ON CYCLIC POLAR CODES AND THE BURST ERASURE PERFORMANCE OF SPATIALLY-COUPLED LDPC CODES

A Thesis

by

NARAYANAN RENGASWAMY

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Chair of Committee,	Henry D. Pfister
Co-Chair of Committee,	Krishna R. Narayanan
Committee Members,	Gregory H. Huff
	Anxiao Jiang
Head of Department,	Miroslav M. Begovic

December 2015

Major Subject: Electrical Engineering

ABSTRACT

In this thesis, we produce our work on two of the state-of-the-art techniques in modern coding theory: polar codes and spatially-coupled LDPC codes.

Polar codes were introduced in 2009 and proven to achieve the symmetric capacity of any binary-input discrete memoryless channel under low-complexity successive cancellation decoding. Since then, finite length (non-asymptotic) performance has been the primary concern with respect to polar codes. In this work, we construct cyclic polar codes based on a mixed-radix Cooley-Tukey decomposition of the Galois field Fourier transform. The main results are: we can, for the first time, construct, encode and decode polar codes that are cyclic, with their blocklength being arbitrary; for a given target block erasure rate, we can achieve significantly higher code rates on the erasure channel than the original polar codes, at comparable blocklengths; on the symmetric channel with only errors, we can perform much better than equivalent rate Reed-Solomon codes with the same blocklength, by using soft-decision decoding; and, since the codes are subcodes of higher rate RS codes, a RS decoder can be used if suboptimal performance suffices for the application as a trade-off for higher decoding speed. The programs developed for this work can be accessed at `https://github.com/nrenga/cyclic_polar`.

In 2010, it was shown that spatially-coupled low-density parity-check (LDPC) codes approach the capacity of binary memoryless channels, asymptotically, with belief-propagation (BP) decoding. In our work, we are interested in the finite length average performance of randomly coupled LDPC ensembles on binary erasure channels with memory. The significant contributions of this work are: tight lower bounds for the block erasure probability (P_B) under various scenarios for the burst pattern;

bounds focused on practical scenarios where a burst affects exactly one of the coupled codes; expected error floor for the bit erasure probability (P_b) on the binary erasure channel; and, characterization of the performance of random regular ensembles, on erasure channels, with a single vector describing distinct types of size-2 stopping sets. All these results are verified using Monte-Carlo simulations. Further, we show that increasing variable node degree combined with expurgation can improve P_B by several orders of magnitude in the number of bits per coupled code.

I dedicate this thesis to my wonderful parents, *Rengaswamy Purushothaman* and *Sudha Rengasami*, for their unconditional love and support.

ACKNOWLEDGEMENTS

I thank all my committee members for taking the time and effort to conduct my defense. Specifically, I extend my heartfelt thanks to Dr. Anxiao Jiang for humbly adjusting his schedule to fit the common time slot when my advisory committee can meet to conduct the defense.

My advisor, Dr. Henry Pfister, has been instrumental in motivating me throughout my thesis. Despite the amazing, and complex, associations between statistical physics and coding theory that he has developed in his mind, he always communicated clearly, precisely and promptly while resolving the doubts and concerns of the novice in me. I extend my gratitude to him for encouraging me to submit a paper to the prestigious International Symposium on Information Theory (ISIT), of the Institute of Electrical and Electronics Engineers (IEEE), and to present the work in the conference myself. The experience was truly international and exposed me to a lot of different researchers and research problems in the broad area of information theory. I also deeply appreciate his effort to recommend me for an internship in the optical communications group of Alcatel-Lucent Bell Labs at Stuttgart, Germany. The exposure was immensely useful, both professionally and personally, without which the work on spatially-coupled LDPC codes would not have been possible. Due to his encouragement, I have been able to work on two of the state-of-the-art techniques in modern coding theory: polar codes and spatially-coupled LDPC codes.

I would like to thank Dr. Krishna Narayanan for useful discussions during my thesis work and for providing important professional advise that has enhanced my graduate experience. In addition to that, I thank him for his financial support for my conference trip to Hong Kong.

I appreciate the efforts of Dr. Gregory Huff and Dr. Jean-François Chamberland for designing a projects-based-learning course for both undergraduate and graduate students. The course is a great initiative in providing hands-on exposure on building systems and I feel that such a course bridges the gap between theory and practice, and also between college and graduate school. I sincerely thank them for appreciating my efforts during the course, having me as a research assistant in their lab and later recommending me as a teaching assistant for assisting undergraduate students in their senior design projects. Especially, I thank Dr. Chamberland for mentoring and guiding me many a time. I would also like to thank Dr. Samuel Villareal for recruiting me as a teaching assistant despite his preference for doctoral candidates. He always keeps himself professional, lively and motivational with his assertive, confident nature and his wonderful sense of humor.

I thank all my professors for their courses and the productive discussions both in class and in person. Especially, the graduate level courses on Wireless Communications, by Prof. Scott Miller, and on Computer and Communication Networks, by Prof. Narasimha Reddy, were instrumental in understanding the field of communications and networking. The homework assignments were challenging and complemented the class lectures in providing a complete, informative experience. They have helped me immensely in fitting my research into the big picture.

The work on SC-LDPC codes was done under the supervision of Dr. Laurent Schmalen and Dr. Vahid Aref during the summer of 2015, while I was a research intern in Alcatel-Lucent Bell Labs, Stuttgart, Germany funded by the DAAD-RisePro scholarship program. The internship was so productive due to the professional guidance of my supervisors. They were well versed with the literature on spatially-coupled LDPC codes and hence were able to keep me well focused during the course of my internship. My brain storming sessions with them have proven to be instrumental in getting good results in the short period of three months. I extend my sincere thanks to both of them for providing good exposure on industrial research and development.

My thesis work combined with the internship experience, the conference trip, the exposure in Dr. Huff's lab and the ongoing teaching assistantship have, as a whole, given me a lot of confidence in performing theoretical research, presenting in a professional setting, building good prototypes and helping students with their projects. I view this as an important mixture of exposures towards building my career in academia.

A graduate experience without great friends is incomplete. I have been able to make great friends, involve in productive discussions and build wonderful memories throughout my graduate school experience. Specifically, I have had my best times with Nagaraj Janakiraman, Adithyaram Narayan and Kalluri Raja Sreeram, who have been great roommates for more than two years, and with Karthik Kalyanaraman, Balakumar Jayaraman, Karthick Sudhan, Srinivasa Varadhan, Nani Anudeep, Siddharth Agarwal, Sangeeta Panigrahy and Kartic Bhargav who always kept the fun alive. I thank Santosh Emmadi, Avinash Vem and Santhosh Kumar for my great times in our research group. Santhosh Kumar also shared a lot of his graduate school wisdom during our trip to the conference. I found wonderful friends in Michael Bass, Brian Bass, Dipanjan Saha, Shuli Li, Yayun Lau, Zachary Partal and Desmond Uzor during our project for the course with Dr. Huff and Dr. Chamberland. The internship experience helped me get to know fellow interns Lei Zhang, Foad Sohrabi and Nazanin Rastegardoost with whom I have been able to strike good friendship through our chats and our trips in Europe. I have had some productive discussions with Lei Zhang on spatially-coupled LDPC codes and staircase codes. I would also like to thank my fellow teaching assistants Neal Hollingsworth, Geha Chadi, Sean Goldberger, Ahmad Bashaireh, Ahmed Morsy and Mandel Oats for making the job a fun and informative experience. The enthusiasm of the undergraduate students, with whom I have got a chance to interact, has always been refreshing and motivating to learn more.

I would like to stress that my graduate school experience would not have been so smooth without the support of our department's academic advisors Tammy Carda and Jeanie Marshall, our business coordinators Sheryl Mallett and Anni Brunker, and the International Student Services (ISS) at Texas A&M. I have always admired all of their responsibility and enthusiasm at work and, specifically, Anni's ways of keeping herself refreshed and positive. The ISS is offering a commendable service in addressing the concerns and needs of all international students here. Complementing the work of ISS, the student organizations on campus are ensuring a personally comfortable and memorable experience for all students. Specifically, I thank the Indian Graduate Students Association (IGSA) and India Association (IA) for having helped in making the transition from India to the United States very smooth. I also appreciate the Big Event and Aggie Replant communities for doing laudable work for the society and I thank them for giving me a chance to be part of that Aggie spirit.

I deeply thank the university and department committees for having admitted me into the Master of Science (M.S.) program of the Department of Electrical and Computer Engineering and providing a great international exposure early in my career.

Ultimately, I thank my parents, my family and the Almighty for their blessings and continuous support.

NOMENCLATURE

APP	A Posteriori Probability
AWGN	Additive White Gaussian Noise
BEC	Binary Erasure Channel
BMS	Binary Memoryless Symmetric
BP	Belief Propagation
B-DMC	Binary Input DMC
CN	Check Node
DMC	Discrete Memoryless Channel
FFT	Fast Fourier Transform
GF	Galois Field
LDPC	Low-Density Parity-Check
MAP	Maximum A Posteriori
QEC	q-ary Erasure Channel
QSC	q-ary Symmetric Channel
QSCE	q-ary Symmetric Channel with Erasures
RBC	Random Burst Channel
RS	Reed-Solomon
SC-LDPC	Spatially-Coupled LDPC
SP	Spatial Position
SPBC	Single Position Burst Channel
VN	Variable Node
w.l.o.g	without loss of generality

TABLE OF CONTENTS

Page

LIST OF FIGURES

FIGURE Page

LIST OF TABLES

TABLE Page

1. INTRODUCTION

In this thesis, we consider two capacity achieving codes that form the state-of-the-art techniques in coding theory today: *Polar codes* and *Spatially-Coupled Low-Density Parity-Check codes.* Both of these codes are very recent inventions given the six-and-a-half decade history of information and coding theory. To appreciate the significance of these codes, we need to take a brief look into the motivation behind modern communication systems and get an idea of their current maturity. Specifically, it is imperative to glimpse through the fundamentals of coding theory in order to understand the notion of capacity and hence capacity achieving codes. This section attempts to provide an overview that just meets this purpose. The last subsection outlines the rest of this thesis.

1.1 From Telephone Systems to Modern Communication

Telephone systems became prevalent in the United States since the turn of the twentieth century. The Bell Systems company found it increasingly difficult to maintain good service to all customers given their rapid expansion. By the 1930s, manual operation of the telephone systems were starting to get replaced by switching circuits for the need of speed. The introduction of these switching circuits made systems more complex and it became imperative to understand their behavior in theory. This was the primary motivation for Claude E. Shannon, a young electrical engineer working as a research assistant in the Massachusetts Institute of Technology, Cambridge. In his own words, “Examples of these circuits occur in automatic telephone exchanges, industrial motor-control equipment, and in almost any circuits designed to perform complex operations automatically” [26]. Shannon is arguably the first person to introduce boolean logic in the representation of switching circuits, which he did in 1937 in his remarkable Master's thesis titled *A Symbolic Analysis of Relay and Switching Circuits*. Ironically, the thesis was unpublished though a paper abstracted from it was published in the Transactions of the American Institute for Electrical Engineers in 1938 [26]. This is still regarded as one of the most important theses ever written.

He introduced the notion of 1 and 0 to denote the open and closed state of a circuit switch, respectively. In his work, he focused mainly on the problem of network synthesis – how do we synthesize a network that incorporates certain desired characteristics? He showed that several well-known theorems in impedance networks have roughly analogous versions in relay circuits. The simple, yet powerful, approach was to represent relays and switches as mathematical variables and construct systems of equations that describe their interactions and behavior. In this way, it was possible to first write down the desired characteristics in precise mathematical language, solve the set of equations to get the optimal solution using the necessary calculus, and then implement the solution with actual relays and circuits.

Later, Shannon desired to characterize the transport of information through the telephonic systems. Precisely, he wanted to know how to design telephone systems that could carry maximum amount of information and also account for distortion in the lines. To answer that question, he needed to quantify the abstract notion of *information* conveyed through any form – text, sound or image. This inspired him to invent and formulate the profound mathematics called *Information Theory*.

