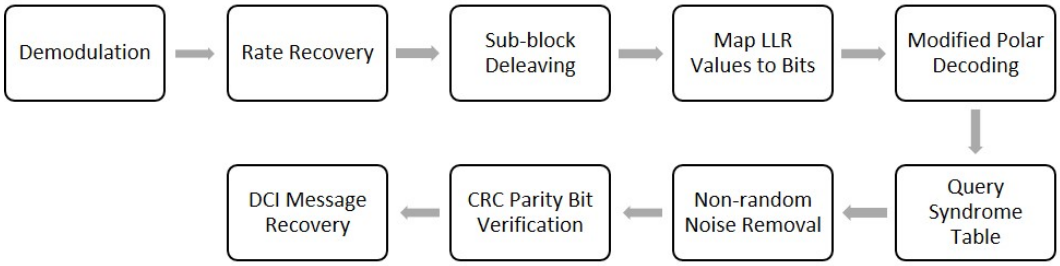


Figure 3.2. RNTI Determination Process Flow in Noiseless Channel.

3.3.1 No Rate-Matching

Applying the parameters specified in Table 2.1, recall from Section 2.1.3 that u_0^{N-1} was formed as shown below.

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	0	0	0	0	0	0	0	1	0	1	0	0
1	1	1	1	0	1	0	0	0	1	0	0	0	0	1	0	1	0	0	0																

Reliability Freezing Data Scrambled CRC Parity

With $N = 128$ and $K = 36$ there are $N - K = 92$ frozen indices in u_0^{N-1} that will serve as reference points and assist in matching an error pattern to a syndrome. Subsequent processing steps for u_0^{N-1} involved polar encoding, sub-block interleaving, scrambling, and modulation. As stated in the beginning of this section, transmission occurs in a noiseless channel.

Demodulation, Rate-Recovery, and Sub-block Deleaving

At this point an eavesdropper receives the transmitted complex-valued symbols, then demodulates to LLR values and performs sub-block deleaving to form j_0^{E-1}.

2	2	2	-2	-2	-2	2	2	-2	2	2	-2	2	2	2	2	-2	2	-2	2	2	-2	2	-2	-2	-2	2	-2	2	-2	2	-2	-2	-2	2	-2
2	2	-2	-2	-2	2	-2	-2	2	2	2	2	-2	-2	2	-2	-2	2	2	-2	-2	-2	2	-2	2	2	-2	-2	-2	-2	-2	2	2	-2	2	-2
2	-2	-2	-2	2	2	-2	2	2	2	2	-2	-2	2	2	-2	-2	2	2	2	-2	-2	-2	2	2	-2	2	-2	2	2	2	-2	2	2	-2	-2
-2	2	-2	2	-2	-2	2	-2	-2	-2	-2	2	2	2	2	-2	2	2	-2	2																

Since $E = N$, there are no rate-recovery steps necessary.

Map LLR Values to Bits

The eavesdropper does not know which scrambling sequence was applied to the codeword prior to transmission, so no descrambling sequence is applied at this point. Since $E = N$ (no rate-matching), h_0^{E-1} is now mapped back to bits (bit value of zero maps to positive LLR value and one maps to negative (Section 2.2.1)) to form $\hat{d}_0^{N-1}$.

0	0	0	1	1	1	0	0	1	0	0	1	0	0	0	0	1	0	1	0	0	1	0	1	1	1	0	1	0	1	0	1	1	1	0	1
0	0	1	1	1	0	1	1	0	0	0	0	1	1	0	1	1	0	0	1	1	1	0	1	0	0	1	1	1	1	1	0	0	1	0	1
0	1	1	1	0	0	1	0	0	0	0	1	1	0	0	1	1	0	0	0	1	1	1	0	0	1	0	1	0	0	0	1	0	0	1	1
1	0	1	0	1	1	0	1	1	1	1	0	0	0	0	1	0	0	1	0																

Modified Polar Decoding

The eavesdropper can now determine the error pattern z_0^{N-1}, by applying Equation 3.6 such that

$$z_0^{N-1} = \hat{d}_0^{N-1} G_N. \tag{3.9}$$

z_0^{N-1}, with frozen indices highlighted in blue, is determined as shown below.

1	1	1	1	1	0	1	0	1	0	1	1	1	0	1	0	0	1	0	1	1	1	1	1	0	1	0	1	0	0	1	0	1	0	1	0
1	1	1	0	0	0	0	1	1	1	0	1	0	0	1	0	0	1	1	0	1	0	1	1	1	1	1	1	0	1	1	1	0	0	1
0	1	1	1	1	1	0	0	1	1	1	0	1	1	0	1	0	0	1	1	0	1	0	0	1	0	1	1	1	0	0	1	0	0	1	0
1	0	0	0	0	0	0	0	0	1	0	0	0	1	0	1	1	0	1	0																

Of note, the data and scrambled CRC parity bits of z_0^{N-1} are not highlighted in their respective colors as in previous sections. While the relative positioning of the data and parity bits in z_0^{N-1} has not changed, the bit values contain an unknown amount of errors in them from the non-random noise introduced by the scrambling sequence. As such, the data and parity bits cannot be used in any constructive manner until the error pattern is correctly mapped to a syndrome and the effects of the non-random noise are removed.

Query Syndrome Table

Recall from Section 3.2 that in the eight-bit example the error pattern determined had 16 matching candidate syndromes. Were there a proportional relationship between codeword length and number of syndrome candidates, it would quickly become unmanageable when dealing with longer codewords. For DCI messages, however, for a given η_{ID} value and codeword length there are only 32768 possible unique scrambling sequences (see Section 2.1.7). Therefore, the number of possible syndromes is $|\mathcal{S}| = 32768$ where $\mathcal{S}$ represents the set of all syndromes. A syndrome table N rows by $|\mathcal{S}|$ columns, with each column referring to the respective RNTI used to initialize the scrambling sequence, can readily be constructed.

As mentioned at the end of Section 3.2, compared to the simplified eight-bit example there are additional steps required to generate a PDCCH syndrome table due to the sub-block interleaving and rate-matching processes. When $E = N$ (no rate-matching), the scrambling sequence s_0^{E-1} needs to be deleaved (Section 2.2.4) before it can be multiplied by the polarization matrix G_N in order to form the syndrome $z_0'^{N-1}$. This process is repeated for each of the possible scrambling sequences to generate the set of syndromes $\mathcal{S}$. The syndrome table generated will be specific to the cell scrambling ID value η_{ID} used to initialize the scrambling sequences, as well as the length E.

When $E \neq N$, the process of generating the syndrome table $\mathcal{S}$ is slightly more involved. This is partly due to the length of the scrambling sequence s_0^{E-1} no longer matching that of the polar code length N. In order to generate the syndrome table, first generate an arbitrary data payload a_0^{A-1} and arbitrary RNTI value, and follow all of the PDCCH encoding steps outlined in Chapter 2. While the data payload a_0^{A-1} and RNTI values chosen for syndrome generation are arbitrary, the syndrome table $\mathcal{S}$ will be specific to the rate-matching codeword length E and the cell scrambling ID η_{ID} values selected. Transmission over a noiseless channel occurs, followed by demodulation. Next, 2^{15} copies of the demodulated LLR values have a unique descrambling sequence applied to each of them, where the scrambling sequences are initialized by sequentially higher RNTI values beginning at one. The original instance of the demodulated LLR values is left without a descrambling sequence applied, termed a control set. This will be used later to remove the effects of the arbitrarily selected data payload a_0^{A-1} from earlier. Rate-recovery, sub-block deleaving, and modified polar decoding are next conducted on both the control set and each set of LLR values. After modified polar

decoding, there have been $1 + 2^{15}$ sequences generated, corresponding to the control set and each of the 2^{15} descrambling sequences that were applied. In order to transform these sequences into the set of syndromes $\mathcal{S}$, the effects of the data payload a_0^{A-1} are removed by an XOR operation between each sequence and the control set. After this XOR operation the syndrome table $\mathcal{S}$ generation is complete.

Once the syndrome table has been generated, the frozen indices from z_0^{N-1} are compared to respective bit locations in each column of the table $\mathcal{S}$ until a match is found. In the example, this comparison returns the matching syndrome $z_0'^{N-1}$ that maps to a scrambling sequence initialized by a RNTI value of 100. Figure 3.3 displays both the error pattern z_0^{N-1} and the matching syndrome $z_0'^{N-1}$.

z_0^{N-1}	1	1	1	1	1	0	1	0	1	0	1	1	1	0	1	0	0	1	0	1	1	1	1	1	0	1	0	1	0	0	1	0	1	0	1	0
	1	1	1	0	0	0	0	1	1	1	0	1	0	0	1	0	0	1	1	0	1	0	1	1	1	1	1	1	0	1	1	1	0	0	1	
	0	1	1	1	1	1	0	0	1	1	1	0	1	1	0	1	0	0	1	1	0	1	0	0	1	0	1	1	1	0	0	1	0	0	1	0
	1	0	0	0	0	0	0	0	0	1	0	0	0	1	0	1	1	0	1	0																
$z_0'^{N-1}$ (RNTI = 100)	1	1	1	1	1	0	1	0	1	0	1	1	1	0	1	0	0	1	0	1	1	1	1	1	0	1	0	1	0	0	1	0	1	0	1	0
	1	1	1	0	0	0	0	1	1	1	0	0	0	0	1	0	0	1	1	0	1	0	1	0	1	0	0	1	1	0	1	1	1	0	0	1
	0	1	1	1	1	1	0	1	1	1	1	0	1	1	0	1	0	0	1	0	0	0	1	0	1	0	1	1	1	0	0	0	0	1	1	0
	0	1	1	1	0	1	0	0	0	0	0	0	0	1	1	1	0	0	1	0																

Figure 3.3. Comparison of Error Pattern z_0^{N-1} and Matching Syndrome $z_0'^{N-1}$.

All of the frozen index values (highlighted in blue) match between z_0^{N-1} and $z_0'^{N-1}$ due to the noiseless channel assumption. That is, the non-random noise introduced from the scrambling sequence initialized by RNTI = 100 results in the frozen index bit parities of the error pattern z_0^{N-1} matching those of the syndrome $z_0'^{N-1}$ that was generated with an equivalent RNTI value. This comparison is enabled by the knowledge that frozen indices are by definition set to zero during encoding, allowing them to serve as reference points for the error pattern. In contrast, the remaining index values (not highlighted) do not match. During encoding, unlike the frozen indices, the data and CRC indices were both zero and one valued. The non-random noise from the scrambling sequence results in bit changes to the data and CRC indices, but the eavesdropper is unable to determine what these changes are from the error pattern alone as it has no *a priori* knowledge of what the bits in question were set to initially.

Next recall from Section 2.1.7 the equation that determines the scrambling sequence initialization value, presented again as Equation 3.10

$$c_{init} = (\text{RNTI} \times 2^{16} + \eta_{\text{ID}}) \bmod 2^{31}. \tag{3.10}$$

The effect of the mod 2^{31} operator is to remove the most significant bit from the RNTI. This means that identical scrambling sequences will be generated for RNTI values of 100 and $100 + 2^{15}$. In order to determine which of the two RNTIs were associated with the received DCI message, the data and CRC parity bits must be recovered and examined.

Non-random Noise Removal

While under the noiseless channel assumption, and with the matching syndrome ${z'_0}^{N-1}$ for the computed error pattern z_0^{N-1} determined, the recovered message $\hat{u}_0^{N-1}$ can be readily determined by Equation 3.11

$$\hat{u}_0^{N-1} = z_0^{N-1} \oplus {z'_0}^{N-1}. \tag{3.11}$$

This direct determination of the recovered message $\hat{u}_0^{N-1}$ from the error pattern z_0^{N-1} is only possible with the noiseless channel assumption. The syndrome ${z'_0}^{N-1}$ provides information on whether any bit index's parity is changed, not just the frozen indices. When comparing an error pattern z_0^{N-1} with a syndrome ${z'_0}^{N-1}$ only the reference points, or frozen indices can be used. But once the matching syndrome is determined, and again only in a noiseless channel, the syndrome can be used to remove the non-random noise from all of the indices of the error pattern (i.e., frozen, data, and CRC).

The recovered message $\hat{u}_0^{N-1}$ is formed as shown below.

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	0	0	0	0	0	0	0	1	0	1	0	0
1	1	1	1	0	1	0	0	0	1	0	0	0	0	1	0	1	0	0	0																

Reliability Freezing Data Scrambled CRC Parity

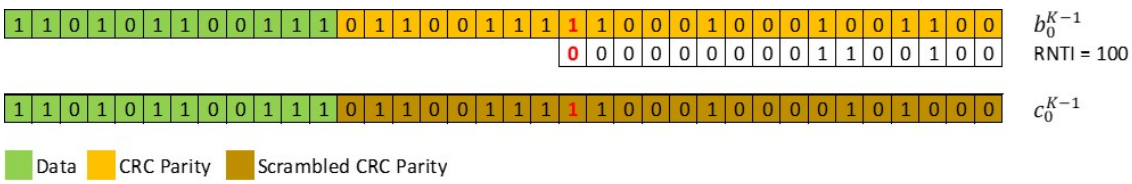
Of note, the data and scrambled CRC parity bits are now highlighted, as the values in these indices have had the effects of the scrambling sequence removed from them and are now

correct. The recovered message $\hat{u}_0^{N-1}$ is equivalent to the original message u_0^{N-1}.

CRC Parity Bit Verification

With the non-random noise removed from the error pattern z_0^{N-1} and the recovered message $\hat{u}_0^{N-1}$ formed, certain CRC bits must still be examined in order to determine whether the scrambling sequence was generated from a RNTI of 100 or 100 + 2^{15}. The CRC bits are also used as the mechanism to detect any bit errors that may have occurred throughout any stage of the process.

Recall the relationship between the CRC parity bits and the RNTI from Section 2.1.2, shown again below.



In the given example where the RNTI value is 100, the most significant bit of the RNTI has a bit value of zero (red font), and there is no change between p_8 and p'_8 (both in red font) during scrambling. Therefore, when the sum of the parity of the data bits corresponding to p'_8 are compared with the parity of p'_8, they should match. If the parity between the compared bits does not match, that would imply that there is a difference between p_8 and p'_8, and subsequently that the RNTI applied to the DCI message must have a most significant bit value of one.