1.2 The Idea of Information Theory

Shannon's view of information in a message was as follows: a message that brought more surprise to the receiver contained more information and a message that was more predictable contained very less information. Concisely, he viewed information as the amount of uncertainty that it resolves upon reception at the receiver. This notion is indeed very intuitive and he defines *entropy* as the amount of information or randomness that a message contained by itself. The more profound idea was to represent any kind of data just by a sequence of 1s and 0s. Probably, it made sense to him to represent data just by states of a set of switches given his background and work on relays and switches. As a New York Times article pointed out [11], he proposed that the information contained in a message had nothing to do with the content but only with the number of 1s and 0s necessary to represent it. All character sets developed for computers rely on this underlying notion of message representation. Combining the idea of the digital notation of messages, his notion of information and his desire to find the maximum amount of information that can be transmitted over a distortion-prone telephone line, he laid down a seminal mathematical treatise titled *A Mathematical Theory of Communication* [27] which was later published as a book, co-authored by Warren Weaver, with the title *The Mathematical Theory of Communication*. The small but significant change in the title was to emphasize the generality of the work.

1.3 The Communication System Model

Shannon modeled a communication system as follows: an information source generates data, then it is transmitted appropriately as a signal over a communication channel that corrupts the signal with noise, and eventually the corrupted signal is received and converted back to the original data format and delivered to the desired destination. The generality of this model is very evident. Also, such a formulation allows us to work on and refine the individual blocks while keeping their interface to the rest of the system unperturbed, i.e. the model has very good modularity. His model of communication systems is simple, elegant and has proven to be very successful having stood the test of time. Even today, research and development in the broad area of communications depends strongly on his model.

A slightly more detailed model than the one in Shannon's paper [27] is given in Fig. 1.1. The source coding block desires to represent the input data in the most concise form but being stringent on the allowance on information loss. This thesis fits into this big picture in the channel encoder-decoder pair of blocks. The function of the channel encoding block is to add calculated redundancy to the source-coded data so that the distortion introduced by the channel can be corrected at the channel decoding block, given the mathematical structure of the added redundancy. The block error probability, P_B, at the receiver is a measure of the chance that the channel decoder will fail to correct the errors introduced by the channel in the codeword. Hence, the goal for coding theory is to construct efficient codes over a particular channel or a class of channels so that, the code introduces minimal redundancy while retaining P_B at an acceptable low level.

1.4 Fundamentals of Coding Theory

1.4.1 What is a Code?

A code is an encoding-decoding pair that explicitly states the details of the two complementary blocks. A channel encoder would typically encode K information bits, that it receives as input from the source encoder, into N code bits by adding $(N-K)$ bits of calculated redundancy. Every codeword is a block of encoded bits of data and its length, called *blocklength*, is N. Hence, for every N bits of coded data, there are K bits of information conveyed and the *rate*, R, of the code is given as

$$R = \frac{K}{N}. \tag{1.1}$$

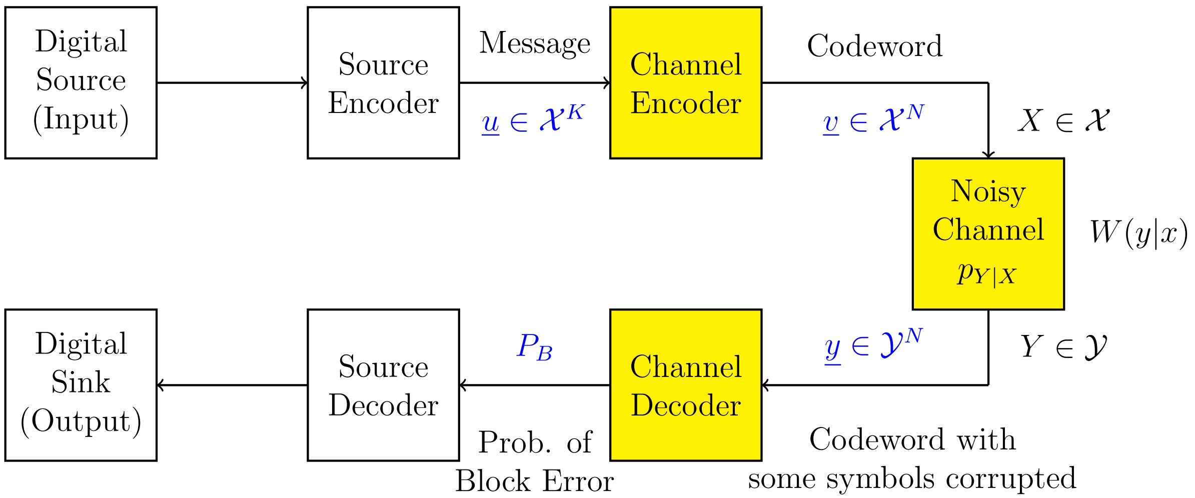


Figure 1.1: A communication system highlighting the main blocks from a coding theoretic perspective. It is slightly more detailed than the model considered by Shannon in [27] in the sense that the source and channel coding blocks are shown for emphasis.

This is the rate of coded transmission into the channel. Different codes have different ways of calculating the redundancy and different ways of decoding the received (corrupted) word. The choice for the encoder and decoder ultimately decides the performance of the channel coding block in the communication system, measured by the *probability of decoding error* P_B. This quantity is also termed the *block error probability* since at least one bit in the codeword block is in error after decoding.

1.4.2 Capacity Achieving Codes

To set up the goal of coding theory, mathematically, we need to revisit the fundamentals of information theory. *Entropy* and *Mutual Information* are two fundamental quantities defined by Shannon to address the information inherently present in a message and the amount of shared, or mutual, information between two correlated messages, respectively. For this reason entropy is also called *self-information.*

Consider a source emitting messages from a binary alphabet $\mathcal{X} = \{0, 1\}$. The distribution on the alphabet is arbitrary and source-dependent. Let X be a random variable that denotes a message from this source. Then, the entropy of the source is given by

$$H(X) \triangleq \mathbb{E}_X\left[\log \frac{1}{p(x)}\right] = -\sum_{x\in\mathcal{X}} p(x)\,\log p(x) \quad \text{bits} \tag{1.2}$$

where, $p(x)$ is the probability that X takes the value x, and the logarithm is over base-2. Unless specified otherwise, all logarithms in this thesis will be base 2. However, the notion of entropy is general and can be extended to non-binary alphabets too.

Now, let the message X be sent over a distortion-prone channel and received as Y which might also belong to the same binary alphabet or to a different one depending on the channel model. For simplicity, let us assume that the output alphabet $\mathcal{Y}$ is also binary. Then, the mutual information between X and Y is defined as

$$I(X;Y) \triangleq \mathbb{E}_{(X,Y)}\left[\log \frac{p(x,y)}{p(x)p(y)}\right] = -\sum_{y\in\mathcal{Y}}\sum_{x\in\mathcal{X}} p(x,y)\,\log \frac{p(x)p(y)}{p(x,y)} \quad \text{bits/channel use.} \tag{1.3}$$

Ideally, if 1 bit of information is sent over one use of the channel, then 1 bit of information must be received. But, since the channel is distortion-prone, the mutual information is less than 1 bit/channel use. Hence, the *capacity* of a channel, C, is defined as the maximum amount of mutual information over all input distributions on $\mathcal{X}$.

$$C \triangleq \max_{p(X)}\ I(X;Y) \quad \text{bits/channel use.} \tag{1.4}$$

This is the maximum rate, R, at which information can be transmitted and recovered reliably through an appropriate coding scheme. The *symmetric capacity* of a channel is the maximum amount of information that can be transmitted reliably over one use of the channel, subject to using the input values 1 and 0 with equal frequency.

In other words, given a noisy channel with capacity C, for every transmission rate $R < C$, there exists a coding scheme which guarantees that information can be reliably transmitted over that channel at the rate R and that the maximum probability of decoding error at the receiver can be made arbitrarily small.

Shannon proved this in his famous noisy channel coding theorem using random coding arguments. But, to implement a practical communication system we need to design specific codes that can achieve this limit. Codes that achieve this limit for a given class of channels are called *capacity-achieving codes.* The following statement formalizes the notion of capacity-achieving codes:

Given a channel with capacity C, there exists a sequence of codes, indexed by n, with rates R_n such that

$$\lim_{n\to\infty} R_n = C \quad \text{with} \quad P_{B_n}^{\max} \to 0. \tag{1.5}$$

Designing such codes with deterministic constructions has been the pursuit of coding theorists over the past six decades.

1.5 Outline of the Thesis

Polar codes are the first codes to have been explicitly shown to achieve the capacity of arbitrary symmetric binary-input discrete memoryless channels (B-DMCs) under low-complexity successive cancellation decoding. This breakthrough was made by Arikan in 2009 [1]. Spatial coupling of multiple low-density parity-check (LDPC) codes was shown to be another way to achieve the capacity of binary erasure channels by Kudekar et al. [14], in 2011. This structure was originally introduced as convolutional LDPC codes by Felström and Zigangirov [8] in 1999 but the proof happened to come much later. These two codes are the state-of-the-art in modern coding theory and are strong competitors for practical applications. However, these codes achieve capacity only asymptotically, i.e. as the blocklength approaches infinity. Hence, the finite length performance of these codes is of primary concern among coding theorists and code designers.

Section 2 proposes a new construction of polar codes which allows us to construct polar codes of arbitrary blocklength that also achieve significantly higher rates than the original polar codes on memoryless erasure channels, at comparable blocklengths. In addition, these codes are cyclic and, specifically, are subcodes of RS codes so that a suitable cyclic encoder-RS decoder pair can be used if suboptimal performance suffices for the application as a trade-off for lower complexity and higher speed. The section details the construction of the transform, proves polarization for the construction and discusses a algebraic successive cancellation decoder. Simulation results are produced for the q-ary erasure channel and q-ary symmetric channel.

Spatially-coupled LDPC codes can be made more robust towards bursts of erasures than block LDPC codes. Since there are multiple applications that exhibit a burst erasure phenomenon, not necessarily in the traditional communication transmission sense, it is of interest to understand their performance in such scenarios. Hence, Section 3 analyzes the average performance of random regular spatially-coupled LDPC ensembles on burst erasure channels. A few practical applications are also provided as a motivation. The stopping sets in the Tanner graphs of the codes are used to characterize the performance in both the unexpurgated and expurgated scenarios. Since the two sections discuss different coding schemes in detail, conclusions are given at the end of each section for coherence and to avoid breaking the continuity.

Appendix A gives a discussion of the Cooley-Tukey fast Fourier transform and derives its Kronecker product formulation. Appendix B details the channel polarization for cyclic polar codes. Appendix C discusses the modified Forney's decoder for the small blocks in the FFT structure of cyclic polar codes and finally, Appendix D derives the Shannon capacity for the q-ary symmetric channel with erasures.

2. CYCLIC POLAR CODES[⋆]

2.1 Introduction to Polar Codes

Polar codes, invented by Arıkan [1], are binary linear codes that can achieve the symmetric capacity of an arbitrary binary-input discrete memoryless channel (B-DMC) under successive cancellation (SC) decoding. The transform used to construct the polar codes is based on the Kronecker product of the 2 × 2 kernel matrix,

$$G_2 = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}.$$

The transform equation for blocklength $N = 2^n$ is given by

$$G_N = B_N G_2^{\otimes n},$$

where B_N is the bit-reversal permutation matrix of size N, $A \otimes B$ is the Kronecker product of matrix A with matrix B and $G_2^{\otimes n} = \underbrace{G_2 \otimes G_2 \otimes \cdots \otimes G_2}_{n \text{ times}}$. Definition 1 in Section 2.3.2 gives a precise definition of the Kronecker product.