Further recall Figure 2.2 from Section 2.1.2, presented again as Figure 3.4 with p_8 and the corresponding data indices highlighted in red.

p_0	p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8	p_9	p_{10}	p_{11}	p_{12}	p_{13}	p_{14}	p_{15}	p_{16}	p_{17}	p_{18}	p_{19}	p_{20}	p_{21}	p_{22}	p_{23}
-	a_0	-	a_0	a_0	a_0	a_0	a_0	a_0	a_0	-	a_0	-	-	-	a_0	-	a_0	-	-	a_0	a_0	a_0	a_0
a_1	a_1	a_1	a_1	-	a_1	a_1	-	a_1	-	a_1	a_1	-	-	-	-	-	-	a_1	-	a_1	a_1	-	-
-	a_2	a_2	a_2	a_2	-	a_2	a_2	-	a_2	-	a_2	a_2	-	-	-	-	-	-	a_2	-	a_2	a_2	-
-	-	a_3	a_3	a_3	a_3	-	a_3	a_3	-	a_3	-	a_3	a_3	-	-	-	-	-	-	a_3	-	a_3	a_3
a_4	a_4	-	-	-	a_4	a_4	a_4	a_4	-	-	-	a_4	a_4	a_4	-	a_4	-	-	-	a_4	a_4	a_4	-
-	a_5	a_5	-	-	-	a_5	a_5	a_5	a_5	-	-	-	a_5	a_5	a_5	-	a_5	-	-	-	a_5	a_5	a_5
a_6	a_6	a_6	-	a_6	-	-	-	a_6	-	a_6	a_6	a_6	-	a_6	a_6	-	-	a_6	-	a_6	-	-	-
-	a_7	a_7	a_7	-	a_7	-	-	-	a_7	-	a_7	a_7	a_7	-	a_7	a_7	-	-	a_7	-	a_7	-	-
-	-	a_8	a_8	a_8	-	a_8	-	-	-	a_8	-	a_8	a_8	a_8	-	a_8	a_8	-	-	a_8	-	a_8	-
-	-	-	a_9	a_9	a_9	-	a_9	-	-	-	a_9	-	a_9	a_9	a_9	-	a_9	a_9	-	-	a_9	-	a_9
a_{10}	a_{10}	-	a_{10}	-	a_{10}	a_{10}	a_{10}	a_{10}	a_{10}	-	a_{10}	-	-	a_{10}	a_{10}	-	-	a_{10}	a_{10}	a_{10}	-	-	a_{10}
a_{11}	-	a_{11}	a_{11}	-	-	a_{11}	-	a_{11}	-	a_{11}	a_{11}	-	-	-	a_{11}	-	-	-	a_{11}	-	a_{11}	a_{11}	a_{11}

Figure 3.4. Relationship Between CRC Parity Bit p_8 and its Corresponding Data Bits.

In order for the eavesdropper to test whether the RNTI associated with the received DCI message is 100 or $100 + 2^{15}$, the sum of the parities of data bits $a_0, a_1, a_3, a_4, a_5, a_6, a_{10}$, and a_{11} (highlighted in red) from $\hat{u}_0^{N-1}$ are compared to the parity of bit p'_8 from $\hat{u}_0^{N-1}$. As mentioned earlier, if the parity between the data bits and p'_8 match, the RNTI value is 100, otherwise the RNTI value is $100 + 2^{15}$. The relevant data and scrambled CRC parity bits are highlighted red in $\hat{u}_0^{N-1}$, keeping in mind that the data and scrambled CRC parity bits are still interleaved (see Section 2.1.2) at this time.

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	0	0	0	0	0	0	0	1	0	1	0	0
1	1	1	1	0	1	0	0	0	1	0	0	0	0	1	0	1	0	0	0																

Reliability Freezing Data Scrambled CRC Parity

Comparing the parity of the data bits in red font with that of the CRC parity bit in red font, it is evident that the parity matches and the RNTI is determined to be 100.

DCI Message Recovery

The last step for the eavesdropper to perform is to recover the DCI message into a usable format. This step is described in Section 2.2.6, and the process remains the same for this application. A final check for bit errors can be conducted by dividing the parity bits p_0^{L-1} by a_0^{A-1} in $\mathbb{Z}/2$.

3.3.2 Repetition

Recall from Section 2.2.3 that, with a receiver operating under normal conditions, the scheme of repetition provided a degree of signal processing gain to those bit indices that had been repeated. This processing gain is not realizable, however, to an eavesdropper applying the modified decoding process. To illustrate this disparity, let $E = 144$ as in the repetition example from Section 2.1.6, where the rate-matched codeword e_0^{E-1} was formed as shown below.

0	0	0	0	1	0	1	0	1	1	1	1	0	0	1	0	1	0	0	1	0	1	0	0	0	0	1	0	1	0	0	0	0	1	1	0
0	1	1	0	0	0	1	1	0	0	1	1	1	0	0	1	1	0	0	1	0	0	0	0	0	0	0	0	1	0	1	1	1	0	1	1
0	0	1	0	0	0	1	0	1	0	1	1	1	0	1	1	1	1	1	0	1	1	1	0	0	0	0	0	1	0	1	0	1	1	1	1
0	0	1	0	1	0	0	1	0	1	0	0	0	0	1	0	1	0	0	0	0	0	0	0	1	0	1	0	1	1	1	1	0	0	1	0

Repeated Bits

The next step in the encoding process is to apply the scrambling sequence s_0^{E-1} to form the scrambled codeword x_0^{E-1}.

0	0	0	1	1	1	0	0	1	0	0	1	1	0	1	0	0	0	0	0	0	1	0	1	1	1	0	1	0	1	0	1	1	1	0	1
1	1	1	0	0	0	1	1	0	1	0	1	1	0	1	1	0	1	1	1	0	0	0	0	0	0	1	0	1	1	0	1	0	0	0	1
1	0	0	1	1	0	0	1	1	1	0	1	1	0	0	0	0	0	1	1	1	1	1	0	0	1	0	1	0	0	0	1	0	0	1	1
1	1	0	1	1	0	1	0	1	1	1	0	0	0	0	1	0	0	1	0	1	0	1	0	0	1	1	1	0	0	1	1	1	1	0	1

Repeated Bits

Examining x_0^{E-1}, due to the application of the scrambling sequence, it is evident that the $E-N$ highlighted repeated bits are no longer equivalent to the first $E-N$ bits of the codeword as they were for e_0^{E-1}. This disparity is normally removed during the descrambling step of the decoding process. However, an eavesdropper who is applying the modified decoding process, where a descrambling sequence is not applied to a received transmission, is unable to constructively use the $E - N$ repeated bits. In this instance, rate-recovery consists of simply truncating and discarding the demodulated LLR values from length $E = 144$ to

length $N = 128$. The remaining steps are identical to those of Section 3.3.1.

3.3.3 Puncturing

Maintaining consistency with the puncturing example from Chapter 2, let $E = 86$ and recall that u_0^{N-1} was determined as shown below.

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	1	1	0	0	0	0	0	0	0	1	0	1	0	0
1	1	1	1	0	1	0	0	0	1	0	0	0	0	1	0	1	0	0	0																

Punctured Freezing Reliability Freezing Extra Freezing Data Scrambled CRC Parity

Polar encoding, sub-block interleaving, rate-matching, scrambling, and modulation steps per Chapter 2 are conducted and the codeword is transmitted under noiseless channel conditions. An eavesdropper receives the transmission and demodulates the complex-valued symbols into LLR values (exponent 10^{10} omitted for brevity) as shown below.

						2	2	-2	-2	2	-2	-2	-2	-2	-2	-2	2	2	2	2	2	-2	2	2	2	2	2	2	2	-2	-2	-2	-2	-2	-2
2	-2	2	2	-2	2	2	2	-2	2	-2	-2	-2	-2	-2	2	-2	2	2	-2	-2	-2	-2	2	-2	2	2	2	-2	2	-2	2	2	-2	-2	2
-2	2	2	2	2	2	-2	-2	-2	2	-2	2	-2	-2	-2	-2	2	2	2	2																

Rate-recovery outputs the following codeword.

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	2	2	-2	-2	2	-2	-2	-2	-2	-2	-2	2	2	2	2	2	-2	2	2	2	2	2	2	2	-2	-2	-2	-2	-2	-2
2	-2	2	2	-2	2	2	2	-2	2	-2	-2	-2	-2	-2	2	-2	2	2	-2	-2	-2	-2	2	-2	2	2	2	-2	2	-2	2	2	-2	-2	2
-2	2	2	2	2	2	-2	-2	-2	2	-2	2	-2	-2	-2	-2	2	2	2	2																

Sub-block deleaving outputs the following codeword.

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	2	2	-2	-2	-2	-2	2	2	-2	2	2	2	-2	-2	2	-2	2	2	-2	2	-2	-2	-2	2	2	-2	0	0	0	0	-2	-2	2	-2
-2	2	2	2	2	2	2	2	-2	-2	-2	-2	-2	2	2	2	-2	-2	-2	2	-2	-2	-2	2	-2	2	2	2	-2	2	-2	2	2	-2	-2	2
2	2	-2	-2	-2	2	2	2	-2	2	-2	2	-2	-2	-2	-2	2	2	2	2																

LLR to bit mapping outputs $\hat{d}_0^{N-1}$.

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	1	1	1	0	0	1	0	0	0	1	1	0	1	0	0	1	0	1	1	1	0	0	1	0	0	0	0	1	1	0	1
1	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	1	1	1	0	1	1	1	0	1	0	0	0	1	0	1	0	0	1	1	0
0	0	1	1	1	0	0	0	1	0	1	0	1	1	1	1	0	0	0	0																

Modified polar decoding outputs the error pattern z_0^{N-1}.

0	0	0	0	1	0	0	1	1	0	1	1	0	1	1	0	0	0	0	1	1	1	1	1	1	0	1	1	1	0	0	1	1	0	1	0
0	1	0	0	0	0	1	1	1	0	0	0	1	0	0	0	1	0	0	1	1	0	1	1	0	1	1	1	1	0	0	0	1	0	0	0
1	0	1	0	1	0	1	1	0	0	1	0	0	1	0	0	0	0	0	1	1	1	1	0	0	0	1	0	0	1	0	1	0	0	1	0
0	1	0	1	1	0	1	1	0	0	1	0	0	0	0	1	0	0	0	0																

Punctured Freezing Reliability Freezing Extra Freezing

Note that the different frozen indices of z_0^{N-1} are highlighted in their respective shades of blue. When comparing the frozen indices of an error pattern that had puncturing implemented, only the reliability and extra freezing indices may be used for comparison against the syndrome table $\mathcal{S}$, not the punctured freezing indices. This is due to any error pattern information that would be contained in the punctured bits being removed during the process of rate-matching and rate-recovery. Additionally, the message length N and codeword to be modulated length E that are present in the received transmission must match those that were used in the construction of the syndrome table $\mathcal{S}$.

After the matching syndrome $z_0'^{N-1}$ is determined from $\mathcal{S}$, the recovered codeword $\hat{u}_0^{N-1}$ is determined as shown below.

1	0	0	0	0	1	0	1	0	1	0	0	1	1	1	1	0	1	0	0	0	1	0	0	0	0	1	0	0	1	0	0	0	0	1	0
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	1	1	1	0	1	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	1	1	0	0	0	0	0	0	0	1	0	1	0	0
1	1	1	1	0	1	0	0	0	1	0	0	0	0	1	0	1	0	0	0																

Punctured Freezing Reliability Freezing Extra Freezing Data Scrambled CRC Parity

All of the bit values in $\hat{u}_0^{N-1}$ are equivalent to those of u_0^{N-1} with the exception of the punctured freezing bits. This is intuitive, as the error pattern information is removed from the punctured bits during rate-matching and rate-recovery, causing the application of the syndrome $z_0'^{N-1}$ to result in erroneous values for those indices (which is why they were not used during syndrome comparison).

3.3.4 Shortening

Maintaining consistency with the shortening example from Chapter 2, let $E = 74$ and recall that u_0^{N-1} was determined as shown below.

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	1	1	0	1	0	0	0	0	0
0	0	1	0	0	1	1	0	1	1	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																	

Shortened Freezing Reliability Freezing Data Scrambled CRC Parity

Polar encoding, sub-block interleaving, rate-matching, scrambling, and modulation steps per Chapter 2 are conducted and the codeword is transmitted under noiseless channel conditions. An eavesdropper receives the transmission and demodulates the complex-valued symbols into LLR values (exponent 10^{10} omitted for brevity).

2	-2	2	-2	2	2	-2	2	2	-2	2	-2	2	-2	-2	-2	2	2	-2	-2	-2	2	2	2	2	2	-2	2	-2	-2	-2	-2	-2	-2	-2	2
2	-2	2	2	-2	2	-2	2	-2	2	-2	2	2	2	2	2	-2	-2	2	-2	2	-2	2	2	2	2	2	-2	-2	2	-2	-2	-2	2	-2	2
-2	-2																																				

Rate-recovery outputs (exponent 10^{20} omitted from indices with a value of 9 for brevity) the codeword shown below.

2	-2	2	-2	2	2	-2	2	2	-2	2	-2	2	-2	-2	-2	2	2	-2	-2	-2	2	2	2	2	2	-2	2	-2	-2	-2	-2	-2	-2	-2	2
2	-2	2	2	-2	2	-2	2	-2	2	-2	2	2	2	2	2	-2	-2	2	-2	2	-2	2	2	2	2	2	-2	-2	2	-2	-2	-2	2	-2	2
-2	-2	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9
9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9																		

Sub-block deleaving outputs the following codeword.

2	-2	2	-2	2	2	-2	2	2	-2	2	-2	2	2	-2	-2	2	-2	-2	-2	-2	2	2	2	2	2	-2	2	-2	-2	-2	-2	-2	-2	-2	2
-2	2	-2	2	2	2	2	2	2	-2	2	2	-2	2	-2	-2	-2	-2	9	9	9	9	9	9	9	9	9	2	-2	2	2	-2	2	-2	2	2
-2	-2	2	-2	2	2	2	-2	-2	2	-2	2	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9
9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9																		

LLR to bit mapping outputs $\hat{d}_0^{N-1}$.

0	1	0	1	0	0	1	0	0	1	0	1	0	0	1	1	0	1	1	1	1	0	0	0	0	0	1	0	1	1	1	1	1	1	1	0
1	0	1	0	0	0	0	0	0	1	0	0	1	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	1	0
1	1	0	1	0	0	0	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																		

Modified polar decoding outputs the error pattern z_0^{N-1}.