The binary input sequence $\underline{u} = (u_1, \ldots, u_N)$ consists of K information bits and $(N - K)$ known frozen bits. The codeword is encoded using $\underline{v} = \underline{u} G_N$ and then transmitted via N independent uses of the underlying B-DMC W. The successive cancellation decoder attempts to decode the i^{th} bit u_i given the knowledge of the received vector $\underline{y}$, which is a noisy observation of $\underline{v}$, and all the previously decoded inputs u_1^{i-1}. This allows one to view the i-th input bit as being transmitted over the

[⋆] Reprinted, with permission from N. Rengaswamy and H.D. Pfister, Cyclic Polar Codes, In *Proc. IEEE Int. Symp. Inform. Theory*, pages 1287-1291, June 2015. © 2015 IEEE.

coordinate channel $W_N^{(i)} : \{0,1\} \longrightarrow \mathcal{Y}^N \times \{0,1\}^{i-1}$ with transition probabilities

$$W_N^{(i)}(y_1^N, u_1^{i-1}|u_i) = \frac{1}{2^{N-1}} \sum_{u_{i+1}^N} W^N(y_1^N|u_1^N),$$

where $1 \leq i \leq N$ and $W^N(y_1^N|u_1^N) = \prod_{j=1}^N W(y_j|v_j)$ for the B-DMC $W : \{0,1\} \longrightarrow \mathcal{Y}$. Therefore, for coordinate channel $W_N^{(i)}$, bit u_i is the input and the output vector $\underline{y}$ combined with the $(i-1)$ previously decoded inputs, u_1^{i-1}, are the outputs. These are the channels that the successive cancellation decoder "sees" even though the actual transmission of $\underline{v}$ is over N independent uses of the "actual", physical, channel W. Figs. 2.1, 2.2 and 2.3 show the evolution of coordinate channels for blocklengths $N = 2, 4$ and 8, respectively, for an underlying channel W. A numerical example for $W \triangleq \text{BEC}(\epsilon = 0.5)$ is also shown. The numbers in the figures represent average erasure rates of the corresponding bits under successive cancellation decoding. The process of initializing the channel density for W, which is ϵ for BEC(ϵ), and allowing it to evolve until the input stage to determine the average erasure rates of the input coordinate channels is called *density evolution*. For an arbitrary B-DMC W, the coordinate channels are obtained through channel combining operations [1].

The idea of *polarization* is that, for large values of n, the capacities of these coordinate channels either approach 1, for "good" channels, or approach 0, for "bad" channels, and *no* value in between. This means that each coordinate channel is polarized to either full capacity or zero capacity. Hence, information can be transmitted at rate 1 over the "good" channels and the bits in $\underline{u}$ corresponding to the "bad" channels can be *frozen* to a fixed value, thereby implying a transmission at rate 0. It is important to note that the previous statement is valid because the notion of "channel" in coding theory is from the decoder's perspective, and not necessarily the physical channel.

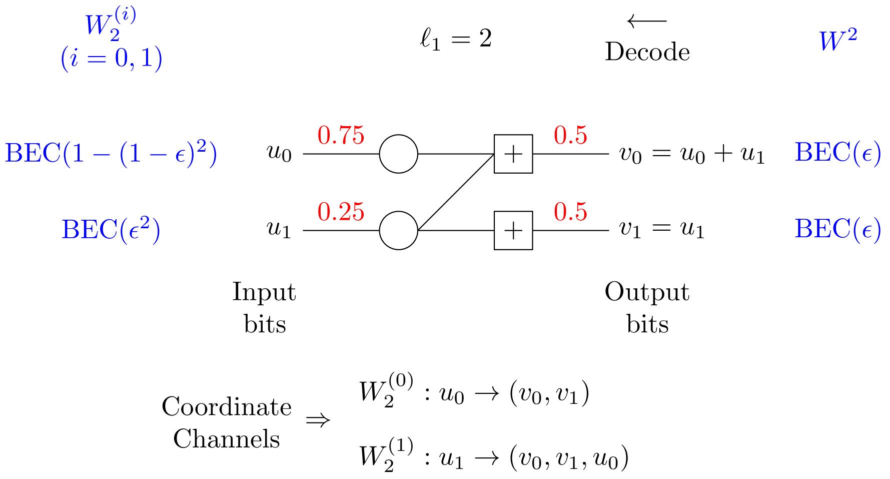


Figure 2.1: Coordinate channels for binary polar code of blocklength $N = 2$ over an underlying binary erasure channel $W \triangleq \text{BEC}(\epsilon)$. A numerical example for $W \triangleq \text{BEC}(0.5)$ is also shown. The numbers represent average erasure rates of the corresponding bits under successive cancellation decoding. The input erasure rates are obtained through one stage of density evolution.

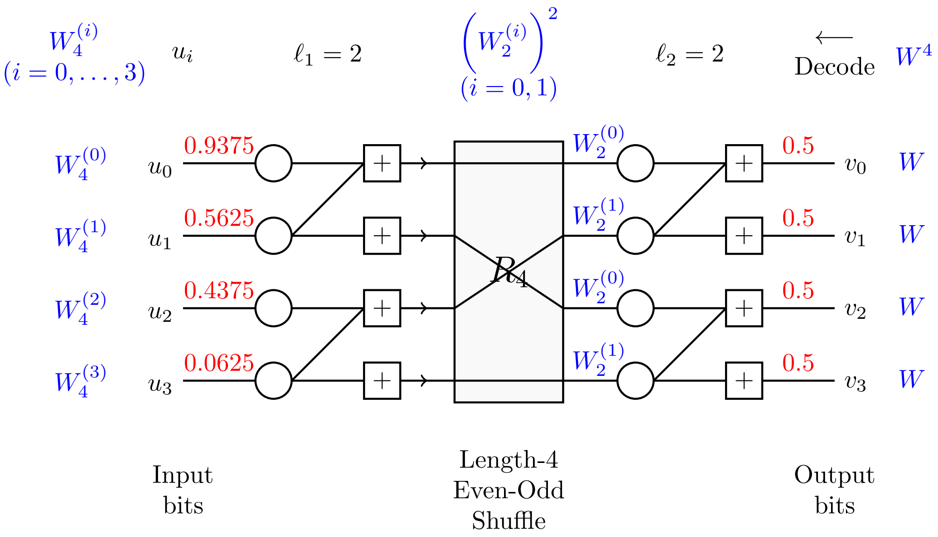


Figure 2.2: Coordinate channels for binary polar code of blocklength $N = 4$ over a general channel W. A numerical density evolution example over BEC(0.5) is also shown.

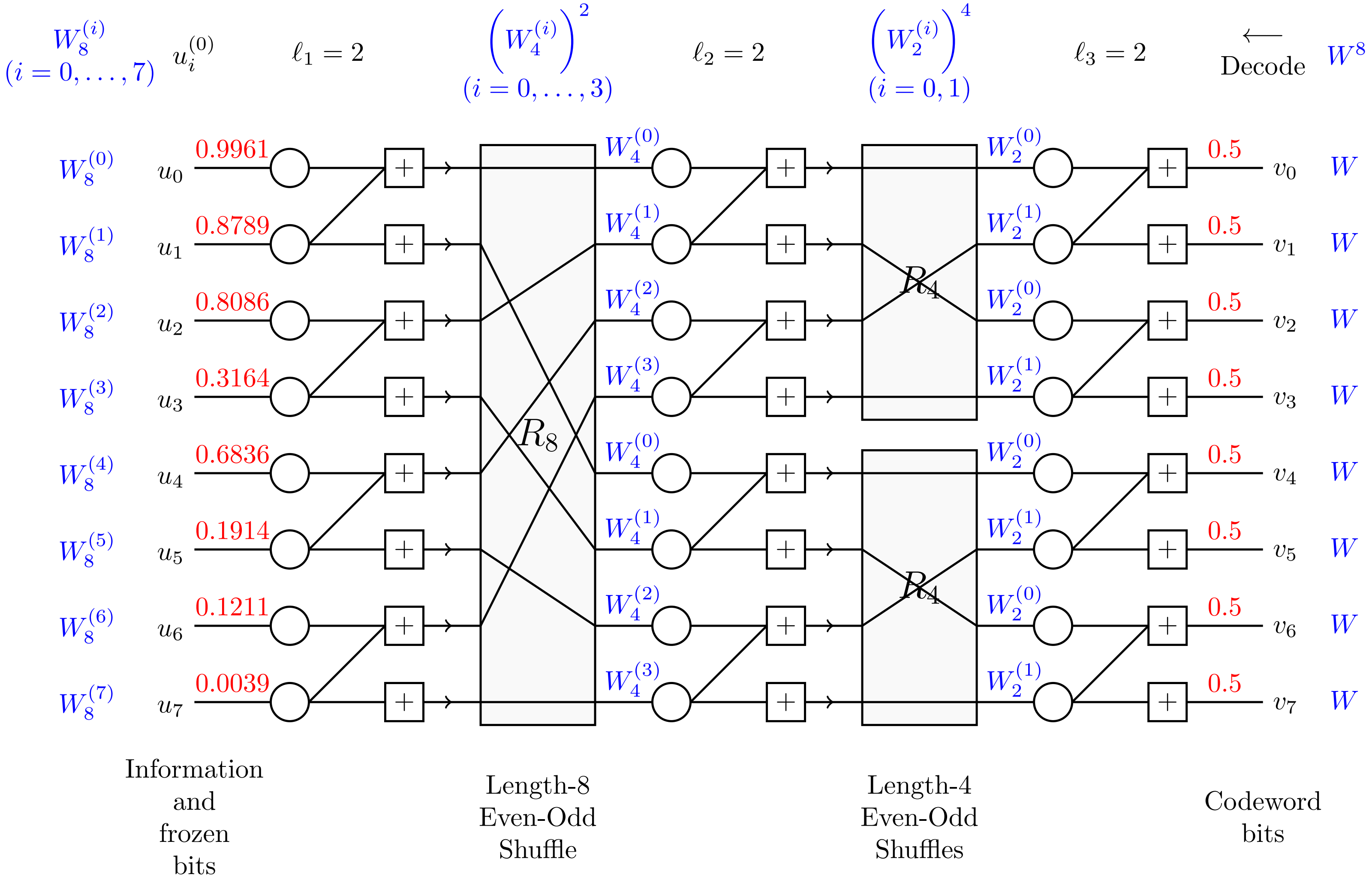


Figure 2.3: Coordinate channels for binary polar code of blocklength $N = 8$ over a general channel W. A numerical density evolution example for BEC(0.5) is also shown.

For a given blocklength, the design phase for the polar codes computes the capacities of the coordinate channels and decides the information and frozen bits according to a target rate or a target block error rate for the code. The indices corresponding to information bits are collected in a set $\mathcal{A}$ so that $|\mathcal{A}| = K$. For example, in Fig. 2.3, if we fix the target block error rate as $\delta = 0.1$ so that $P_B \leq \delta$, then we have $\mathcal{A} = \{7\}$ and hence, the rate of the code is $R = 1/8$. The design procedure is described in Section 2.5. The rest of the bits are frozen to values known both to the encoder and decoder, i.e. they carry no information from the source but help the successive cancellation decoder immensely. Typically, these bits are frozen to zeros. Hence, for our example, $u_0 = \cdots = u_6 = 0$.

Given the notion of polar coding, let us see how these codes achieve capacity. $I(W)$ denotes the symmetric capacity of the underlying channel W, i.e. the maximum amount of information that can be transmitted reliably over one use of the channel W, subject to using the input values 1 and 0 with equal frequency.

Polar codes achieve capacity because the fraction of channels that are "good" is equal to the symmetric capacity $I(W)$ of the underlying channel W, at sufficiently large blocklengths. Mathematically, this means that for any $\theta \in (0, 1)$,

$$\lim_{n\to\infty} R_n = \lim_{n\to\infty} \frac{1}{N} \left| \left\{ i : I(W_N^{(i)}) \in (1-\theta, 1] \right\} \right| = I(W) \textit{ for } N = 2^n. \tag{2.1}$$

Refer the notes of Pfister [23] for a very good introduction to polar codes. In this thesis, the following terminologies refer to Arıkan's G_2 polar codes: original, standard or binary polar codes.

2.2 Background Work

Further work has shown that polar codes can be constructed using larger kernel sizes by using an $\ell \times \ell$ binary matrix G_ℓ as the base matrix in the Kronecker product. Korada, Şaşoğlu, and Urbanke established that, as long as the transformation matrix G is not upper triangular it will polarize the coordinate channels [13]. They also showed that the rate of polarization can be strictly better than the original construction when $\ell = 16$.

In [7], Şaşoğlu, Telatar, and Arıkan show that the original polar code construction achieves the symmetric capacity of q-ary channels when q is prime. Mori and Tanaka consider polar codes over non-binary alphabets in [19, 20] and use Reed-Solomon (RS) and algebraic geometry codes to construct good polarizing kernels.

In this work, we construct polar codes, of arbitrary blocklength, that are also cyclic. Unlike previously proposed constructions, we use mixed kernel sizes to exploit connection with non-binary Galois field Fourier transforms (GFFTs) and fast Fourier transforms (FFTs). Using this construction, the information sequence $\underline{u}$ and the code sequence $\underline{v}$ become Fourier-transform pairs.

The design phase of these codes defines the positions of the frozen symbols. These symbols are set to zero in the Fourier transform of all codewords. Using the polynomial representation of messages and codewords, it follows that the frozen symbols define a set of common roots for all the code polynomials. Thus, the code is cyclic and we refer to these codes as *cyclic polar codes*. In this work, we show that these codes achieve the symmetric capacity of q-ary erasure channels and that these codes achieve higher rates than the original polar codes on memoryless erasure channels, at comparable finite blocklengths.

One benefit of these codes is that they can be made backwards compatible for a system that currently uses RS codes. This is because cyclic polar codes can be designed to be subcodes of higher rate RS codes. The polarizing matrices used at each stage of the transform act essentially identical to RS codes during the successive cancellation decoding process. Overall, while a standard RS decoder for the whole code only has one chance to correct all errors and erasures, the decoder of a cyclic polar code can exploit multistage decoding that converts some errors into erasures at each stage. Thus, the existing RS decoder in the system could be used with some performance penalty and a SC decoder could be used to improve performance.

Arıkan discusses systematic polar codes in [2]. Since the codes proposed here are cyclic, a systematic encoder can be realized by implementing suitable message mapping prior to the (non-systematic) encoder, so that the system is backwards compatible.

The remainder of this discussion is organized as follows. Section 2.3 discusses the GFFT, the Cooley-Tukey FFT algorithm and the channel definitions. Section 2.4 describes the cyclic polar code construction and decoding. Section 2.5 considers code design and Section 2.6 discusses results. Finally, Section 2.7 concludes this work.

2.3 Preliminaries

2.3.1 Galois-Field Fourier-Transform

Let $\mathbb{F} \triangleq \mathbb{F}_q$ denote the Galois field with q elements, $\alpha \in \mathbb{F}$ be a distinguished primitive element, and $\omega_\ell = \alpha^{(q-1)/\ell}$ be a primitive ℓ-th root of unity (i.e., $\ell \mid q-1$). Then, the length-ℓ Galois-field Fourier-transform (GFFT) of the vector $\underline{v} = (v_0, \ldots, v_{\ell-1})$ is given by

$$u_i = [F_\ell \underline{v}]_i = \sum_{j=0}^{\ell-1} \omega_\ell^{ij} v_j,$$

where the matrix F_ℓ is defined by $[F_\ell]_{i,j} \triangleq \omega_\ell^{ij}$. The inverse Fourier transform is given by

$$v_j = \ell^{-1}[F'_\ell \underline{u}]_j = \ell^{-1} \sum_{i=0}^{\ell-1} \omega_\ell^{-ij} u_i,$$

where the matrix F'_ℓ is defined by $[F'_\ell]_{i,j} \triangleq \omega_\ell^{-ij}$ and ℓ^{-1} is multiplicative inverse of ℓ in $\mathbb{F}_q$. Since the naïve complexity of this transform is $O(\ell^2)$, we use the reduced complexity FFT version popularized by Cooley and Tukey [6].

2.3.2 Cooley-Tukey Fast Fourier Transform

Let a and b be arbitrary positive integers and define $\ell = ab$, $\gamma = \omega_\ell^b$, and $\beta = \omega_\ell^a$. It is easy to verify that the elements γ and β have multiplicative orders a and b in the field $\mathbb{F}_q$. The Cooley-Tukey formula [3, 6] for $\underline{u} = F_\ell \underline{v}$ is given by

$$u_{ai'+i''} = \sum_{j'=0}^{b-1} \beta^{j'i'} \left[\omega_\ell^{j'i''} \left(\sum_{j''=0}^{a-1} \gamma^{j''i''} v_{j'+bj''} \right) \right]. \tag{2.2}$$

From this equation, one can see there are four steps in the Cooley-Tukey FFT. First, b Fourier transforms of length a are computed on b interleaved blocks. Next, the i-th element of the resulting vector, which is indexed by $i = bi'' + j'$, is multiplied by the twiddle factor $\omega_\ell^{j'i''} = \omega_\ell^{\lfloor i/b \rfloor (i \bmod b)}$. Then, a Fourier transforms of length-b are computed on a adjacent blocks in the resulting vector. Finally, the output vector is formed by deinterleaving the result of the previous step by a. The complexity is now reduced to $O(\ell(a+b))$. This process is described in more detail in Appendix A.1.

Definition 1. Let A and B be $a \times a$ and $b \times b$ square matrices. The Kronecker product of A and B is defined to be

$$A \otimes B \triangleq \begin{bmatrix} A_{1,1}B & \cdots & A_{1,a}B \\ \vdots & \ddots & \vdots \\ A_{a,1}B & \cdots & A_{a,a}B \end{bmatrix}.$$

Definition 2. For a vector $\underline{v}$ of length ab, the perfect-shuffle permutation matrix, $S_{a,b}$, is the permutation matrix associated with writing v into an $a \times b$ matrix column-wise and then reading it out row-wise. Using this definition, one finds that

$$S_{a,b}^T(A \otimes B)S_{a,b} = (B \otimes A),$$

where $S_{a,b}^T = S_{b,a}$.

Based on these definitions, we give an expression for the transform using matrix operations.

Lemma 3. *The Cooley-Tukey decomposition of the length-ab fast Fourier transform can be expressed in terms of Kronecker products as*

$$\begin{aligned} F_{ab} &= S_{b,a}(I_a \otimes F_b)D_{a,b}(F_a \otimes I_b) \\ &= (F_b \otimes I_a)S_{b,a}D_{a,b}(F_a \otimes I_b), \end{aligned}$$

where I_a denotes the $a \times a$ identity matrix and the diagonal twiddle-factor matrix is defined by $[D_{a,b}]_{i,i} = \omega_{ab}^{\lfloor i/b \rfloor (i \bmod b)}$.

This can be extended to the general mixed-radix FFT of length $N = \prod_{m=1}^{n} \ell_m$ by recursion.

Lemma 4. *Let $p_j = \prod_{m=1}^{j} \ell_m$. Then the length-N fast Fourier transform can be decomposed as*

$$F_N = U_n U_{n-1} \cdots U_1, \tag{2.3}$$

where

$$U_m = (S_{N/p_m,\ell_m} D_{\ell_m,N/p_m} \otimes I_{p_m/\ell_m})(F_{\ell_m} \otimes I_{N/\ell_m}). \tag{2.4}$$

For the inverse transform, $F_N' = U_n' U_{n-1}' \cdots U_1'$ where U_m' is defined by replacing F_{ℓ_m} by F_{ℓ_m}' and $D_{a,b}$ by $D_{a,b}' = D_{a,b}^{-1}$.

The proofs are given in Appendix A.1. The existence of this Kronecker-product formulation of the GFFT is a key reason that one can construct cyclic polar codes based on the GFFT. We note that an alternative construction based on the Good-Thomas (or prime-factor) FFT is also possible [3]. The main difference is that no twiddle factors are required but the block sizes must be relatively prime.

2.3.3 Channels

A *q-ary symmetric channel with erasures* is determined by the parameters (q, β, ϵ) and is denoted by QSCE(q, β, ϵ). Its transition probabilities are defined, for $x \in \mathbb{F}$ and $y \in \mathbb{F} \cup \{?\}$, by

$$W(y|x) = \begin{cases} 1 - \beta - \epsilon & \text{if } y = x, \\ \frac{\beta}{q-1} & \text{if } y \in \mathbb{F} \backslash \{x\}, \\ \epsilon & \text{if } y =? \end{cases}$$

The Shannon capacity of this channel is derived in Appendix D. Two important special cases of this channel can be obtained by setting either of its parameters to zero. The *q-ary erasure channel* with parameter ϵ is denoted and defined as QEC(q, ϵ) $\triangleq$ QSCE$(q, 0, \epsilon)$. The *q-ary symmetric channel* with parameter β is denoted and defined as QSC(q, β) $\triangleq$ QSCE$(q, \beta, 0)$. For simplicity of notation, we will denote these channels as QSCE(β, ϵ), QEC(ϵ) and QSC(β), respectively.

2.4 Cyclic Polar Code Construction

2.4.1 Overview

In this section, we describe our construction of cyclic polar codes over the finite field $\mathbb{F}$ with q elements. The construction depends on the block length N, which must divide $q - 1$, and the ordered integer factorization $N = \prod_{m=1}^{n} \ell_m$ where each

ℓ_m is a positive integer. In contrast to the SC decoder of Arıkan's uniform G_2 polar codes, some changes are required. First, the component matrices are not necessarily 2×2 or even the same size. Next, there are multiplications by twiddle factors after each encoding stage to make the overall transform into a Fourier transform.

The encoder mapping follows from the mixed-radix Cooley-Tukey inverse FFT decomposition for $N = \ell_1 \ell_2 \cdots \ell_n$ based on (2.4). In particular, let $u_i^{(0)} \in \mathbb{F}$ for $i = 0, 1, \ldots, N-1$ be the GFFT of a codeword. Each element of the spectrum is either assigned to carry information or to be frozen to 0. Let $\mathcal{A} \subseteq \{0, 1, \ldots, N-1\}$ be the set of indices that carry information and let its complement $\mathcal{A}^c$ denote the set of indices that are frozen to 0. The set $\mathcal{A}$ is the output of the code design process discussed in Section 2.5.

Recollect that

$$u_i = \sum_{j=0}^{N-1} \omega_N^{ij} v_j,$$

where ω_N has order N in $\mathbb{F}$. In polynomial notation, with $v(x) = \sum_{j=0}^{N-1} v_j x^j$, we have

$$u(x) = \sum_{i=0}^{N-1} u_i x^i = \sum_{i=0}^{N-1} v(\omega_N^i) x^i.$$

So, we see that u_i's are evaluations of $v(x)$. Given $\mathcal{A}^c$, the set of indices frozen to zeros in $u(x)$ such that $u_i = 0 \; \forall \; i \in \mathcal{A}^c$, there exists a generator $g(x)$ such that

$$v(x) = u_{\mathcal{A}}(x) g(x) = u_{\mathcal{A}}(x) \prod_{i \in \mathcal{A}^c} (x - \omega_N^i),$$

where $u_{\mathcal{A}}(x)$ represents the information polynomial of degree at most $K-1$. Hence, we have a cyclic code. Since we need $N | (q-1)$ for ω_N to exist in $\mathbb{F}$, the field size must grow with the blocklength.

The encoder proceeds by filling the vector $u_i^{(0)}$ and using the mixed-radix Cooley-

Tukey algorithm to compute the inverse Fourier transform. The formula for one stage of the transform is given by,

$$\underline{u}^{(m)} = U'_m \underline{u}^{(m-1)}$$

where U'_m is defined in (2.4) for $m = 1, 2, \ldots, n$ and $\underline{u}^{(n)} = \underline{v}$. An example with $N = 5 \cdot 3 = 15$ is shown in Fig. 2.4 and with $N = 5 \cdot 3 \cdot 2 = 30$ is shown in Fig. 2.5.

Like other polar code constructions, the set of frozen indices is chosen using a design process that depends on the channel. In this work, we focus on a number of special cases that allow simplifications. First, we consider the case where $N = 2^n$ is a power of 2 and q is prime. In this case, polarization is based on the standard radix-2 Cooley-Tukey FFT and the decoder can be implemented efficiently for arbitrary q-ary channels. After that, we consider the q-ary erasure channel for arbitrary $\mathbb{F}_q$ because both the decoder and the design process can be implemented efficiently in this case too. Subsequently, we also discuss a decoding strategy in the presence of errors and erasures.

2.4.2 Arbitrary q-ary Channel with q prime and $N = 2^n$

For $N = 2^n$, the design and decoding operations are quite similar to standard polar codes. Based on the factor-graph perspective on polar codes [18], the successive cancellation decoder is equivalent to a particular message-passing schedule on a factor graph with q-ary probability messages. In particular, one needs to keep track of q probabilities for each symbol in the graph. For the variable denoted by a, these will be denoted as $p_a(x) = \Pr(a = x)$ for $x \in \mathbb{F}$.