0	1	0	0	0	0	0	1	0	1	0	1	0	1	1	1	0	0	0	1	1	1	0	1	1	0	1	1	0	0	0	1	1	0	0	1
1	0	1	0	1	1	0	0	1	1	0	0	1	0	0	1	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	1	1	0	1	
0	1	0	0	1	1	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																

Shortened Freezing Reliability Freezing

Note that the different frozen indices of z_0^{N-1} are highlighted in their respective shades of blue. When comparing the frozen indices of a message that had shortening implemented, only the reliability freezing indices may be used for comparison against the syndrome table $\mathcal{S}$, not the shortened freezing indices. This is due to the process of shortening, where rate-recovery will always insert LLR values that map to zero in the shortened indices. These zeros that are inserted carry through the modified polar decoding process also as zeros. Simply put, regardless of the scrambling sequence applied before transmission, the shortened bits, when rate-recovered, will always have a value of zero. Also, similar to puncturing, the message length N and codeword to be modulated length E that are present in the received transmission must match those that were used in the construction of the syndrome table $\mathcal{S}$.

After the matching syndrome $z_0'^{N-1}$ is determined from $\mathcal{S}$, the recovered codeword $\hat{u}_0^{N-1}$ is determined as shown below.

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	1	1	0	1	0	0	0	0	0
0	0	1	0	0	1	1	0	1	1	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																

Shortened Freezing Reliability Freezing Data Scrambled CRC Parity

All of the bit values in $\hat{u}_0^{N-1}$ are equivalent to those of u_0^{N-1}.

3.3.5 Effect of Rate-Matching on Methodology Performance

Now that examples have been provided that implement the presented methodology for the different rate-matching schemes, it is constructive to examine what the effects of rate-matching are on the performance of the methodology. Recall that in both the puncturing and shortening examples, a subset of the frozen indices in the error pattern z_0^{N-1} were unable to be matched against the syndrome table $\mathcal{S}$ due to the manner in which rate-recovery is

performed. Table 3.1 summarizes the parameters that were used in each of the examples in sections 3.3.1 — 3.3.4.

Table 3.1. Effects of Rate-Matching Schemes on the Quantity of Usable Error Pattern Frozen Indices.

Scheme	N	E	Total Frozen Indices	Restricted Frozen Indices	Usable Frozen Indices
No Rate-Matching	128	128	92	0	92
Repetition	128	144	92	0	92
Puncturing	128	86	92	42	50
Shortening	128	74	92	54	38

The no rate-matching and repetition schemes have equivalent methodology performance, due to the previously discussed fact that the scrambled, repeated bits are unable to constructively be used in the modified decoding process and are discarded. Puncturing and shortening, however, do potentially have an adverse effect on methodology performance in that there are reduced quantities of frozen indices in the error pattern z_0^{N-1} that can be used for comparison purposes. For the presented examples, in a noiseless channel, 50 and 38 error pattern frozen indices are more than adequate quantities to determine a matching syndrome. This may not always be the case once additive noise is considered however. Section 3.4.1 will examine the effect of additive noise on an error pattern, and Chapter 4 will investigate and quantify the impact that rate-matching schemes have on methodology performance.

3.4 Methodology Optimization in Presence of Additive Noise

Throughout this chapter so far the noiseless channel assumption has been applied. By definition, a noiseless channel results in no bit errors being introduced to the codeword during transmission due to additive noise. This allowed the sole change to a message after encoding but before decoding to be wholly attributable to the scrambling sequence that was applied. This single source of non-random noise in turn facilitated the modified polar decoding process, since there was not a requirement to correct bit errors during

decoding. While a noiseless channel environment is beneficial for academic purposes, it is also recognized as not being a realistic assumption.

In order to provide a functional methodology in the presence of additive noise, two optimization techniques can be used with the previously described process. First, the quantity and location of the frozen indices in the error pattern that are compared to the syndrome table are adjusted. By only requiring a subset of the error pattern to match a syndrome, it is possible to still find a correct match in the presence of additive noise. Second, an approach is taken where the eavesdropper employs multiple antennas that simultaneously collect the same transmission. By increasing the number of antennas, the processing gain of the received signal is increased, helping to mitigate the effects of the additive noise.

3.4.1 Syndrome Bit Quantity and Location

Figure 3.5 illustrates the process flow for identifying the RNTI from a given transmission in the presence of additive noise.

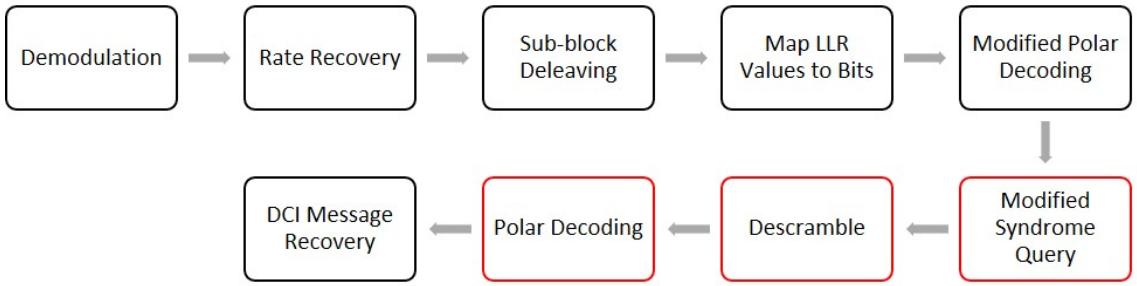


Figure 3.5. RNTI Determination Process Flow in Presence of Additive Noise (Changes from Figure 3.2 Highlighted in Red).

Note that the Modified Syndrome Query block will use a subset of the error pattern indices to compare against the syndrome table instead of the complete set. Also note that after the syndrome table has been queried, and provided a matching syndrome is found, the received transmission must then be descrambled and processed via standard polar decoding. While a syndrome matching an error pattern can often be determined when bit errors occur in the

presence of additive noise, the modified polar decoding process does not correct any bit errors. Before the DCI message can be recovered, and in order to verify that the matching syndrome is indeed the correct syndrome, any errors that may be present in the data and CRC parity bits must first be corrected through standard polar decoding techniques.

Recall the parameters from Table 2.1, and the example from Section 2.1.4 where $E = N = 128$, with u_0^{N-1} formed as shown below.

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	0	0	0	0	0	0	0	1	0	1	0	0
1	1	1	1	0	1	0	0	0	1	0	0	0	0	1	0	1	0	0	0																

Reliability Freezing Data Scrambled CRC Parity

After polar encoding, sub-block interleaving, rate-matching, and scrambling the codeword x_0^{E-1} was formed as shown below.

0	0	0	1	1	1	0	0	1	0	0	1	1	0	1	0	0	0	0	0	0	1	0	1	1	1	0	1	0	1	0	1	1	1	0	1
1	1	1	0	0	0	1	1	0	1	0	1	1	0	1	1	0	1	1	1	0	0	0	0	0	0	1	0	1	1	0	1	0	0	0	1
1	0	0	1	1	0	0	1	1	1	0	1	1	0	0	0	0	0	1	1	1	1	1	0	0	1	0	1	0	0	0	1	0	0	1	1
1	1	0	1	1	0	1	0	1	1	1	0	0	0	0	1	0	0	1	0																

Demodulation

Next, x_0^{E-1} is modulated and transmitted with a SNR of 8 dB present. Figure 3.6 illustrates the pairs of demodulated LLR values that an eavesdropper receives.

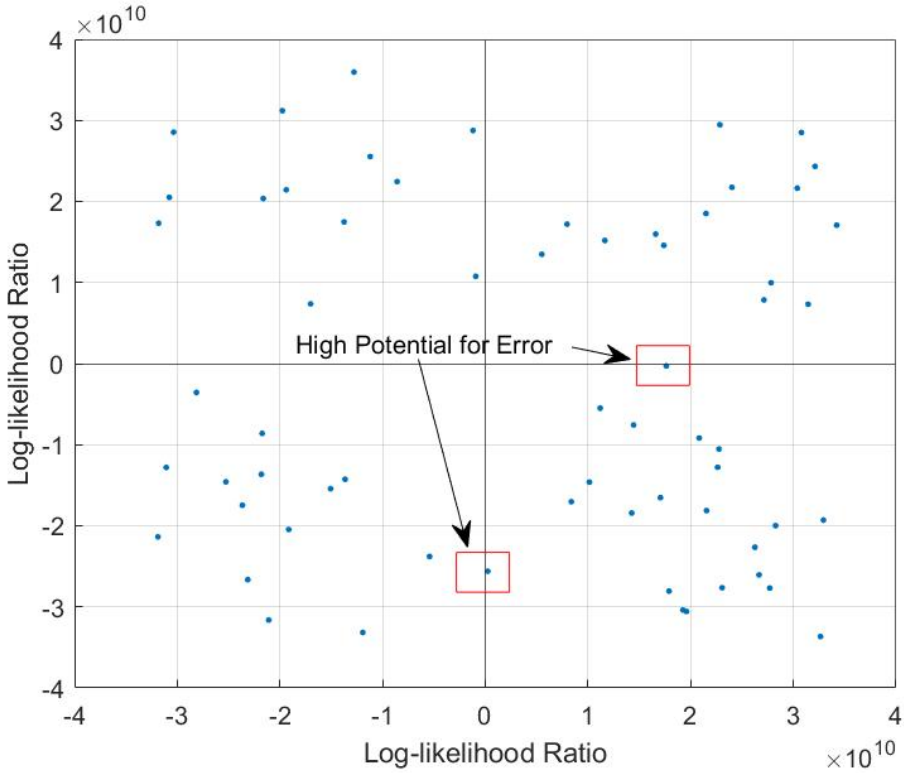


Figure 3.6. Demodulated LLR Values with 8 dB AWGN Present.

Examining Figure 3.6 it is evident that there are several LLR values close to zero in magnitude, which is an indicator that bit errors may have been introduced during transmission. The received complex-valued symbols are next demodulated in order to form the LLR values h_0^{E-1} (exponent 10^{10} omitted for brevity).

3.1	2.8	2.8	-2.8	-2.3	-2.7	1.2	1.5	-2.0	3.1	0.8	-1.7	-2.8	-0.4	-1.1	2.6	3.1	0.7	2.1	1.9	2.2	-1.8	2.7	-2.6	-2.5	-1.5	1.4	-0.8	2.3	-2.8	3.3	-1.9	-2.2	-0.9	1.7	-1.7
-1.4	-1.4	-1.9	2.1	0.6	1.3	-1.2	-3.3	2.6	-2.3	1.8	-2.8	-0.1	2.9	-2.4	-1.7	2.3	-1.1	-1.9	-2.0	2.8	1.0	3.0	2.2	0.8	1.7	-1.7	0.7	-0.5	-2.4	1.0	-1.5	2.4	2.2	2.0	-3.1
-1.3	3.6	3.3	-3.4	-3.2	1.7	1.9	-3.0	-3.2	-2.1	2.1	-0.9	-2.2	2.0	1.7	1.6	1.7	1.5	0.1	-2.6	-3.1	-1.3	-3.1	2.1	1.4	-1.8	1.1	-0.5	2.7	0.8	2.8	-2.0	2.3	2.9	-2.2	-1.4
-2.1	-3.2	1.8	0.0	-3.0	2.9	-1.4	1.7	-1.5	-1.5	-0.1	1.1	3.2	2.4	2.3	-1.3	3.4	1.7	-0.9	2.2																

The eavesdropper has no way of identifying bit errors that have been introduced into the LLR values h_0^{E-1} at this point, however for example purposes the two errors present are highlighted in red (h_{13} and h_{90}). Comparing h_0^{E-1} with x_O^{E-1}, and recalling that in the absence of noise positive LLR values map to bit values of 0 and negative maps to 1, it is evident that the signs of the two highlighted LLR values are incorrect ($h_{13} = -0.4$ which maps to 1 but $u_{13} = 0$, $h_{90} = 0.1$ which maps to 0 but $u_{90} = 1$) which indicates errors have

been introduced.

Rate-recovery and Sub-block Deleaving

Were there a difference between E and N rate-recovery would be performed at this step. For the presented example there is no need for rate-recovery. Recall that if puncturing or shortening had been present in the received codeword, the set of frozen indices in the error pattern compared against the syndrome table would need to be adjusted per sections 3.3.3 or 3.3.4 during the relevant steps.

Sub-block deleaving outputs the codeword shown below.

3.1	2.8	2.8	-2.8	-2.3	-2.7	1.2	1.5	-2.0	3.1	0.8	-1.7	3.1	0.7	2.1	1.9	-2.8	-0.4	-1.1	2.6	2.2	-1.8	2.7	-2.6	-2.5	-1.5	1.4	-0.8	2.3	-2.8	3.3	-1.9	-2.2	-0.9	1.7	-1.7
0.6	1.3	-1.2	-3.3	-0.1	2.9	-2.4	-1.7	2.8	1.0	3.0	2.2	-0.5	-2.4	1.0	-1.5	-1.3	3.6	3.3	-3.4	-3.2	-2.1	2.1	-0.9	1.7	1.5	0.1	-2.6	-1.4	-1.4	-1.9	2.1	2.6	-2.3	1.8	-2.8
2.3	-1.1	-1.9	-2.0	0.8	1.7	-1.7	0.7	2.4	2.2	2.0	-3.1	-3.2	1.7	1.9	-3.0	-2.2	2.0	1.7	1.6	-3.1	-1.3	-3.1	2.1	1.4	-1.8	1.1	-0.5	2.7	0.8	2.8	-2.0	2.3	2.9	-2.2	-1.4
-3.0	2.9	-1.4	1.7	-2.1	-3.2	1.8	0.0	-1.5	-1.5	-0.1	1.1	3.2	2.4	2.3	-1.3	3.4	1.7	-0.9	2.2																

Map LLR Values to Bits

LLR to bit mapping outputs $\hat{d}_0^{N-1}$ as shown below.

0	0	0	1	1	1	0	0	1	0	0	1	0	0	0	0	1	1	1	0	0	1	0	1	1	1	0	1	0	1	0	1	1	1	0	1
0	0	1	1	1	0	1	1	0	0	0	0	1	1	0	1	1	0	0	1	1	1	0	1	0	0	0	1	1	1	1	0	0	1	0	1
0	1	1	1	0	0	1	0	0	0	0	1	1	0	0	1	1	0	0	0	1	1	1	0	0	1	0	1	0	0	0	1	0	0	1	1
1	0	1	0	1	1	0	1	1	1	1	0	0	0	0	1	0	0	1	0																

Modified Polar Decoding and Modified Syndrome Query

Modified polar decoding occurs next that outputs z_0^{N-1}. Figure 3.7 illustrates both z_0^{N-1} and the matching syndrome for $z_0'^{N-1}$ for RNTI = 100.

z_0^{N-1}

1	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	1	0	1	0	1	1	1	1	1	1	0	0	0	0	0	0	0
0	1	0	0	1	0	1	1	0	1	1	1	1	0	0	0	1	1	0	0	0	0	0	1	0	1	0	1	1	0	1	1	1	0	0	1
0	1	1	1	1	1	0	0	1	1	1	0	1	1	0	1	0	0	1	1	0	1	0	0	1	0	1	1	1	0	0	1	0	0	1	0
1	0	0	0	0	0	0	0	0	1	0	0	0	1	0	1	1	0	1	0																

z'^{N-1}_0 (RNTI = 100)

1	1	1	1	1	0	1	0	1	0	1	1	1	0	1	0	0	1	0	1	1	1	1	1	0	1	0	1	0	0	1	0	1	0	1	0
1	1	1	0	0	0	0	1	1	1	0	0	0	0	1	0	0	1	1	0	1	0	1	0	1	0	0	1	1	0	1	1	1	0	0	1
0	1	1	1	1	1	0	1	1	1	1	0	1	1	0	1	0	0	1	0	0	0	1	0	1	0	1	1	1	0	0	0	0	1	1	0
0	1	1	1	0	1	0	0	0	0	0	0	0	1	1	1	0	0	1	0																

Figure 3.7. Comparison of Error Pattern z_0^{N-1} and Matching Syndrome $z_0'^{N-1}$ in Presence of Additive Noise.

Inspection of the frozen indices (highlighted in blue) reveals discrepancies between z_0^{N-1} and $z_0'^{N-1}$. This is different from the previous example in Figure 3.3, where there was a noiseless channel and all frozen index values between z_0^{N-1} and $z_0'^{N-1}$ were equivalent. The discrepancies between frozen index values in Figure 3.7 are a result of the two errors that were introduced during transmission, which have now propagated throughout the modified decoding process.

In order to more clearly illustrate the similarities and differences between z_0^{N-1} and $z_0'^{N-1}$, Figure 3.8 illustrates z_0^{N-1} with additional descriptive information.

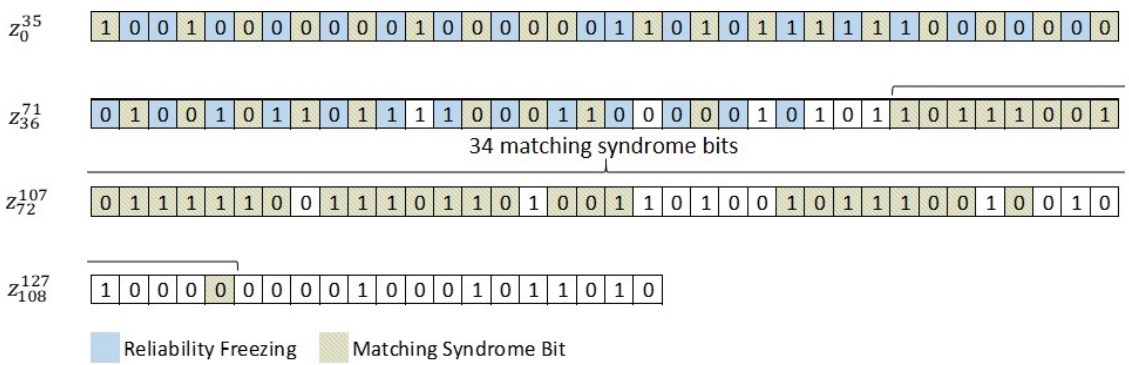


Figure 3.8. Examination of Error Pattern Bits of z_0^{N-1}.

Each error pattern frozen index of z_0^{N-1} whose value matches the respective index of $z_0'^{N-1}$ is highlighted in yellow. Frozen indices whose values do not match $z_0'^{N-1}$ are highlighted in blue. Those indices that are not highlighted represent data and CRC parity bits that have an unknown number of errors in them, rendering them unusable in the current format.

The eavesdropper was able to determine the correct syndrome $z_0'^{N-1}$ by not requiring every frozen bit in the error pattern z_0^{N-1} to match a value in the syndrome table $\mathcal{S}$, but rather by only searching for a matching subset of consecutive frozen indices of sufficient length. Examining the first row of frozen indices, there are few consecutive indices whose values of the error pattern equal that of the syndrome. Beginning on the second row, however, and extending towards the last frozen index there are 34 consecutive frozen indices that match a syndrome. Note that the data and CRC parity bits are ignored when comparing an error pattern to a syndrome, the only requirement is that the subset be comprised of consecutive frozen indices. This method of examining the frozen indices between the error pattern z_0^{N-1} and the syndrome $z_0'^{N-1}$ is known as determination of the longest common substring (LCS).

Additionally, an alternate determination technique of examining the Hamming distance of the frozen indices between the error pattern and syndrome will also be analyzed and compared against that of the LCS method in Chapter 4.

It is desired to determine a threshold value for the LCS so that it is of sufficient length to reduce the probability of a Type I error (incorrect syndrome erroneously identified as a match to the error pattern), but not so long that it results in high probability of a Type II error (correct syndrome not identified). In order to begin to understand the trade space, it is generally observed that, in a noiseless channel, with each additional frozen bit considered the set of matching syndromes $\mathcal{S}' \subseteq \mathcal{S}$ is halved. This relationship is given by

$$|\mathcal{S}'| \approx 2^{15-t} \qquad 0 \leq t \leq 15 \tag{3.12}$$

where t is the number of frozen bits considered. When assuming a noiseless channel, it is generally observed that the set of matching syndromes $\mathcal{S}'$ is reduced to a single matching syndrome when $t \geq 16$. This relationship is derived from properties of the scrambling sequence being a Gold sequence, and that there are 2^{15} unique scrambling sequences (and by extension 2^{15} unique syndromes).

Chapter 4 will further examine the LCS threshold trade space, but for this example let the eavesdropper's LCS threshold value be 20. That is, any syndrome whose LCS with the error pattern is 20 bits or larger is identified as a possible match. This value will allow for a high probability of determining the correct syndrome while minimizing the probability of Type I errors. Recall from Figure 3.8 that the LCS for the syndrome mapping to RNTI = 100 is 34, which is well above the threshold value of 20, and as such the respective syndrome is determined to be a match.

For this specific instance of AWGN that has been added, were the eavesdropper to continue comparing the error pattern z_0^{N-1} against syndromes in the table $\mathcal{S}$ another match that exceeds the LCS threshold of 20 consecutive frozen indices would be found at RNTI = 5506. The syndrome ${z'}_0^{N-1}$ associated with RNTI = 5506 has 28 consecutive indices that match the error pattern z_0^{N-1}. The eavesdropper is incapable of determining which syndrome is correct at this time, as the data and CRC parity bits are still in an unusable format. During

the polar decoding step the syndrome associated with RNTI = 5506 will be determined to be incorrect, making it an example of a type I error.

Descramble

Recall Section 3.3.1, where in the non-random noise removal step the recovered message $\hat{u}_0^{N-1}$ was able to be directly determined from the error patter z_0^{N-1} and the syndrome $z_0'^{N-1}$. This direct computation of $\hat{u}_0^{N-1}$ was facilitated through the noiseless channel assumption. Now that additive noise is present, in order to verify the syndrome match and the data and CRC parity bits, the errors caused by the noise must be corrected. This is achieved by applying the scrambling sequence associated with the matching syndrome to the demodulated LLR values. After descrambling, rate-recovery (which for this example is not applicable) and sub-block deleaving are performed per sections 2.2.3 and 2.1.5.

Polar Decoding

Standard polar decoding techniques are conducted per Section 2.2.5 in order to recover the message bits in the presence of additive noise. Throughout the polar decoding process, as determinations are successively made for the data and CRC parity bits, these bits are used to ensure that any errors have indeed been removed from the recovered message. It is at this point that the syndrome associated with RNTI = 100 would be determined to be correct, as the scrambling sequence generated by this RNTI produces the correct parity for the data and CRC bits. Similarly, the syndrome associated with RNTI = 5506 would be determined to be incorrect (type I error), as the scrambling sequence generated by this RNTI causes the parity of the third CRC parity bit to be wrong.

DCI Message Recovery

After any errors have been removed from the codeword and the recovered message $\hat{u}_0^{N-1}$ has been formed, the DCI payload is extracted and interpreted by the eavesdropper.

3.4.2 Receiver Antenna Quantity

The optimization technique of examining only a subset of frozen indices in the previous section demonstrated the capability to identify the correct syndrome, and by extension the associated RNTI, in the presence of additive noise. This technique is not without fault,

however, as even a single bit error introduced during transmission can at times prevent a matching syndrome from being identified due to the non-linear manner in which errors can propagate through the modified decoding process. In order to further mitigate the effects of additive noise, a complimentary optimization technique of increasing the receiver antenna quantity is presented. Figure 3.9 illustrates the process flow.

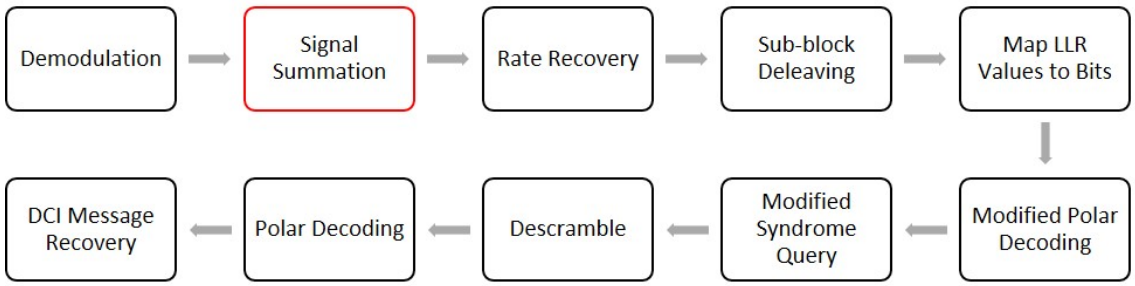


Figure 3.9. RNTI Determination Process Flow Incorporating Multiple Antennas in Presence of Additive Noise (Changes from Figure 3.2 Highlighted in Red).

The only difference between figures 3.5 and 3.9 is the addition of the Signal Summation process step, which supports the previous assertion that varying the receiver antenna quantity is complimentary to other optimization techniques.

For an example on how receiver antenna quantity helps to mitigate the effects of additive noise, consider a setup where there are three antennas that each simultaneously receive the same transmission. Each of the antennas is positioned at least 0.5λ from each other in order to maintain statistical independence. Figure 3.10 illustrates the LLR value pairing received by each antenna.

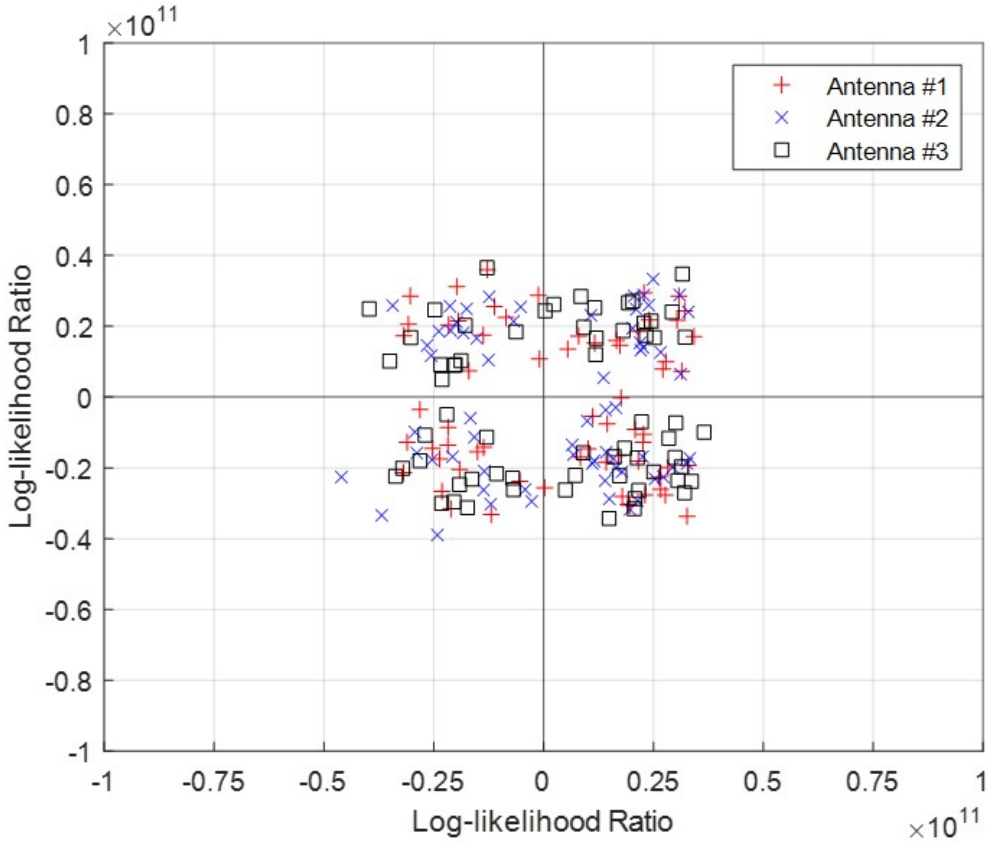


Figure 3.10. Multiple-Antenna Received LLR Value Plot.

Examining Figure 3.10 there are multiple LLR values that are very close to zero in magnitude, indication a high potential of errors being present in the transmission. Figure 3.11 illustrates the effective values received once the separate values are summed.