Consider a 2×2 butterfly operation defined by the input (a_0, a_1), output (b_0, b_1),

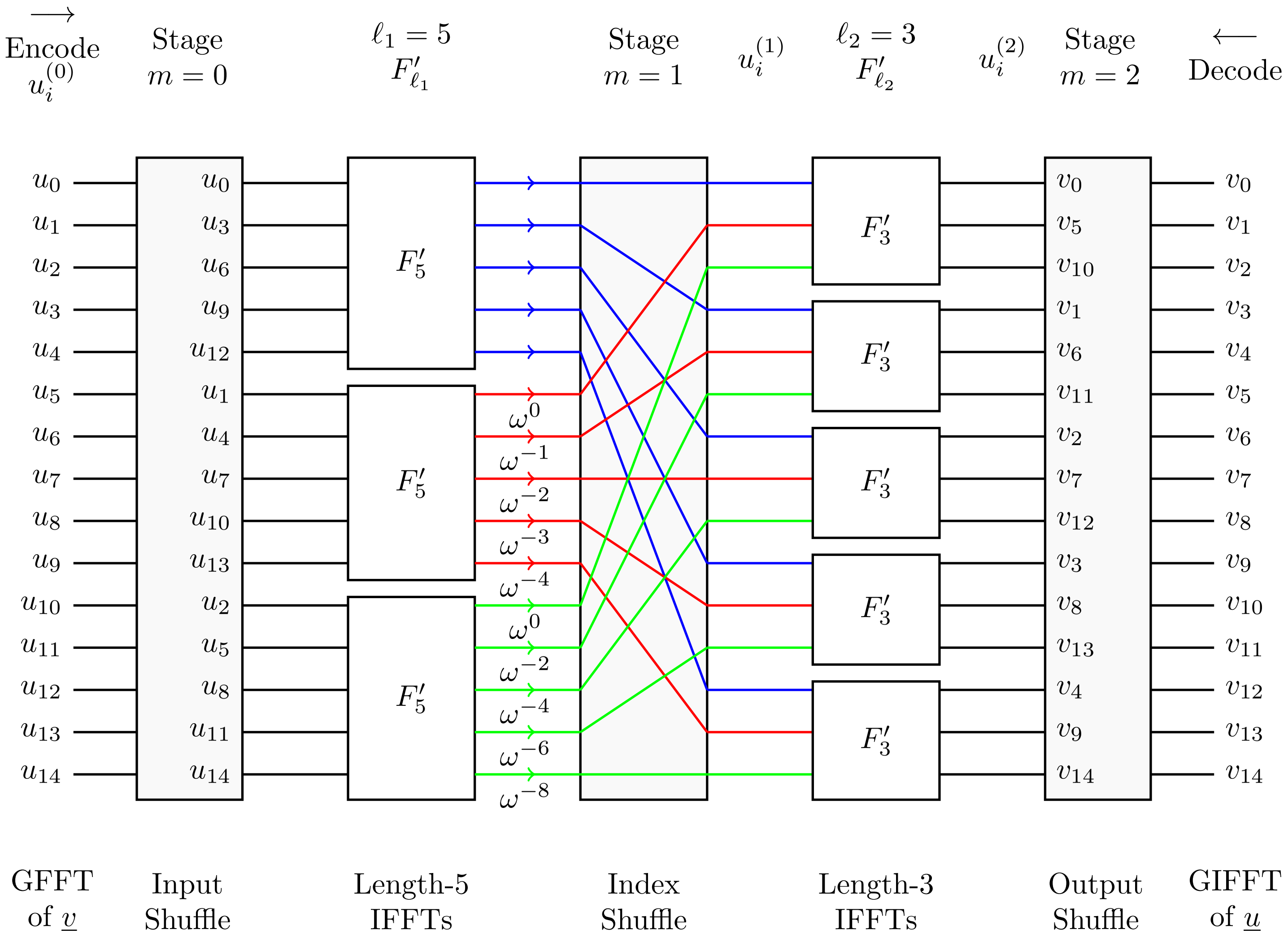

$\longrightarrow$ Encode $u_i^{(0)}$
Stage $m = 0$
$\ell_1 = 5$ F'_{ℓ_1}
Stage $m = 1$
$u_i^{(1)}$
$\ell_2 = 3$ F'_{ℓ_2}
$u_i^{(2)}$
Stage $m = 2$
$\longleftarrow$ Decode
u_0 u_1 u_2 u_3 u_4 u_5 u_6 u_7 u_8 u_9 u_{10} u_{11} u_{12} u_{13} u_{14}
u_0 u_3 u_6 u_9 u_{12} u_1 u_4 u_7 u_{10} u_{13} u_2 u_5 u_8 u_{11} u_{14}
F'_5
F'_5
F'_5
ω^0 ω^{-1} ω^{-2} ω^{-3} ω^{-4} ω^0 ω^{-2} ω^{-4} ω^{-6} ω^{-8}
F'_3
F'_3
F'_3
F'_3
F'_3
v_0 v_5 v_{10} v_1 v_6 v_{11} v_2 v_7 v_{12} v_3 v_8 v_{13} v_4 v_9 v_{14}
v_0 v_1 v_2 v_3 v_4 v_5 v_6 v_7 v_8 v_9 v_{10} v_{11} v_{12} v_{13} v_{14}
GFFT of $\underline{v}$
Input Shuffle
Length-5 IFFTs
Index Shuffle
Length-3 IFFTs
Output Shuffle
GIFFT of $\underline{u}$

Figure 2.4: An example for $N = 5 \cdot 3 = 15$ over $\mathbb{F}_{16}$ depicting the Cooley-Tukey fast Fourier transform. The F'_5 and F'_3 blocks are a naïve implementation of the inverse Fourier transform. ω is a N^{th} root of unity in $\mathbb{F}_{16}$. Some lines are colored just for visual clarity as they cross paths during shuffling operations.

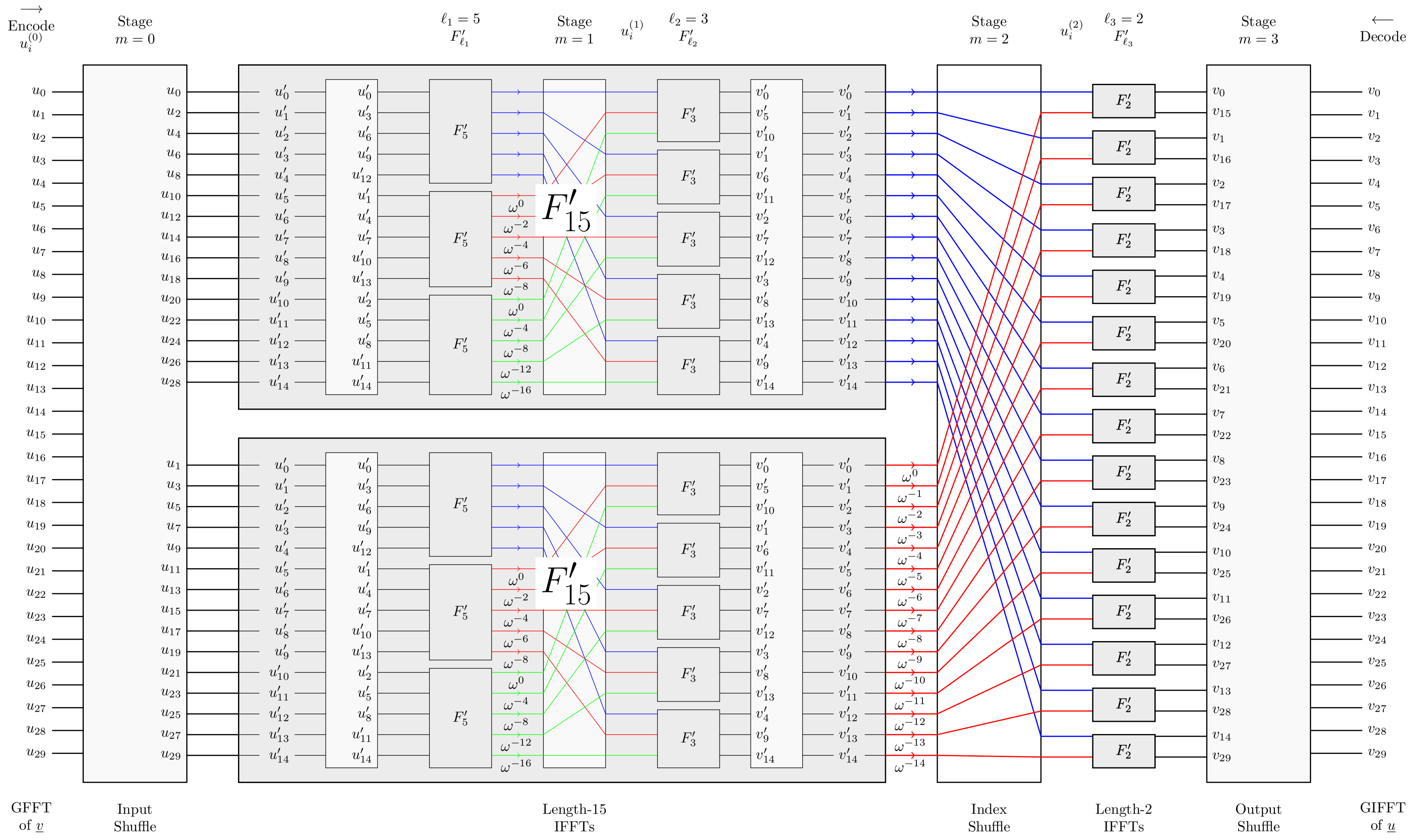

$\longrightarrow$ Encode $u_i^{(0)}$
Stage $m=0$
$\ell_1 = 5$ F'_{ℓ_1}
Stage $m=1$
$u_i^{(1)}$
$\ell_2 = 3$ F'_{ℓ_2}
Stage $m=2$
$u_i^{(2)}$
$\ell_3 = 2$ F'_{ℓ_3}
Stage $m=3$
$\longleftarrow$ Decode
F'_5
F'_3
F'_2
F'_{15}
ω^0 ω^{-2} ω^{-4} ω^{-6} ω^{-8} ω^0 ω^{-4} ω^{-8} ω^{-12} ω^{-16}
ω^0 ω^{-1} ω^{-2} ω^{-3} ω^{-4} ω^{-5} ω^{-6} ω^{-7} ω^{-8} ω^{-9} ω^{-10} ω^{-11} ω^{-12} ω^{-13} ω^{-14}
GFFT of $\underline{v}$
Input Shuffle
Length-15 IFFTs
Index Shuffle
Length-2 IFFTs
Output Shuffle
GIFFT of $\underline{u}$

Figure 2.5: An example for $N = 5 \cdot 3 \cdot 2 = 30$ over $\mathbb{F}_{31}$ depicting the Cooley-Tukey fast Fourier transform. The F'_5, F'_3 and F'_2 blocks are a naïve implementation of the inverse Fourier transform. ω is a N^{th} root of unity in $\mathbb{F}_{31}$. Some lines are colored just for visual clarity as they cross paths during shuffling operations.

and the relations

$$b_0 = a_0 + a_1, \tag{2.5}$$

$$b_1 = a_0 + \alpha a_1. \tag{2.6}$$

Now, to estimate (a_0, a_1) from (b_0, b_1) in the polar decoding order, we can write

$$\hat{a}_0 = (1-\alpha)^{-1}(b_1 - \alpha b_0), \tag{2.7}$$

$$\hat{a}_1 = b_0 - a_0, \tag{2.8}$$

$$\hat{a}_1' = \alpha^{-1}(b_1 - a_0). \tag{2.9}$$

Using these equations, one can use standard techniques from low-density parity-check codes to compute the optimal soft estimates of (a_0, a_1) from soft estimates of (b_0, b_1) [25, Section 2.4]. Since all arithmetic is modulo the prime q, the soft estimate for the addition of two symbols is given by the circular convolution of their probability vectors. For $b_0 = a_0 + a_1$, we have,

$$p_{b_0}(y) = \textstyle\sum_{x \in \mathbb{F}} p_{a_0}(x) p_{a_1}(y - x)\,.$$

Similarly, the soft estimate for the multiplication of a symbol by a fixed scalar is given by a permutation of the probability vector. For $b_1' = \beta b_1$, we have

$$p_{b_1'}(x) = \Pr(b_1 = \beta^{-1}x) = p_{b_1}(\beta^{-1}x).$$

Also, independent estimates (e.g., $\hat{a}_1$ and $\hat{a}_1'$) are combined by renormalizing the product of their probability vectors:

$$p_{a_1|\hat{a}_1,\hat{a}'_1}(x) = \frac{p_{\hat{a}_1}(x)p_{\hat{a}'_1}(x)}{\sum_{x'\in\mathbb{F}} p_{\hat{a}_1}(x')p_{\hat{a}'_1}(x')}.$$

In this case, the SC decoder can be defined recursively for the whole graph based on these operations. Hard decisions are made for the information symbols based on the maximum value in their associated probability vectors.

Generalization to the mixed-radix case with arbitrary block sizes is straightforward but computationally expensive. Soft estimates are stored as vectors of probabilities but a-posteriori-probability (APP) decoding is required for the FFT blocks, which is why we resort to algebraic hard-decision successive-cancellation decoding.

2.4.3 Algebraic Erasures Decoding

For erasure channels, cyclic polar codes can be efficiently designed and decoded for an arbitrary $\mathbb{F}_q$ and $N|(q-1)$. Each factor ℓ of N requires the decoding of a $\ell \times \ell$ matrix F'_ℓ defined by $[F'_\ell]_{i,j} \triangleq \omega_\ell^{-ij}$. Similar to [19], polar decoding for F'_ℓ essentially requires the decoding of a nested sequence of RS Codes. Fortunately, erasures only RS decoding can be implemented efficiently using Forney's algorithm [9].

Let $\underline{v} = F'_\ell \underline{u}$ and $\underline{y}$ be an observation of $\underline{v}$ through an erasure channel. The polar decoding problem for F'_ℓ is, for $j = 0, 1, \ldots, \ell - 1$, decode u_j from $\underline{y}$ and $u_0, \ldots, u_{j-1}$. For the j-th decode, this can be viewed as decoding a known coset of an $(\ell, \ell - j)$ RS code. To use Forney's algorithm, we calculate j modified syndromes by removing the contribution of known inputs. The details of this process are discussed in Appendix C.

Due to the nature of Forney's algorithm, decoding either recovers all or none of the unknown inputs. This leads to the following rule for the SC decoding of each block: *if ν inputs of the block are known and at most ν outputs are erased, then use*

Forney's decoder to recover all unknown inputs; otherwise, pass an erasure as the decoded input. The operations performed during SC decoding are given below:

- Begin with the output $\underline{v}$ set to the received (hard) values from the channel.

- While decoding the set of blocks F'_{ℓ_m} for the j^{th} input, $j = 0, \ldots, \ell_m - 1$, use the decoding rule above and pass the newly decoded j^{th} inputs at stage m to the j^{th} F'_{N/ℓ_m} block from the top at the previous stage. Then, recurse and execute the decoder at that stage.

- While decoding blocks F'_{ℓ_1}, use the frozen symbols at the input as knowledge to compute syndromes for the Forney decoder. A block F'_{ℓ_1} that does not have any frozen symbols must have all outputs already known for successful decoding. Once the outputs for these blocks are determined, pass them forward to the next stage.

- When the procedure returns to the set of blocks F'_{ℓ_m} with an update for the j^{th} input, the updated inputs are used to decode the next input according to the above rule. Based on the SC decoder, the outputs of the block are not updated until all inputs are "recovered".

For the erasure channel, Forney's decoder is run exactly once for each block (when the number of known inputs and outputs equals the block length). The decoding complexity of Forney's algorithm for a length-ℓ block is at most $C\ell^2$ operations for some $C > 0$. Since there are $\prod_{j \neq m} \ell_j = N/\ell_m$ blocks at stage ℓ_m, the decoding complexity is bounded by

$$\sum_{m=1}^{n} \prod_{j \neq m} \ell_j \left(C\ell_m^2 \right) = CN \sum_{m=1}^{n} \ell_m \leq CNn \max_m \ell_m.$$

2.4.4 Algebraic Errors and Erasures Decoding

The approach in Section 2.4.3 can also be extended to handle errors and erasures. In this case, each small block is decoded using algebraic errors and erasures decoding. The decoding of each small block results in success, failure, or miscorrection. In the event of decoder failure, an erasure is passed back to the previous stage. Otherwise, the value estimated by algebraic decoding is passed back to the previous stage.

Using this decoding strategy, cyclic polar codes can be efficiently designed and decoded for arbitrary q and $N|(q-1)$. For each of the ℓ_m decoding iterations of block F'_{ℓ_m}, the Berlekamp-Massey (BM) algorithm is used to obtain the error-erasure locator polynomial which is fed into Forney's decoder to correct errors and erasures [4, Section 7.5][1]. Unlike the erasure case, the decoding operation must be executed during each decoding stage. Thus, the decoding complexity is increased to $CNn \max_m \ell_m^2$.

Since each iteration of the SC decoder involves decoding a RS code whose minimum distance depends on the number of inputs already recovered, the decoder efficiency is increased if errors are converted into erasures during the multistage decoding process. This is because, with ν known input symbols, the decoder can correct t errors and e erasures iff $\nu \geq 2t + e$. Hence, this decoding strategy is sub-optimal. All of the intermediate channels in this process can be modeled as QSCE.

We need to perform density evolution of the error and erasure probabilities to design the code. Since density evolution is complex to be performed on arbitrary q-ary channels, a Monte Carlo design methodology is employed to compute the capacities of input coordinate channels for an arbitrary blocklength N. An example of the design results is discussed in Section 2.5.2.

[1] In Fig. 7.10 of [4, Section 7.5], the update equation in the left-most box above the bottom-most box in the flowchart must be $L \leftarrow r - (L - \rho)$ and not $L \leftarrow r - L - \rho$.

2.5 Code Design

2.5.1 Erasure Channels

The input parameters to the design procedure for the QEC(ϵ) are (N, q, ϵ, δ) where δ is the target block erasure rate. Consider the upper bound on P_B given by

$$P_B \leq \sum_{i \in \mathcal{A}} \epsilon_i^{(0)}, \tag{2.10}$$

where $\epsilon_i^{(0)}$ is the erasure probability of the coordinate channel $W_N^{(i)}$ and $\mathcal{A}$ is the set of information symbols [1]. The design procedure chooses $\mathcal{A}$ to be the largest subset $S \subseteq \{0, 1, \ldots, N-1\}$ such that $\sum_{i \in \mathcal{S}} \epsilon_i^{(0)} \leq \delta$. This design strategy is applicable to both binary polar codes and cyclic polar codes.

The erasure probabilities of the output symbols are initialized to $\epsilon_0^{(n)} = \epsilon$. The design process commences by performing density evolution – recursively computing the erasure rates of the coordinate channels from stage $m = n$ down to $m = 0$. Due to the structure of polar codes, there will be at most $\ell_{m+1}\ell_{m+2} \cdots \ell_n$ distinct erasure probabilities in the m-th stage. The i-th distinct erasure probability at stage m is denoted by $\epsilon_i^{(m)}$ for $i = 0, 1, \ldots, \ell_{m+1}\ell_{m+2} \cdots \ell_n - 1$ and $m = 0, 1, \ldots, n$.

Consider the erasure decoding of a single block of length ℓ as described in Section 2.4.3. Given the knowledge of j previously decoded inputs, the next input can be computed if and only if at least $\ell - j$ of the outputs are not erased. This is because the j known symbols imply that the output sequence lies in a known coset of an $(\ell, \ell - j)$ RS code that can correct j erasures. Thus, if the outputs are erased i.i.d. with probability ϵ' and j previous inputs are known, then next input is erased with probability $\psi(\ell, j, \epsilon')$ given by

$$\psi(\ell, j, \epsilon') \triangleq \sum_{i=0}^{(\ell-1)-j} \binom{\ell}{i} (1-\epsilon')^i (\epsilon')^{\ell-i}. \tag{2.11}$$

We note that this formula is due to Mori and Tanaka [19].

For an ordered factorization $N = \ell_1 \ell_2 \cdots \ell_n$, this implies that the distinct erasure probabilities of the coordinate channels satisfy the recursion

$$\epsilon_{\ell_m k+j}^{(m-1)} = \psi\left(\ell_m, j, \epsilon_k^{(m)}\right) \tag{2.12}$$

for $j = 0, 1, \ldots, \ell_m - 1$ and $k = 0, 1, \ldots, \ell_{m+1}\ell_{m+2}\cdots\ell_n - 1$.

Having established the stage-by-stage evolution of the erasure rates in (2.11) and (2.12), we re-state the polarization theorem in [1] for the case of a q-ary erasure channel (QEC).

Theorem 5. *For a QEC W with erasure rate ϵ, the input coordinate channels $W_N^{(i)}$ polarize in the sense that, for any fixed $\theta \in (0, 1)$, as N goes to infinity through multiples of positive integers, the fraction of indices $i \in \{0, 1, \ldots, N-1\}$ for which $\epsilon_i^{(0)} \in (1-\theta, 1]$ goes to ϵ and the fraction for which $\epsilon_i^{(0)} \in [0, \theta)$ goes to $(1-\epsilon)$.*

Proof. We will need the following properties of the mapping defined in (2.11) to motivate this proof.

Lemma 6. *Eqn.* (2.11) *defines a mapping from $\mathbb{R}$ to $\mathbb{R}^\ell$ with the following two properties:*

(*i*) *The mapping preserves the mean erasure rate through each stage of density evolution so that*

$$\frac{1}{\ell} \sum_{j=0}^{\ell-1} \psi(\ell, j, \epsilon') = \epsilon'. \tag{2.13}$$

(ii) If $\epsilon' \in (0,1)$, then the erasure rates of the new coordinate channels polarize away from the mean such that

$$\psi(\ell, \ell-1, \epsilon') < \epsilon' < \psi(\ell, 0, \epsilon'). \tag{2.14}$$

Based on these two properties, the martingale convergence analysis in [1] can be used to show that the erasure rates must polarize to 0 and 1 as $n \to \infty$ and that the fraction of coordinate channels with erasure rate 0 must be ϵ. A detailed proof is provided in Appendix B. □

To motivate the limit $n \to \infty$, let p be a prime that satisfies $\gcd(N,p) = 1$ for some $N = \ell_1 \ell_2 \cdots \ell_n$. Then, there exists an extension finite field $\mathbb{F}_{p^m} = \mathbb{F}_q$ with $m \leq N-1$ such that $N|(q-1)$. Of course, the field size may be exceedingly large for a given N and p.

2.5.2 An Example

Fig. 2.6 shows an example for $N = 15$ over $\mathbb{F}_{16}$ depicting the transform, density evolution process and code construction over QEC(ϵ). The design parameters chosen for this example are channel erasure rate $\epsilon = 0.5$ and maximum block erasure rate $\delta = 0.1$. The design methodology is described in Section 2.5.1. The various erasure rates at intermediate stages are shown in the graph for easy comprehension. For example, for the F_3' block, the output erasure probabilities are $\epsilon_0^{(2)} = 0.5$ and the input erasure probabilities are given by $\epsilon_0^{(1)} = 0.875$, $\epsilon_1^{(1)} = 0.5$, and $\epsilon_2^{(1)} = 0.125$. These values are repeated for the other two F_3' blocks as well, because we only track the distinct erasure probabilities at each stage and not all N indices.