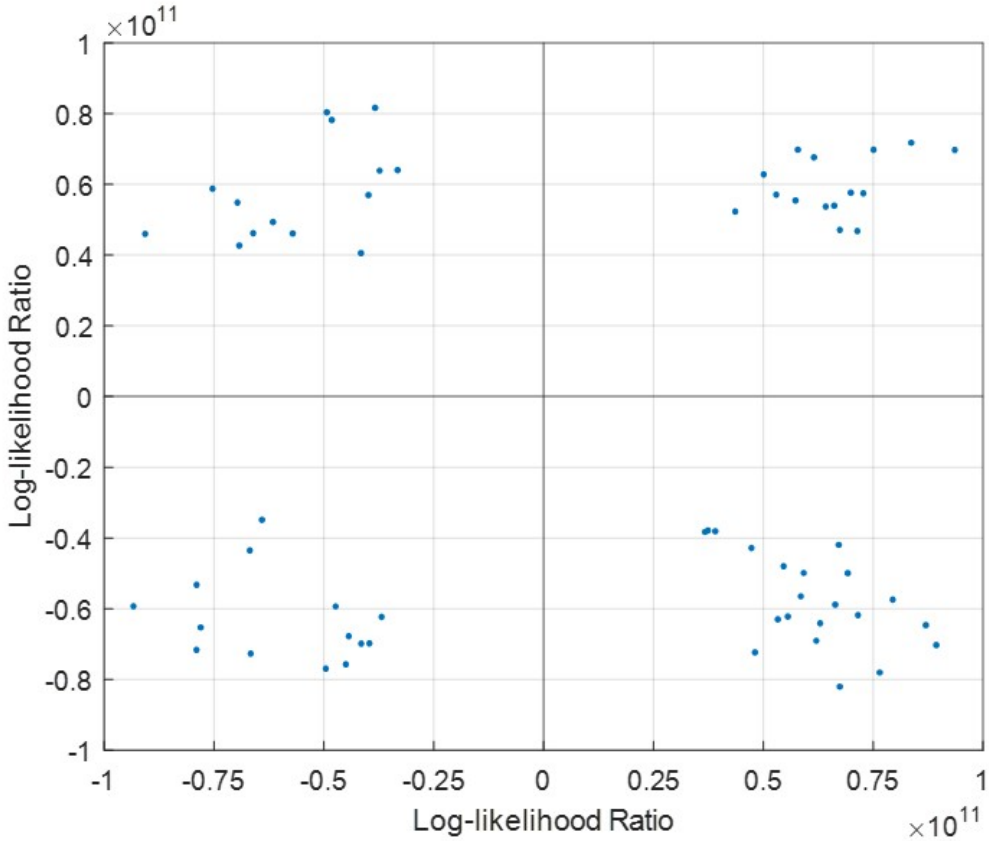


Figure 3.11. Signal Summation LLR Value Plot.

Note that the effect of summing not only removes those values whose magnitude was previously approximately zero, but it also increases the magnitude of every LLR value. This will aid in the polar decoding step, as the greater the magnitude an LLR value has, the more confidence the decoder will apply to it when making a bit determination. Chapter 4 will provide an analysis through simulation on the relationship between receiver antenna quantity and SNR.

Chapter 3 outlined the presented methodology, by which a passive eavesdropper of a network can determine the scrambling sequence, and by extension the RNTI, associated with a given received PDCCH transmission. The methodology was first presented in a noiseless transmission channel, so that the scrambling sequence could be viewed as non-random noise. A modified polar decoding process yielded an error pattern, which in turn was

compared to a table of pre-determined syndromes, enabling mapping back to the associated scrambling sequence and RNTI. Furthermore, several techniques were discussed to mitigate the effects of bit errors caused by additive noise during transmission. Chapter 4 will examine the effectiveness of the presented methodology under a variety of environmental conditions.

CHAPTER 4:
Simulation Results and Analysis

This chapter is adapted from [1], previously published for the 2020 ICASSP, © 2020 IEEE.

Now that background on the PDCCH encoding and decoding processes has been discussed and an overview of a method by which an eavesdropper to a network can determine the RNTI associated with a transmitted DCI message by examining an error pattern to a syndrome table has been presented, it is constructive to examine the effectiveness of the methodology under a variety of conditions.

4.1 Simulation Variables and Metrics

Table 4.1 provides an overview of the different variables that have been discussed in previous chapters and will be used throughout the following simulations.

Table 4.1. Simulation Variables and Associated Bounds.

Variable	Units	Minimum Value	Maximum Value
LCS Threshold	# Bits	1	Q_F
Hamming Distance Threshold	# Bits	1	Q_F
SNR	dB	-10	10
Receiver Antenna Quantity	#	1	40
Rate-Matched Frozen Index Quantities	# Bits	1	Q_R

The LCS threshold value defines which syndromes the eavesdropper will consider a possible match. An error pattern with a LCS greater than or equal to the threshold is considered a possible match. The bounds for the threshold value are between one and the total number of frozen indices Q_F. Similar to LCS, the Hamming distance threshold value defines which syndromes the eavesdropper will consider a possible match. Opposite from LCS, however, an error pattern having a Hamming distance less than or equal to the threshold is considered a possible match. The bounds are the same as for the LCS. Throughout the simulation the AWGN SNR levels in the transmission channel will be varied to examine sensitivity of the

methodology to bit errors. Overall bounds for the SNR levels are $[-10, 10]$ dB. The receiver antenna quantity will be varied from a single antenna up to a maximum of 40 antennas. Finally, recall from sections 3.3.3 (puncturing) and 3.3.4 (shortening) that the rate recovery process renders $N - E$ frozen indices unusable from an error pattern prospective. The effect that this reduced set of frozen indices Q_R has on methodology performance compared to the full set of frozen indices Q_F when rate matching is not utilized will be examined using the LCS determination method.

Figure 4.1 illustrates the simulation metrics and the manner in which the methodology can arrive at each of them. As discussed in Chapter 3, once the error pattern is determined it is successively compared against syndromes that comprise the syndrome table. If the LCS or Hamming threshold is exceeded, the CRC bits are examined via standard polar decoding methods. If the parities of all of the CRC bits match those of the respective data bits, the correct RNTI has been identified (true positive). If the parities of the CRC bits are incorrect, a false positive (Type I error) has been identified. If the LCS or Hamming threshold is not exceeded when the error pattern is compared to a given syndrome, but the syndrome was that which corresponded to the correct RNTI, a false negative (Type II error) has been identified. Lastly, a true negative is identified when the determination threshold was not exceeded for a syndrome that corresponds to an incorrect RNTI. An eavesdropper would normally continue to compare syndromes against the error pattern until a true positive is identified, or the set of syndromes has been exhausted and a true positive is not achievable due to bit errors caused by additive noise, resulting in a false negative (Type II error).

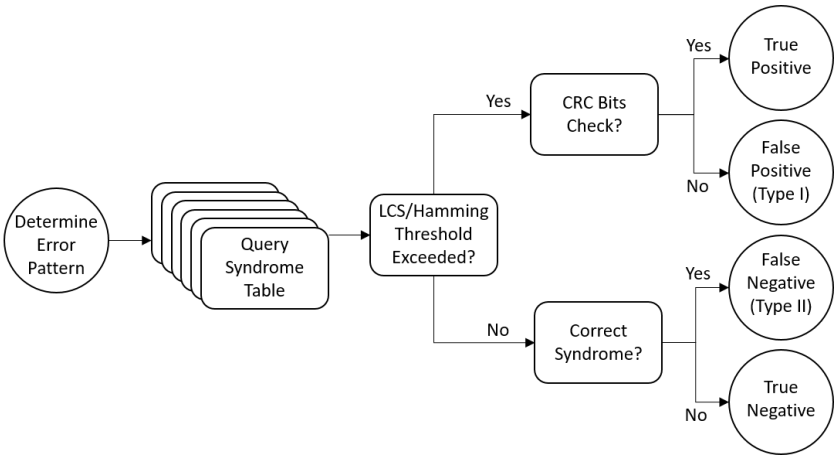


Figure 4.1. Simulation Metric Process Flow.

Table 4.2 summarizes each of the simulation metrics from Figure 4.1. True positives for this paper are defined to be synonymous with the success rate or $P_{Success}$.

Table 4.2. Simulation Metrics.

Metric	Units
True Positive	Probability
Type I Error (False Positive)	Probability
Type II Error (False Negative)	Probability
True Negative	Probability

For sections 4.2 — 4.4 it is assumed that if the threshold value is exceeded when the error pattern is compared to the syndrome that corresponds to the correct RNTI, any bit errors present can be corrected by subsequent polar decoding so that the parity of the CRC bits match their respective data bits. This implies that a false positive (Type I error) is not possible when the error pattern is compared to the syndrome corresponding to the correct RNTI. There are instances where a large number of bit errors can affect the error pattern in a non-linear fashion, such that the determination threshold value is exceeded but the successive-cancellation decoder is unable to correct all of the errors in order to verify the CRC parity bits. This scenario results in a true positive being erroneously classified as a false positive (Type I error) due to error-correcting limitations of the polar decoder. Section 4.5 addresses this uncommon scenario.

Across all of the following simulations, each data point was derived through taking the mean value of 2500 independent trials. Additionally, unless a rate-matching scheme is explicitly listed for a section, it is assumed that no rate-matching has been implemented ($E = N$).

4.2 LCS and Hamming Distance

The first simulation variables to examine are the threshold values for the decoder when comparing an error pattern with a given syndrome. Recall the example from Section 3.3.1, where there are $N - K = 128 - 36 = 92$ frozen indices in the error pattern z_0^{N-1}. Requiring all 92 indices to match would eliminate the possibility of a Type I error, but also does not allow for any additive noise to propagate into the error pattern. The following subsections

will examine the LCS and Hamming determination methods to better understand threshold values.

4.2.1 LCS Trade Space

In order to understand the LCS trade space, Figure 4.2 illustrates the success rate and Type I error rate as functions of LCS threshold values.

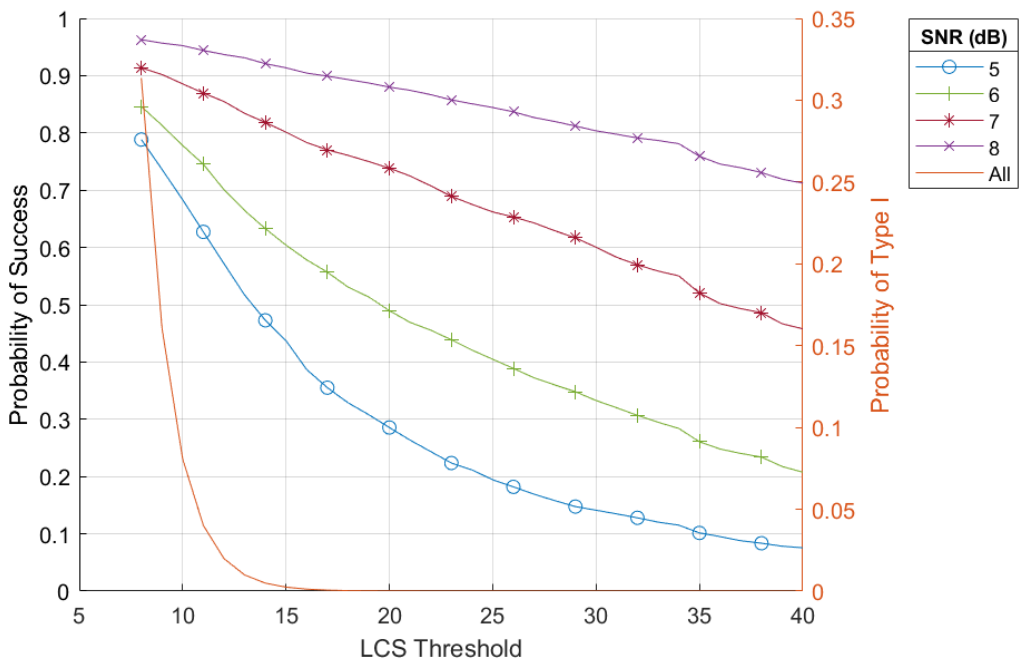


Figure 4.2. $P_{Success}$ and $P_{Type\ I}$ as a Function of LCS Threshold for Varying SNR Levels.

An error pattern and a given syndrome are deemed matches if their LCS value is greater than or equal to the threshold value. Examining Figure 4.2, it is observed that as the threshold value increases, the probability of success for each SNR level decreases. This is intuitive, as for a given SNR level, increasing the threshold decreases the quantity of syndromes that will match with the error pattern, thereby decreasing the probability that the correct syndrome is one of the matching syndromes. It is also observed that the probability of Type I errors is independent of SNR, and decreases exponentially at low threshold levels. To understand this trend, recall Equation 3.12 from Section 3.4.1 where it was observed that

for each additional frozen index considered in an error pattern, the resulting set of matching syndromes is approximately halved, up to a point. It follows that if the set of syndromes that exceed the error pattern is continually halved as the threshold is increased, the probability of identifying a false positive will be halved as well.

Figure 4.3 uses the same data set from Figure 4.2, but focuses on that portion of the Type I error curve where the exponential decrease appears to flatten.