According to (2.10), the information indices are chosen as $\mathcal{A} = \{8, 11, 13, 14\}$ and are represented with the prefix (D), for data, in the input side of the graph. Thus,

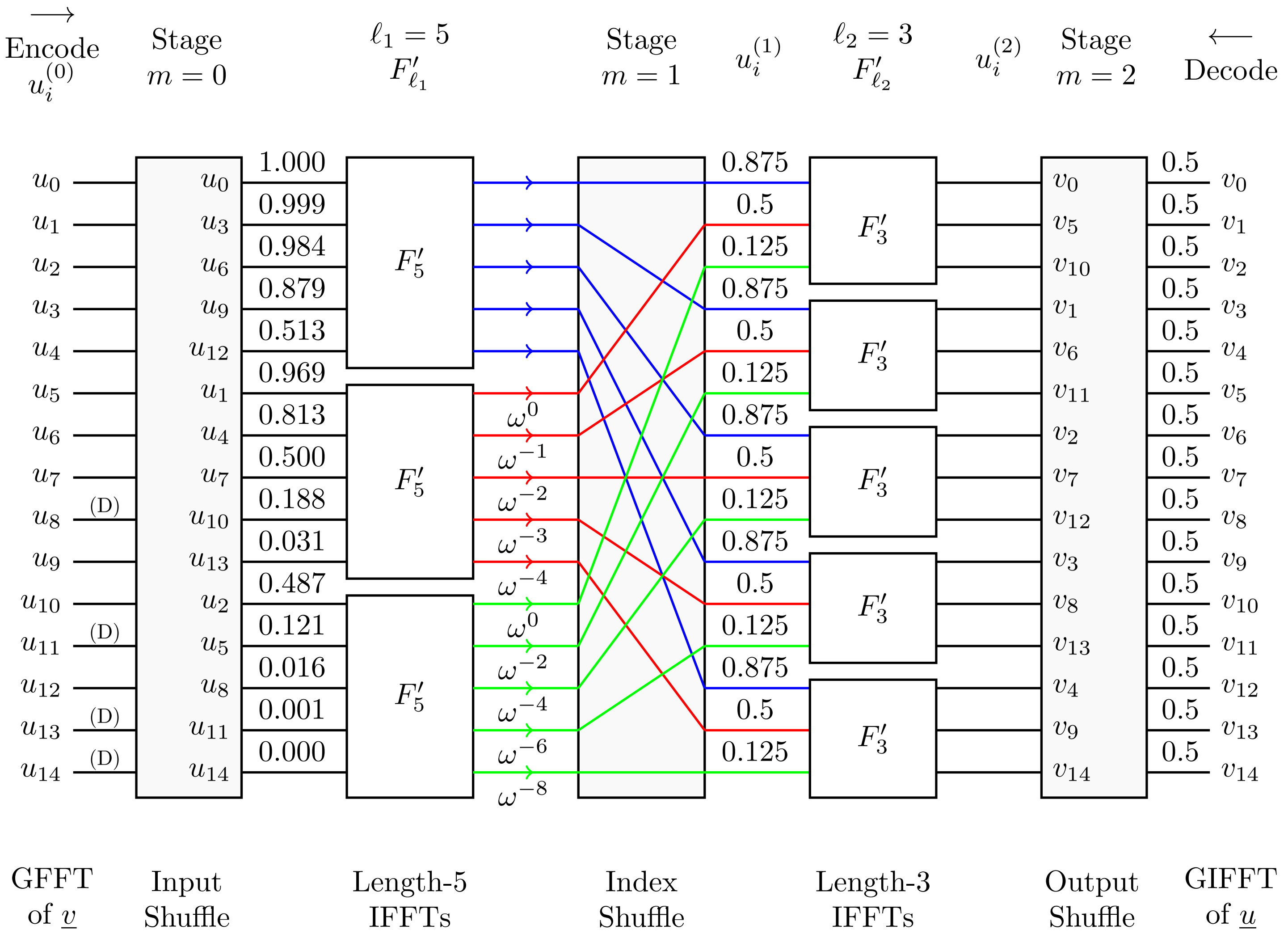

⟶
Encode
$u_i^{(0)}$
Stage
$m = 0$
$\ell_1 = 5$
F'_{ℓ_1}
Stage
$m = 1$
$u_i^{(1)}$
$\ell_2 = 3$
F'_{ℓ_2}
$u_i^{(2)}$
Stage
$m = 2$
⟵
Decode
u_0 u_1 u_2 u_3 u_4 u_5 u_6 u_7 u_8 u_9 u_{10} u_{11} u_{12} u_{13} u_{14}
(D) (D) (D) (D)
u_0 u_3 u_6 u_9 u_{12} u_1 u_4 u_7 u_{10} u_{13} u_2 u_5 u_8 u_{11} u_{14}
1.000 0.999 0.984 0.879 0.513 0.969 0.813 0.500 0.188 0.031 0.487 0.121 0.016 0.001 0.000
F'_5 F'_5 F'_5
ω^0 ω^{-1} ω^{-2} ω^{-3} ω^{-4} ω^0 ω^{-2} ω^{-4} ω^{-6} ω^{-8}
0.875 0.5 0.125 0.875 0.5 0.125 0.875 0.5 0.125 0.875 0.5 0.125 0.875 0.5 0.125
F'_3 F'_3 F'_3 F'_3 F'_3
v_0 v_5 v_{10} v_1 v_6 v_{11} v_2 v_7 v_{12} v_3 v_8 v_{13} v_4 v_9 v_{14}
0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5
v_0 v_1 v_2 v_3 v_4 v_5 v_6 v_7 v_8 v_9 v_{10} v_{11} v_{12} v_{13} v_{14}
GFFT
of $\underline{v}$
Input
Shuffle
Length-5
IFFTs
Index
Shuffle
Length-3
IFFTs
Output
Shuffle
GIFFT
of $\underline{u}$

Figure 2.6: An example for $N = 15$ over $\mathbb{F}_{16}$ depicting the transform, density evolution process and code construction. Design parameters: $\epsilon = 0.5$, $\delta = 0.1$. Information symbols are marked with (D). The F'_5 and F'_3 blocks are a naïve implementation of the inverse Fourier transform. ω is a N^{th} root of unity in $\mathbb{F}_{16}$. Some lines are colored just for visual clarity as they cross paths during shuffling operations.

the rate of this code is $\frac{4}{15} = 0.2667$. The generator polynomial is given by

$$g(x) = \prod_{i \in \mathcal{A}^c} (x - \omega_{15}^i).$$

Now, the encoder fills the indices in $\mathcal{A}^c$ with zeros and the other four indices with information, performs the transform to get the output vector $\underline{v}$ and transmits it via N independent uses of the underlying channel.

The results obtained for $N = 15$ over $\mathbb{F}_{256}$ on QSCE(0.5,0) using Monte Carlo design over $M = 10^4$ iterations are below. Each column from the far left of the matrix corresponds to the inputs of that $\ell = 5$ block from the top in Fig. 2.4. The $(p_{\text{error}}, p_{\text{erasure}})$ pair for the input channels are:

$$\begin{bmatrix} (0.9354, 0) & (0, 0.9999) & (0.0176, 0.9475) \\ (0.0613, 0.9354) & (0.0238, 0.9751) & (0.3257, 0.4885) \\ (0.2745, 0.7054) & (0.0140, 0.9718) & (0.0836, 0.4620) \\ (0.0167, 0.9799) & (0.0555, 0.8369) & (0.0516, 0.2145) \\ (0.0471, 0.9192) & (0.0858, 0.4822) & (0.0150, 0.0730) \end{bmatrix}$$

It can be noted that the resultant probabilities of error are significantly lower than the corresponding probabilities of erasure. The channels with high error probabilities are those decoded with no input information. While the capacity of QSCE$(0.5, 0)$ is 0.3753, the average capacity of the input channels are significantly low at 0.1807 due to the use of a sub-optimal decoding strategy, as noted in Section 2.4.4. The design can be significantly improved if an APP decoder is employed in place of the hard errors and erasures decoder, at the cost of additional computational complexity.

2.6 Results and Discussion

Binary and cyclic polar codes were designed for various blocklengths on BEC(0.5) and QEC(0.5), respectively, for a target block erasure rate $\delta = 0.1$. The resulting rates are tabulated in Table 2.1. We see that cyclic polar codes achieve higher rates at much smaller blocklengths than equivalent rate binary polar codes. As a more fair comparison, let us consider the cyclic polar code of blocklength $N = 1023$ over GF(1024) and the binary polar code of blocklength $N = 2^{16} = 65536$ over GF(2) (or GF(65537), if we ignore complexity comparisons). The equivalent binary blocklength for the length-1023 code would be $N = 10 \cdot 1023 = 10230$ bits. So, the cyclic polar code of length $N = 10230$ bits can achieve a rate almost equal to that of a binary polar code with length $N = 65536$ bits which is more than 6 times higher. This shows that this construction allows us to achieve the capacity of the erasure channel, in this case 0.5, at much smaller blocklengths than binary polar codes, with the only constraint being the complexity introduced by higher field size. Also, it is to be noted that the channel for cyclic polar codes is assumed to introduce symbol erasures rather than bit erasures.

Experiments also show that the order of the factorization of N affects the code rate. For $q = 1024$, $N = 1023$, and $\delta = 0.1$, the order [31 3 11] results in a rate of 0.4340 while the order [3 11 31] gives a rate of 0.4291. This is the reason for multiple rates for some blocklengths in the table. We note that [3 11 31] implies that the length-3 blocks are close to the channel.

A standard polar code with $N = 256$ over $\mathbb{F}_{257}$ was designed for the QEC(0.5) to achieve $P_B \leq \delta = 0.1$. The code was simulated for channels with erasure probabilities $\epsilon = 0.1, 0.2, 0.3, \ldots, 1$ and the block erasure rate for each ϵ (averaged over $M = 1000$ blocks) is plotted in Fig. 2.7.

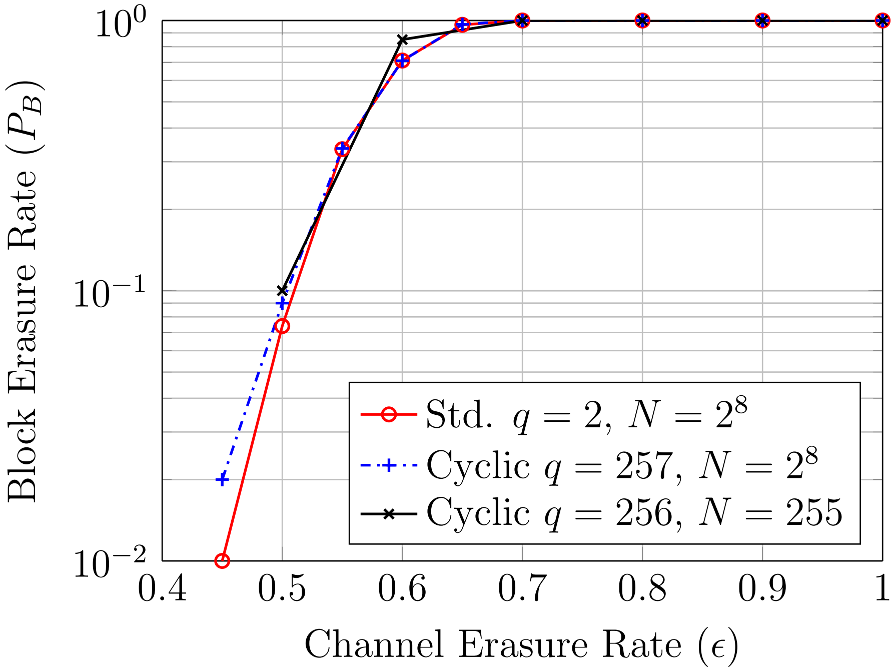


Figure 2.7: Comparison of performance of standard polar and cyclic polar codes on QEC(ϵ). Design parameters were: $\delta = 0.1$; $\epsilon = 0.5$. $R = 0.328$ and $R = 0.384$ for $N = 256$ and $N = 255$, respectively. There were no block erasures observed for $N = 256$ and $N = 255$ at $\epsilon \leq 0.4$ over 1000 blocks each and 100 blocks each, respectively.

Blocklength N	**Rate R**
$2^3 = 8$	0.125
12	0.25
13	0.3077
14	0.2857
$2^4 = 16$	0.25
30	$0.2667, 0.3$
60	$0.2833, 0.3, 0.3167$
$2^6 = 64$	0.2812
255	$0.3843, 0.3882, 0.3922, 0.3961$
$2^8 = 256$	0.3281
1023	$0.4291, 0.4340$
$2^{16} = 65536$	0.4397

Table 2.1: Rates achieved by Binary and Cyclic Polar Codes when designed over BEC(0.5) and QEC(0.5), respectively, for $P_B \leq \delta = 0.1$. The entries for cyclic polar codes are highlighted.