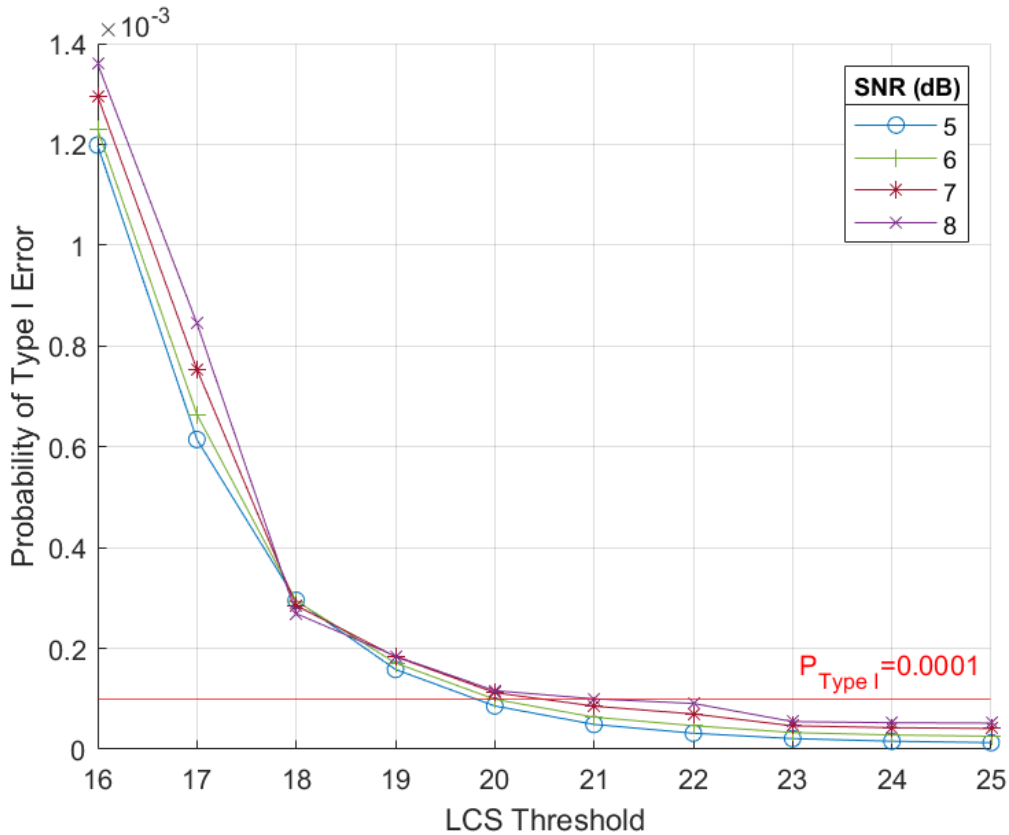


Figure 4.3. $P_{\text{Type I}}$ as a Function of LCS Threshold for Varying SNR Levels.

Examining Figure 4.3, it is determined that the probability of Type I error curve remains relatively flat until a threshold value of approximately 20, at which point the curve begins to increase rapidly. To assist in minimizing the amount of Type I errors throughout the remainder of the simulations, a threshold value of 20 will commonly be used. This threshold value will provide for a reasonable probability of success, while keeping the probability of Type I errors at or below one in ten thousand.

4.2.2 Hamming Distance Trade Space

Similar in concept to the analysis of the LCS threshold analysis, the Hamming distance threshold is now investigated. Figure 4.4, using the same data set as the previous LCS comparison, illustrates the relationship between the Hamming threshold, probability of success, and probability of Type I error for varying SNR levels.

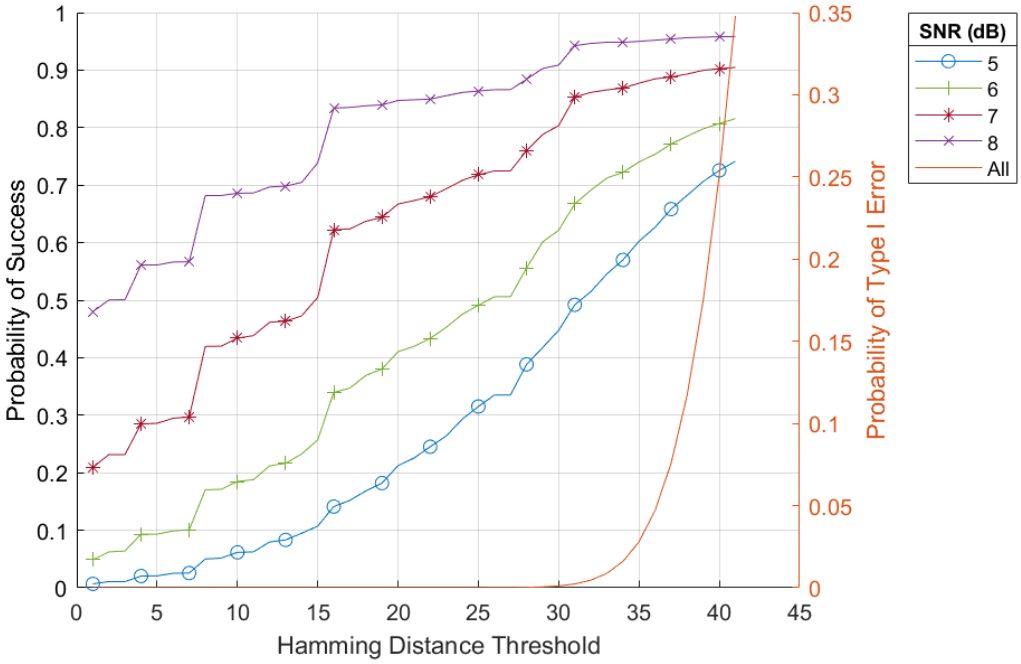


Figure 4.4. $P_{Success}$ and $P_{Type\ I}$ as a Function of Hamming Distance Threshold for Varying SNR Levels.

An error pattern and given syndrome are deemed matches if their Hamming distance is less than or equal to that of the threshold value. The trends in Figure 4.4 are very similar to that of Figure 4.2, namely that the probability of success increases as the threshold is made less restrictive, that the probability of Type I errors are independent of SNR levels, and the Type I errors demonstrate an exponential increase after a certain threshold level. Note that differences in the manner in which the determination method is computed produce trend lines that decrease when the magnitude of the threshold value is increased for LCS, and trend lines that increase when the magnitude of the threshold value is decreased for Hamming distance. An error pattern in a noiseless channel, with no bit errors, would have an LCS of 92 (total number of frozen indices) but a Hamming distance of zero.

Figure 4.5 further investigates the threshold value at which an exponential growth in Type I errors are observed. Consistent with Figure 4.3, it is determined that the probability of Type I errors increases rapidly above the value of one in ten thousand. To maintain this error rate throughout subsequent simulations, the corresponding Hamming distance threshold value of 27 will commonly be used.

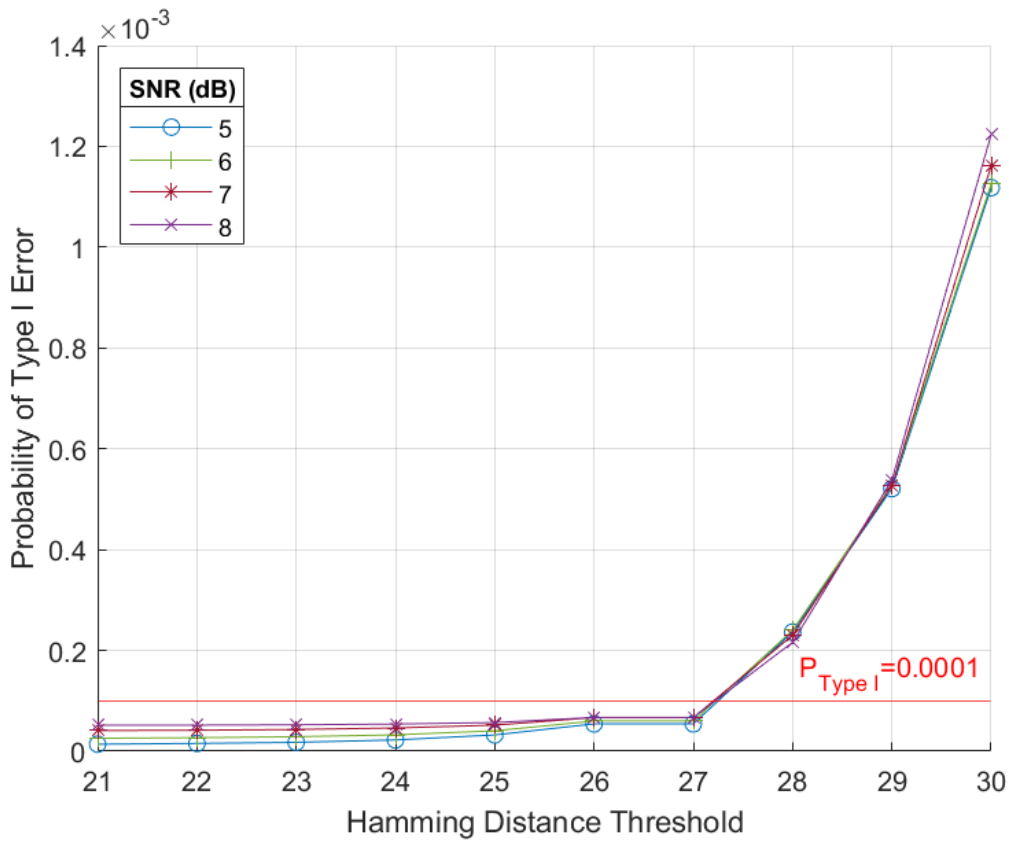


Figure 4.5. $P_{\text{Type I}}$ as a Function of Hamming Distance Threshold for Varying SNR Levels.

Table 4.3 provides a direct comparison between the LCS and Hamming determination methods for each of the SNR levels, using threshold values (20 and 27, respectively) that produce $P_{\text{Type I}} \leq 0.0001$.

Examining Table 4.3, it is apparent that both the LCS and Hamming determination methods perform very similar to each other. The simulations conducted do illustrate that there is a slight advantage to using the Hamming method for lower SNR levels, and the LCS method

Table 4.3. Comparison of LCS and Hamming Method Success Rates for Varying SNR Levels when $P_{\text{Type I}} \leq 0.0001$.

	Probability of Success	
SNR (dB)	LCS	Hamming
5	0.286	0.335
6	0.490	0506
7	0.738	0.725
8	0.880	0.866

for higher SNR levels. This relationship will be further explored in Section 4.3 when P_{Success} for both determination methods are plotted on the same figure.

Despite similarities in the probability of success between the two determination methods, they are not equivalent mapping functions. That is, successful determination of a RNTI using the LCS method does not imply that the Hamming method, used on the same data, would be successful and vice versa. Figure 4.6 illustrates disparities between the methods.

Each subplot of Figure 4.6 was constructed by comparing an error pattern derived from a received transmission that experiences an AWGN channel with an SNR of 7 dB against all 2^{15} possible syndromes using both the LCS and Hamming methods. The $2^{15} - 1$ empty blue circles represent incorrect syndromes and the one filled red circle represents the correct syndrome. Figure 4.6(a), where the correct syndrome is located in Region I, illustrates an instance where both LCS and Hamming methods would have correctly identified it. Figure 4.6(b) represents an instance where the LCS method would have correctly identified the match but the Hamming method would not have. Figure 4.6(c) represents the inverse of 4.6(b), and Figure 4.6(d) represents where neither method would have been able to correctly identify the match. It is assessed that the reason the two methods are non-equivalent mapping functions is the non-linear manner in which additive noise can propagate through the error pattern. In a particular instance the noise may cause bit errors to spread throughout the error pattern (favorable to Hamming method), while another instance may cause errors that are concentrated in only certain portions of the error pattern (favorable to LCS).

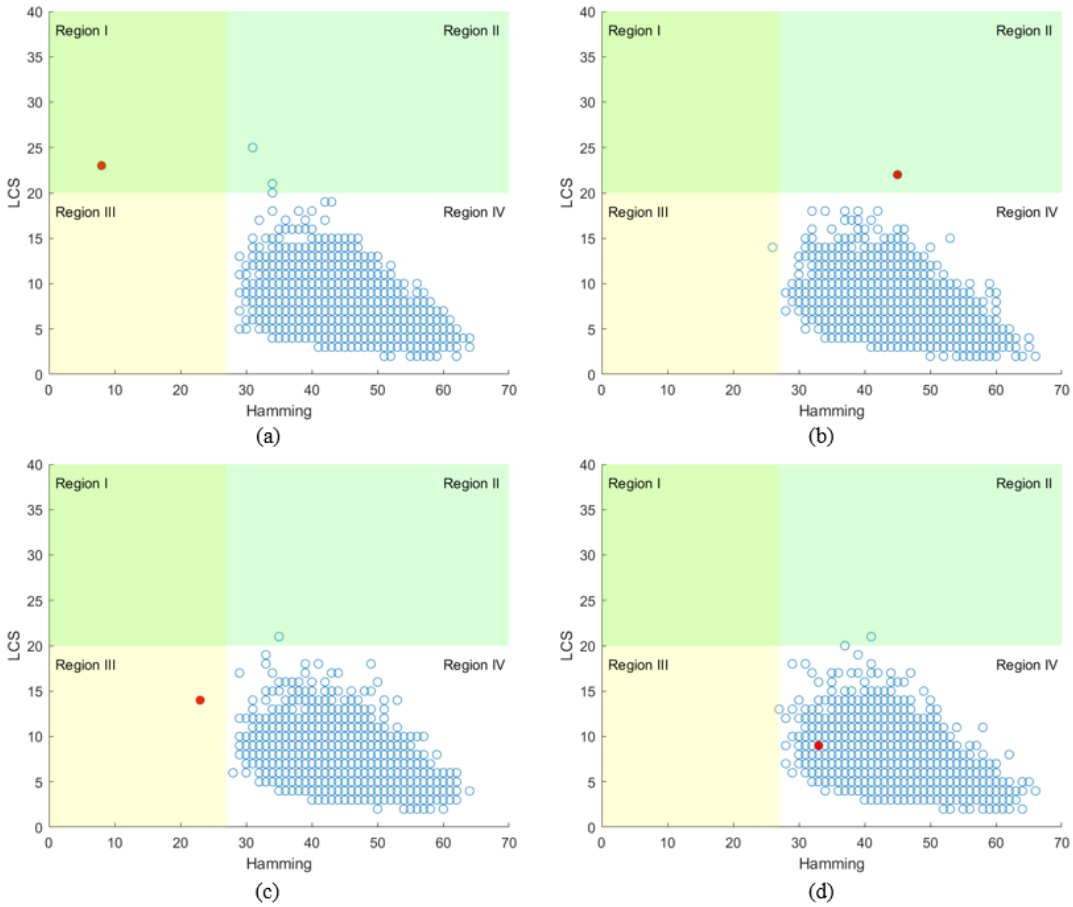


Figure 4.6. LCS and Hamming Determination Methodology Comparison.

4.3 SNR Levels and Receiver Antenna Quantity

The next simulation variable to be examined is that of adjusting the receiver antenna quantity. Section 3.4.2 discussed how by increasing the antenna quantity, the signal processing gain is also increased. Figure 4.7 illustrates the probability of success as a function of SNR for varying antenna quantities for both LCS and Hamming methods.

LCS and Hamming threshold values used (20 and 27, respectively) were those derived from Section 4.2 that corresponds to $P_{\text{Type I}} \leq 0.0001$. For each of the antenna quantities, for relatively low SNR values the $P_{\text{Success}} = 0$ due to the excessive numbers of errors in the received transmission, then begins to increase rapidly until the processing gain effectively

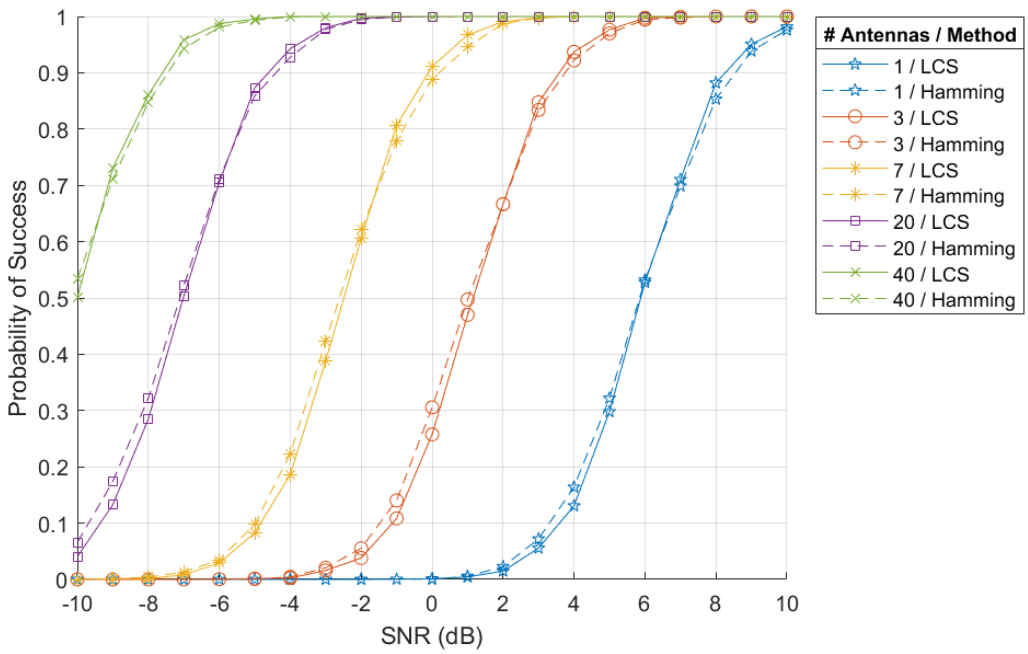


Figure 4.7. $P_{Success}$ as a Function of SNR for Varying Antenna Quantities and Determination Methods. Adapted from [1].

removes all errors from additive noise and $P_{Success} = 1$. Figure 4.7 also visually reinforces the findings of Table 4.4, where it was observed that the Hamming method has a slightly better $P_{Success}$ at relatively low SNR values for each antenna quantity tested, and the LCS method performs better at higher SNR levels. It is hypothesized that the Hamming method works better than the LCS method in low SNR environments due to the difficulty in identifying sufficiently long common substrings when a large number of bit errors have propagated into the error pattern.

4.4 Rate-Matching and Frozen Bit Quantities

The effect that the various rate-matching schemes (i.e., repetition, puncturing, and shortening) have on methodology performance is now examined. Recall Table 3.1, presented again as Table 4.4, which outlines the quantities of usable frozen indices for various rate-matching schemes, to aid in interpreting the results in the following subsections.

Table 4.4. Effects of Rate-Matching Schemes on the Quantity of Usable Error Pattern Frozen Indices.

Scheme	N	E	Total Frozen Indices	Restricted Frozen Indices	Usable Frozen Indices
No Rate-Matching	128	128	92	0	92
Repetition	128	144	92	0	92
Puncturing	128	86	92	42	50
Shortening	128	74	92	54	38

4.4.1 Repetition

Section 3.3.2 established that the $E - N$ repeated indices cannot be constructively used during the modified polar decoding process, and as such are discarded. This is due to the manner in that the scrambling sequence is constructed, where the $E - N$ repeated indices are scrambled differently than the first bits of the codeword, resulting in an inability for the eavesdropper to constructively use the repeated values during rate recovery. For this reason, the performance of a repetition rate-matching scheme is identical to that of a no rate-matching scheme.

4.4.2 Puncturing

To investigate the effect of puncturing on methodology performance, let $E = 86$, similar to previous puncturing examples. From Table 4.4 it is determined that there are 50 usable frozen indices for comparison between the error pattern and given syndrome. Due to the lower number of usable frozen indices, it is reasonable to conclude that, for a given SNR level, the $P_{Success}$ for puncturing will be reduced compared to the no rate-matching scheme. Figure 4.8 compares the performance of a puncturing scheme to that of no rate-matching.

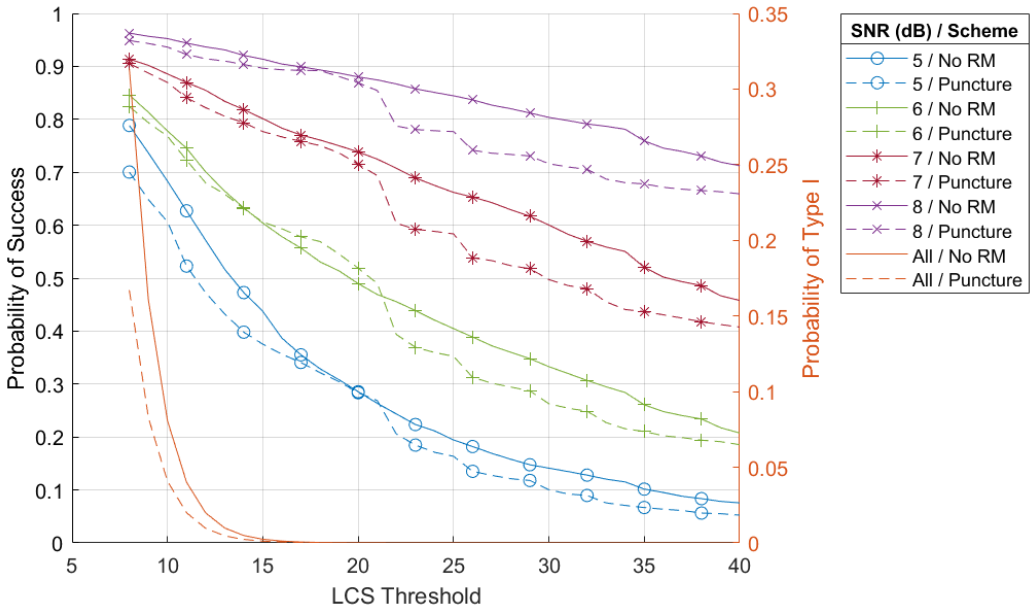


Figure 4.8. $P_{Success}$ and $P_{Type\ I}$ as a Function of LCS Threshold for Varying SNR Levels for Both No Rate-Matching and Puncturing Schemes.

Examining Figure 4.8, it is observed that $P_{Success}$ for puncturing generally follows the trend of the no-rate match scheme, albeit at a slightly reduced level. The $P_{Type\ I}$ values for puncturing are independent of SNR level, similar to no-rate matching. The $P_{Type\ I}$ values do differ between puncturing and no rate-matching, suggesting that $P_{Type\ I}$ is a function of the quantity of usable frozen indices in the error pattern. This is intuitive, in that reducing the quantity of usable frozen indices reduces the likelihood that bit errors are able to propagate into the error pattern is such a manner as to result in a false positive. It is assessed that, under most conditions, utilizing no rate-matching provides a greater $P_{Success}$ than puncturing for similar $P_{Type\ I}$ levels. There are certain conditions, specifically at SNR of 6 dB and threshold values of 16 through 21, where the $P_{Success}$ for puncturing is better than that of no rate-matching. Chapter 5 recommends further research into greater combinations of data message length and rate-matching codeword length combinations to identify other potential conditions that exhibit non-linear gains in performance.

4.4.3 Shortening

For the shortening scheme, let $E = 74$ as in previous examples, resulting in 38 usable error pattern frozen indices per Table 4.4. It is expected that shortening will produce similar results to that of puncturing, but with an even greater degradation in performance due to the greater reduction in usable frozen indices compared to the no rate-matching scheme. Figure 4.9 illustrates the results of the simulation.

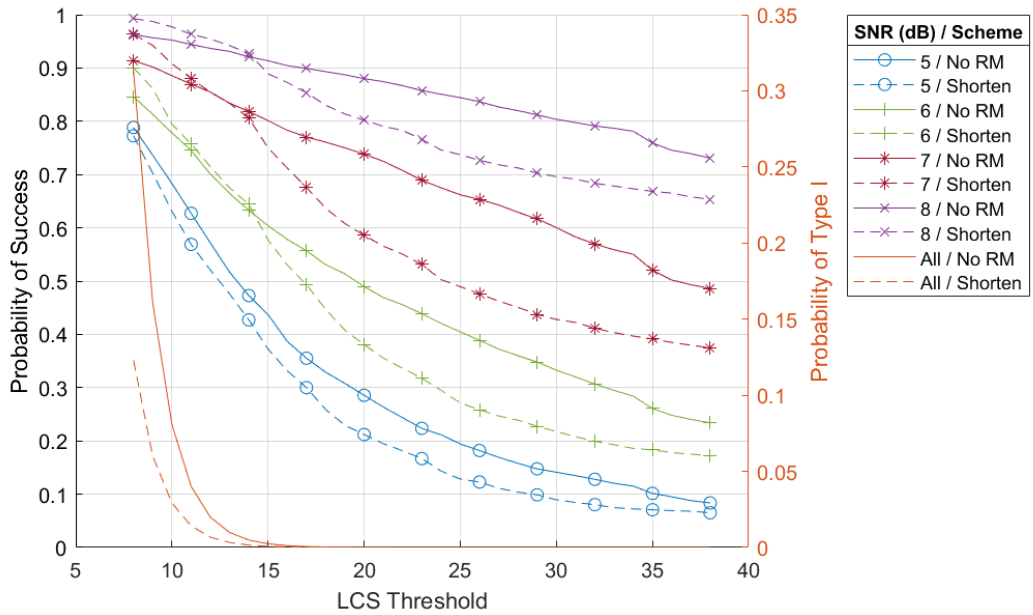


Figure 4.9. $P_{Success}$ and $P_{Type\ I}$ as a Function of LCS Threshold for Varying SNR Levels for Both No Rate-Matching and Shortening Schemes.

At the majority of threshold values examined, shortening exhibits reduced $P_{Success}$ compared to no rate-matching for a given SNR level. At the lower threshold values, beginning at 14 and extending down to the minimum tested value of 8, for some SNR levels the shortening scheme has a greater $P_{Success}$ than that of no rate-matching. Similar to the non-intuitive trends observed from the puncturing example, Chapter 5 recommends a greater examination of possible data message lengths and rate-matching codeword lengths to better identify combinations that produce non-linear performance gains. The shortening $P_{Type\ I}$ values are also observed to be reduced for a given LCS threshold value compared to those of the no rate-matching scheme, further reinforcing that $P_{Type\ I}$ is a function of the quantity of usable frozen indices in the error pattern.

4.4.4 Summary of Rate-Matching Schemes on Methodology Performance

Table 4.5 summarizes the $P_{Success}$ for the no rate-matching, repetition, puncturing, and shortening schemes.

Table 4.5. Comparison of Rate-Matching Scheme Success Rates for Varying SNR Levels when $P_{\text{Type I}} \leq 0.0001$.

	Rate-Matching Scheme $P_{Success}$			
SNR (dB)	None	Repetition	Puncturing	Shortening
5	0.286	0.286	0.284	0.259
6	0.490	0.490	0.519	0.447
7	0.738	0.738	0.715	0.637
8	0.880	0.880	0.867	0.830

The only value in Table 4.5 that does not follow the expected trends is for puncturing when SNR = 6. In this instance, despite the puncturing scheme having fewer usable frozen indices to compare between the error pattern syndromes than no rate-matching, a higher $P_{Success}$ was observed. To identify whether this is statistical anomaly or indicative of codeword length conditions that the methodology performs abnormally well at, Chapter 5 in part recommends a more comprehensive analysis of possible message and rate-matching codeword lengths.

4.5 Type II Errors

Recall from Section 4.1 the assumption that the polar decoder was able to accurately correct any and all errors present once the correct syndrome was identified. Under certain conditions, for the successive-cancellation decoder that was used, this is not always true. Figure 4.10 illustrates a more detailed process flow than what was earlier presented in Figure 4.1 at the beginning of the chapter.

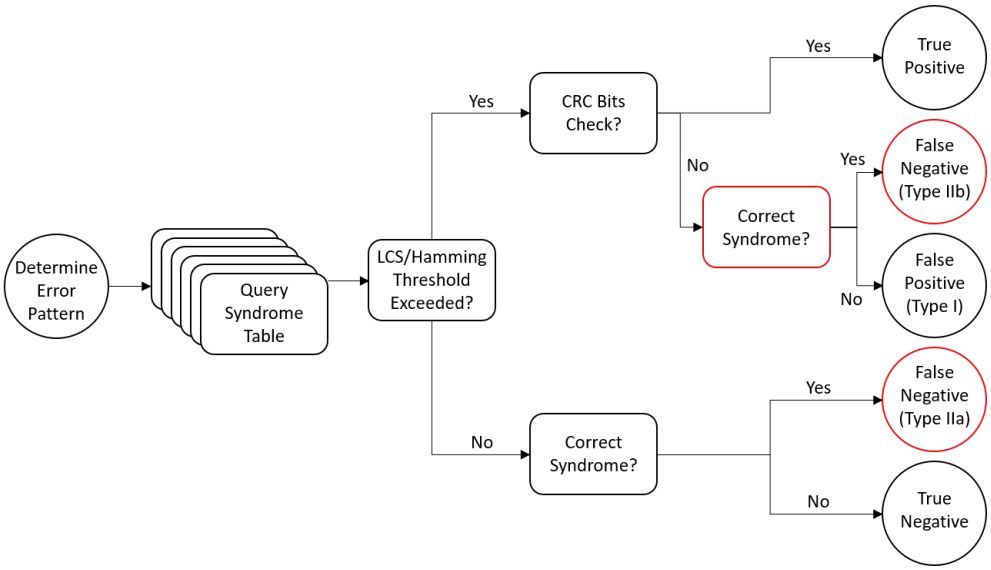


Figure 4.10. Detailed Simulation Metric Process Flow (Changes from Figure 4.1 Highlighted in Red.

Type II errors are now differentiated into Type IIa and IIb errors. A Type IIa error is equivalent to the initial definition of a Type II error, namely the probability that the threshold value will not be exceeded when the error pattern is compared to the syndrome that corresponds to the correct RNTI. A Type IIb error is the probability that the threshold value will be exceeded when the error pattern is compared to the syndrome that corresponds to the correct RNTI, but the RNTI is determined to be incorrect regardless due to uncorrectable errors in the decoding process that cause the CRC check bits to fail.

In order to investigate the occurrence of Type IIb errors, a simulation was conducted where an error pattern was compared to the syndrome that corresponds to the correct RNTI under a specific set of conditions for 2500 independent trials. The conditions used were no rate-matching scheme, a single receiving antenna, an LCS threshold value of eight, and the SNR level varied between 1 dB and 8 dB in 1 dB increments. The threshold value of eight, along with the relatively low SNR levels for a single antenna, were selected so that the LCS threshold value would be exceeded in a large number of trials despite the presence of a high quantity of bit errors. These are not generally favorable operating parameters, as a threshold value of eight normally equates to $P_{\text{Type I}} = 0.3135$. Lastly, since the objective of this simulation is to examine the occurrence of Type IIb errors, comparing the error pattern

against only the correct syndrome will preclude the occurrence of Type I errors or true negatives. Table 4.6 tabulates the results from the simulation.

Table 4.6. Summary of Type II Error Investigation.

	SNR (dB)							
Metric	1	2	3	4	5	6	7	8
$P_{\text{True Positive}}$	0.2720	0.3884	0.4812	0.6304	0.7536	0.8632	0.9276	0.9588
$P_{\text{Type IIb}}$	0.0188	0.0040	0	0.0004	0	0	0	0
$P_{\text{Type IIa}}$	0.7092	0.6076	0.5188	0.3692	0.2464	0.1368	0.0724	0.0412

It is observed that a small number of Type IIb errors do occur at the lower SNR levels. Again, this is a byproduct of the low threshold value selected and the low SNR levels present, which combined to allow for a large number of errors to propagate into the error pattern. It is assessed that by using a different polar decoder (i.e., SCL or SCS) an eavesdropper may be able to further reduce the already small probability of Type IIb errors, depending upon the operating environment. Selection of a different polar decoder may help to ensure that the performance of the methodology is not limited by the error-correcting capabilities of the polar decoder utilized.

Chapter 4 established simulation variables and metrics that are of interest, and then systematically examined how each of the identified variables impacted associated metrics. Error pattern thresholds were examined, in order to understand the trade space between the proba-bility of successfully identifying the correct RNTI and the probability of encountering false positive results. The impact of varying the number of receiving antennas in different SNR environments was examined, along with the impact that implementing one of the various rate-matching schemes has on methodology performance. Finally, the degree to which the polar decoding technique selected is a limiting factor in overall performance was examined.

CHAPTER 5:
Summary and Recommendations

Chapter 5 serves to provide a brief summary of the previous chapters, and to also provide recommended areas for related future research work.

5.1 Summary

This thesis has presented a methodology where a passive eavesdropper of a network can, through using a modified polar decoding process, determine the RNTI associated with a given DCI message. The modified polar decoding process leverages the frozen bits contained within the transmitted codeword, not to aid in error correction as is there intended purpose, but as a mechanism to develop an error pattern. The determined error pattern allows for interpreting the scrambling sequence that was applied to the transmitted codeword as a type of non-random noise. Through comparison of the error pattern to entries in a syndrome table, identification of the applied scrambling sequence, and by extension the associated RNTI, can be achieved.

Two techniques for comparing the error pattern to the syndrome table were examined through simulation. These techniques were identifying the longest common substring (LCS) and computation of the Hamming distance. Both techniques produced similar $P_{Success}$ and $P_{Type\ I}$ metrics, with the Hamming distance method having a slightly higher $P_{Success}$ at low SNR levels and the LCS method having slightly higher $P_{Success}$ at high SNR levels. Furthermore, the $P_{Success}$ was demonstrated to generally increase linearly as the determination threshold was made less restrictive, while $P_{Type\ I}$ errors increased exponentially. This relationship between the two metrics results is beneficial in identifying the ideal methodology parameters that produce the greatest $P_{Success}$ while limiting $P_{Type\ I}$.

5.2 Recommend Related Future Research Areas

Several ideas for future research that builds upon the concepts presented in this paper are as follows.

5.2.1 Direct RNTI Derivation From Error Pattern

Recall Figure 3.5, which illustrates the process flow for the presented methodology in the presence of additive noise, presented here again as Figure 5.1.

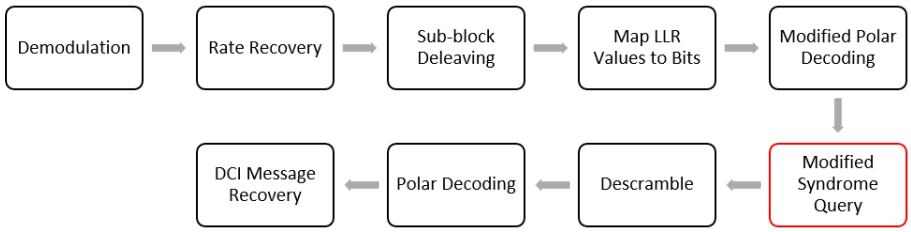


Figure 5.1. Methodology Process Flow in the Presence of Additive Noise.

The process step that has the potential to be the most computationally intensive is the Modified Syndrome Query step, highlighted in red. This step involves comparing the error pattern z_0^{N-1} to each syndrome in the set of syndromes $\mathcal{S}$ until a match is found and verified. The size of the set of syndromes is $|\mathcal{S}| = 2^{15}$, meaning that a relatively large number of comparisons may need to be made before a match is identified.

It is hypothesized that, based off of examining the frozen indices in the error pattern, the scrambling sequence that was applied to the codeword can be determined directly without the use of a lookup syndrome table. Further analysis of the manner in which the scrambling sequences, bit errors caused by additive noise, and the polarization matrix interact with each other may yield the necessary transformation equations to no longer have to use a lookup table.

5.2.2 Identification of Optimal Methodology Conditions

Throughout the conduct of the simulations and analysis of the results, it was observed that under certain conditions there were non-linear gains in methodology performance. Figure 4.8 illustrates the results from the simulation that examined how the puncturing rate-matching scheme affected the success rate, presented here again as Figure 5.2.

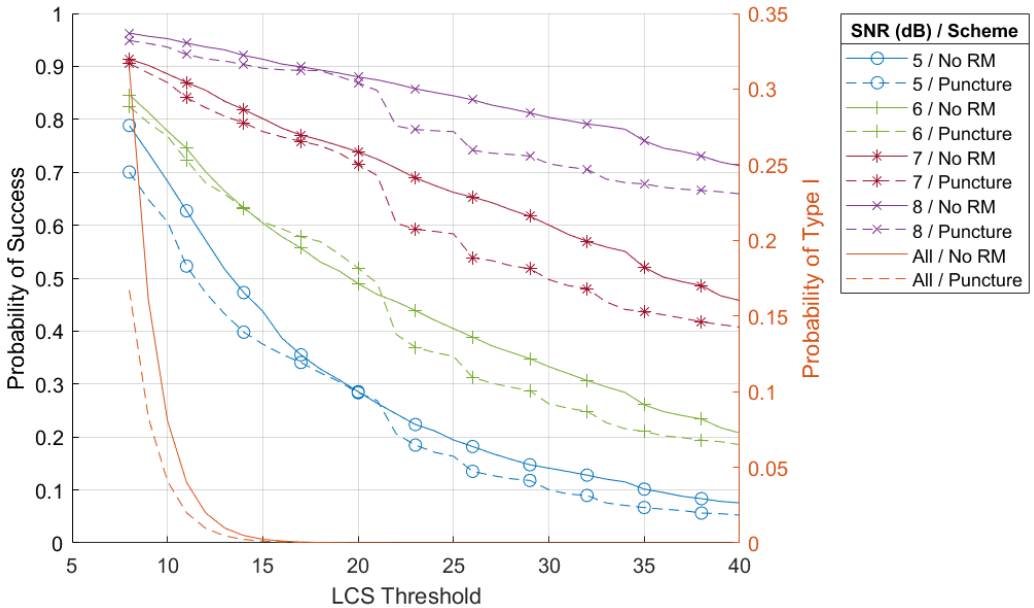


Figure 5.2. $P_{Success}$ and $P_{Type\ I}$ as a Function of LCS Threshold for Varying SNR Levels for Both No Rate-Matching and Puncturing Schemes.

Between the LCS threshold values of 20 and 23 the $P_{Success}$ for the puncturing scheme increased at a significantly higher rate than that of no rate-matching, either nearly equaling the performance of no rate-matching or even exceeding it in the case of SNR = 6 dB. Further investigation into combinations of data message lengths, polar encoding lengths, and rate-matching lengths may better identify those conditions where the methodology yields more desirable results.

5.2.3 Application to Other 5G-NR Physical Channels.

This thesis focused on identifying information within a single 5G-NR physical channel, namely the PDCCH. It is hypothesized that the presented methodology is applicable, either in its current format or with minimal changes, to both the physical uplink control channel (PUCCH) and PBCH. Additionally, using the presented methodology as a basis, it is recommended that research be conducted on applicability to the physical channels that use low density parity check (LDPC) coding. These physical channels include the PDSCH and physical uplink shared channel (PUSCH).

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX: Supplemental MATLAB Code

As a supplement to the discussion provided in this thesis, four MATLAB live script files (.mlx) are available for download from the Naval Postgraduate School Dudley Knox Library at https://library.nps.edu. The files available for download are listed below.

methodology.mlx — This file provides a complete simulation of the eavesdropper methodology. Both the LCS and Hamming distance determination methods can be selected, along with variable threshold levels. This file requires each of the following three functions in order to operate.

encodePDCCH.mlx — This function takes the data payload and PDCCH parameters as inputs and outputs complex-value modulated symbols for transmission.

crcIndexGenerator.mlx — This function takes the data payload length as an input and outputs the relationship between the message bits and the CRC parity bits.

syndromeTableGenerator.mlx — This function generates the syndrome tables for a given set of conditions so that the eavesdropper can compare the error pattern against them.

THIS PAGE INTENTIONALLY LEFT BLANK

List of References

[1] B. E. Gardner and J. D. Roth, "An efficient methodology to de-anonymize the 5G-new radio physical downlink control channel," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 2917–2921.

[2] GSMA. The Mobile Economy. [Online]. Available: https://gsma.com/r/mobileeconomy

[3] Merriam-Webster. Privacy. [Online]. Available: https://www.merriam-webster.com/dictionary/privacy

[4] Merriam-Webster. Anonymous. [Online]. Available: https://www.merriam-webster.com/dictionary/anonymous

[5] H. Zang and J. Bolot, "Anonymization of location data does not work: A large-scale measurement study," in *MobiCom*, 2011, pp. 145–156.

[6] Y. De Mulder, G. Danezis, L. Batina, and B. Preneel, "Identification via location-profiling in GSM networks," in *Workshop on Privacy in the Electronic Society*, 2008, pp. 23–32.

[7] C. Y. T. Ma, D. K. Y. Yau, N. K. Yip, and N. S. V. Rao, "Privacy vulnerability of published anonymous mobility traces," *IEEE/ACM Trans. Netw.*, vol. 21, pp. 720–733, Jun 2013.

[8] R. Shokri, G. Theodorakopoulos, J.-Y. L. Boudec, and J.-P. Hubaux, "Quantifying location privacy," in *IEEE Symp. Security Privacy*, 2011, pp. 247–262.

[9] C. Drane, M. Macnaughtan, and C. Scott, "Positioning GSM telephones," *IEEE Commun. Mag.*, vol. 36, no. 4, pp. 46–54, 1998.

[10] M. Raitoharju, S. Ali-Löytty, and L. Wirola, "Estimation of base station position using timing advance measurements," in *Proc. Intl. Conf. Graphic Image Process.*, 2011.

[11] G. Yost and S. Panchapakesan, "Improvement in estimation of time of arrival (TOA) from timing advance (TA)," in *Proc. IEEE Intl. Conf. Universal Personal Commun.*, 1998, pp. 1367–1372.

[12] J. D. Roth, M. Tummala, J. C. McEachen, and J. W. Scrofani, "On location privacy in LTE networks," *IEEE Trans. Inf. Forensics Security*, vol. 12, pp. 1358–1368, Jun 2017.

[13] T. Hiltunen, J. Turkka, R. Mondal, and T. Ristaniemi, "Performance evaluation of LTE radio fingerprint positioning with timing advancing," in *Proc. Intl. Conf. Info. Commun. Sig. Process.*, 2015.

[14] T.-H. Ngo and Y. Kim, "Using timing advance to support proximity discovery in network-assisted D2D communication," in *Proc. Intl. Conf. Ubiquit. Future Net.*, 2015, pp. 926–928.

[15] T. Wigren, "Fingerprinting localisation using round trip time and timing advance," *IET Commun.*, vol. 6, pp. 419–427, 2012.

[16] J. Jin, Z.-J. Qui, and B. Ran, "Intelligent route-based speed estimation using timing advance," in *Proc. IEEE Intell. Transportation Syst. Conf.*, 2006, pp. 194–197.

[17] 3GPP TS 38.211, release 15, (v15.6.0), "Physical channels and modulation," June 2019.

[18] 3GPP TS 38.212, release 15, (v15.6.0), "Multiplexing and channel coding," June 2019.

[19] 3GPP TS 38.321, release 15, (v15.5.0), "Medium Access Control (MAC) protocol specification," Mar. 2019.

[20] E. Arikan, "Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels," *IEEE Trans. Inf. Theory*, vol. 55, pp. 3051–3073, July 2009.

[21] I. Tal and A. Vardy, "List decoding of polar codes," *IEEE Trans. Inf. Theory*, vol. 61, pp. 2213–2226, May 2015.

[22] K. Niu and K. Chen, "CRC-aided decoding of polar codes," *IEEE Commun. Lett.*, vol. 16, pp. 1668–1671, October 2012.

[23] L. Xiang, Z. B. K. Egilmez, R. G. Maunder, and L. Hanzo, "CRC-aided logarithmic stack decoding of polar codes for ultra reliable low latency communication in 3GPP new radio," *IEEE Access*, vol. 7, 2019.

Initial Distribution List

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California