A cyclic polar code of blocklength $N = 2^8$ over $\mathbb{F}_{257}$ (i.e., $\ell_1 = \cdots = \ell_8 = 2$) was designed for the same parameters and the results are plotted in the same figure. The theory suggests that the performance of these two codes should be identical and simulations support this conclusion. Simulation results for an $N = 255 = 3 \cdot 5 \cdot 17$ cyclic polar code over $\mathbb{F}_{256}$ are also presented (averaged over $M = 100$ blocks) and the performance validates polarization for our proposed construction. The code has a rate of 0.384 compared to 0.328 for $N = 2^8$, for the same design parameters.

Cyclic polar codes with $N = 2^n$ over $\mathbb{F}_q$, q prime, were designed over QEC(0.5) and tested on QSC(β) using the soft-decision decoder discussed in Section 2.4.2. The design parameters were chosen to be the same as that of the simulation discussed previously. The results obtained for $N = 256$ and $N = 16$ are averaged over $M = 1000$ blocks and shown in Fig. 2.8. Similarly, a cyclic polar code with $N = 255$ over $\mathbb{F}_{256}$ was constructed for QEC(0.5). It was tested with the hard-decision decoder discussed in 2.4.4 on QSC(β), and the results (averaged over $M = 100$ blocks) are shown in the same figure.

For comparison, a RS code of rate $R \approx 0.328$ can correct a fraction $(1 - R)/2 \approx 0.336$ errors and the Shannon limit (i.e., maximum error rate) of the QSC for rate 0.328 is roughly 0.548. Similarly, the limit for rate 0.384 is roughly 0.491. The theoretical curves for RS codes of the same rates are also plotted for comparison. It is evident that, on the QSC, the cyclic polar code with a soft decoder clearly outperforms an RS code of same rate (cases $N = 256$ and $N = 16$). The cyclic polar code with hard decision decoding does not outperform the comparable RS code. However, designing the cyclic polar code for hard decision decoding may change this.

While a RS code has only one chance to correct all errors and erasures, the cyclic polar code construction can exploit the depth in the graph to convert errors into erasures and also leverage polarization to provide multiple chances and perform significantly better. These results show that it may not be trivial to exhibit this theoretical advantage.

Also, the pattern of errors and erasures at the outputs can have significant effect on code performance. The number of blocks at the output stage that get affected by errors and/or erasures should be minimum for better performance. But, there is an underlying shuffling of indices at the output. Hence, burst errors/erasures will weaken the code as consecutive erroneous indices affect multiple output blocks and the decoder may not have enough information to perform hard-decision decoding at all stages of the decoding process. It might be preferable to transmit the codeword in the shuffled format if the underlying channel is bursty.

It is interesting to note that, while the decoding of all blocks in the graph is identical to that of RS codes, the cyclic polar code itself is not a RS code because, in

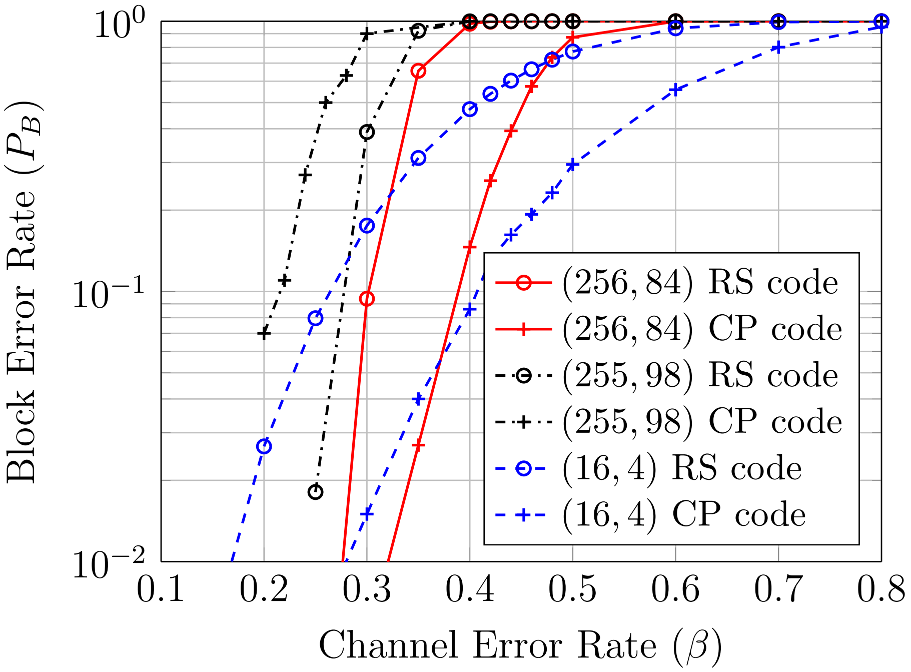


Figure 2.8: Performance of QEC-designed cyclic polar (CP) codes on QSC(β). The design parameters $\delta = 0.1$; $\epsilon = 0.5$ resulted in code rates 0.328, 0.384 and 0.25 for $N = 256$, $N = 255$ and $N = 16$, respectively. No block errors were observed for $N = 256$ and $N = 255$ at $\beta \leq 0.2$ over 1000 blocks each and $\beta \leq 0.18$ over 100 blocks each, respectively. The theoretical performance of RS codes is also plotted for comparison.

general, the design process does not choose a consecutive set of indices for the frozen symbols. Our cyclic polar codes are always subcodes of a (possibly trivial) RS code though. For example, the code in Fig. 2.4 has 8 consecutive zeros in its spectrum and, thus, is a subcode of a $(15, 7, 9)$ generalized RS code.

2.7 Conclusion

This work introduces a method to construct cyclic polar codes over $\mathbb{F}_q$ for any blocklength N satisfying $N|(q-1)$. For the QEC, these codes can be decoded efficiently using Forney's algebraic decoder to decode the intermediate blocks. In our simulations, they outperform standard polar codes. For the case of $N = 2^n$, a soft-decision SC decoder was also implemented and tested on the q-ary symmetric channel. Under SC decoding, cyclic polar codes clearly outperform RS codes of the same rate and blocklength.

An algebraic errors and erasures decoding strategy was also considered for the intermediate block codes. Preliminary results show that this approach is suboptimal when compared to hard decision decoding of a RS code with the same rate and blocklength. In future work, we plan to consider APP decoding of the intermediate blocks for small lengths while retaining a hard-decision decoder at larger blocks typically placed close to inputs in the graph. We will also consider the rate of polarization for these codes based on similar work for standard polar codes [13, 20]. The programs developed for this work can be accessed at `https://github.com/nrenga/cyclic_polar`.

The interesting part about working on polar codes is that one has to learn its literature with the same strategy as its own successive-cancellation decoder – going back and forth in building knowledge slowly, but steadily, towards capacity. Perhaps, this is truly an optimal strategy for learning, in general.

3. SPATIALLY-COUPLED LDPC CODES

3.1 Introduction

Low-density parity-check (LDPC) codes are widely used due to their outstanding performance under low-complexity belief propagation (BP) decoding. However, an error probability exceeding that of maximum-a-posteriori (MAP) decoding has to be tolerated with (sub-optimal) BP decoding. Lately, it has been empirically observed for spatially coupled LDPC (SC-LDPC) codes—first introduced by Felström and Zigangirov as convolutional LDPC codes [8]—that the BP performance of these codes can improve dramatically towards the MAP performance of the underlying LDPC code under many different settings and conditions, e.g. [16]. This phenomenon, termed *threshold saturation*, has been proven rigorously in [14, 15]. In particular, the BP threshold of a coupled LDPC ensemble tends to its MAP threshold on any binary memoryless symmetric (BMS) channel.

Besides their excellent performance on the BEC and AWGN channels, much less is known about the burst error correctability of SC-LDPC codes. In [12], the authors consider SC-LDPC ensembles over a block erasure channel (BLEC) where the channel erases complete spatial positions instead of individual bits. This block erasure model mimics block-fading channels frequently occurring in wireless communications. The authors give asymptotic lower and upper bounds for the bit and block erasure probabilities obtained from density evolution. In [10], the authors construct protograph-based codes that maximize the correctable burst lengths, while the authors in [17] apply interleaving (therein denoted band splitting) to a protograph-based SC-LDPC code to increase the correctable burst length. If windowed decoding is used, this approach results in an increased required window length and thus complexity. Recently, it has been shown that protograph-based LDPC codes can increase the diversity order of block fading channels and are thus good candidates for block erasure channels [29],[28]; however, they require large syndrome former memories if the burst length becomes large.

In this work, we consider the (d_v, d_c, w, L, M) code ensemble introduced in [14] and derive tight lower bounds on the correctability of a long burst of erasures. First, we consider the case when a complete spatial position is erased and then generalize the expression to the case where the burst can occur at any position within a codeword. We show that estimating the capability of correcting long burst erasures reduces to the problem of finding small stopping sets in the code structure. Also, we demonstrate that if we properly expurgate the ensemble, then a random code from the ensemble has very good average burst erasure capabilities. We focus on the general (d_v, d_c, w, L, M) code ensemble as the common protograph-based approach contains unavoidable small stopping sets in each spatial position, which are not recoverable if erased [21].

The discussion is organized as follows: Section 3.2 reviews essential technical background, Sections 3.3 and 3.4 provide finite-length analysis of the random ensemble on burst erasure channels, Section 3.5 gives the error floor for the ensemble on the BEC, Section 3.6 details the effects of expurgating the ensemble, Section 3.7 compares ensembles and highlights important observations from this work, and Section 3.8 concludes the work mentioning potential problems for future research.

3.2 Preliminaries

3.2.1 The Regular (d_v, d_c, w, L, M) SC-LDPC Ensemble

We now briefly review how to sample a code from a random regular (d_v, d_c, w, L, M) SC-LDPC ensemble [14]. We first lay out a set of positions indexed from $z = 1$ to

L on a *spatial dimension.* At each spatial position (SP), z, there are M variable nodes (VNs) and $M\frac{d_v}{d_c}$ check nodes (CNs), where $M\frac{d_v}{d_c} \in \mathbb{N}$ and d_v and d_c denote the variable and check node degrees, respectively. Let $w > 1$ denote the smoothing (coupling) parameter. Then, we additionally consider $w-1$ sets of $M\frac{d_v}{d_c}$ CNs in SPs $L+1, \ldots, L+w-1$. Every CN is assigned with d_c "sockets" and made to impose an even parity constraint on its d_c neighboring VNs. Each VN in SP z is connected to d_v CNs in SPs $z, \ldots, z+w-1$ as follows: each of the d_v edges of this VN is allowed to randomly and uniformly connect to any of the wMd_v sockets arising from the CNs in SPs $z, \ldots, z+w-1$, such that multiple edges are avoided in the resultant bipartite graph. This graph represents the code so that we have $N = LM$ code bits, over L SPs. Because of additional check nodes in SPs $z > L$, the code rate $r = 1 - \frac{d_v}{d_c} - \delta$, where $\delta = O(\frac{w}{L})$. Fig. 3.1 gives a pictorial depiction of this ensemble. Throughout this work, we assume that $d_v \geq 3$ and $wM > 2(d_v+1)d_c$.

Let us define *constellation* and *type* for each VN as introduced in [14]. Again, consider a VN in SP i. Assume that the d_v edges are indexed by $k \in \{1, 2, \ldots, d_v\}$. We define an associated d_v-tuple vector, called its *constellation*, as $c = (c_1, c_2, \ldots, c_{d_v})$ where $c_k \in \{0, 1, \ldots, w-1\}$ and the k^{th} edge connects to a CN at position $i + c_k$. Clearly, there are w^{d_v} constellations. We define an associated *type* vector $t = (t_0, t_1, \ldots, t_{w-1})$ where t_j indicates the number of edges of this VN that connect to a CN in position $i+j$. Hence $\sum_{j=0}^{w-1} t_j = d_v$ and there are $\binom{d_v+w-1}{w-1}$ types.

Note that there exists a many-to-one mapping between constellations and types. In our random ensemble, all constellations are possible while more structured ensembles might have only few constellations that are allowed. We impose a uniform distribution on the set of all constellations and, owing to the many-to-one mapping, this introduces a distribution on the set of all types. Let $\tau(c)$ denote the type of a constellation. Then, the distribution on the types can be expressed using the

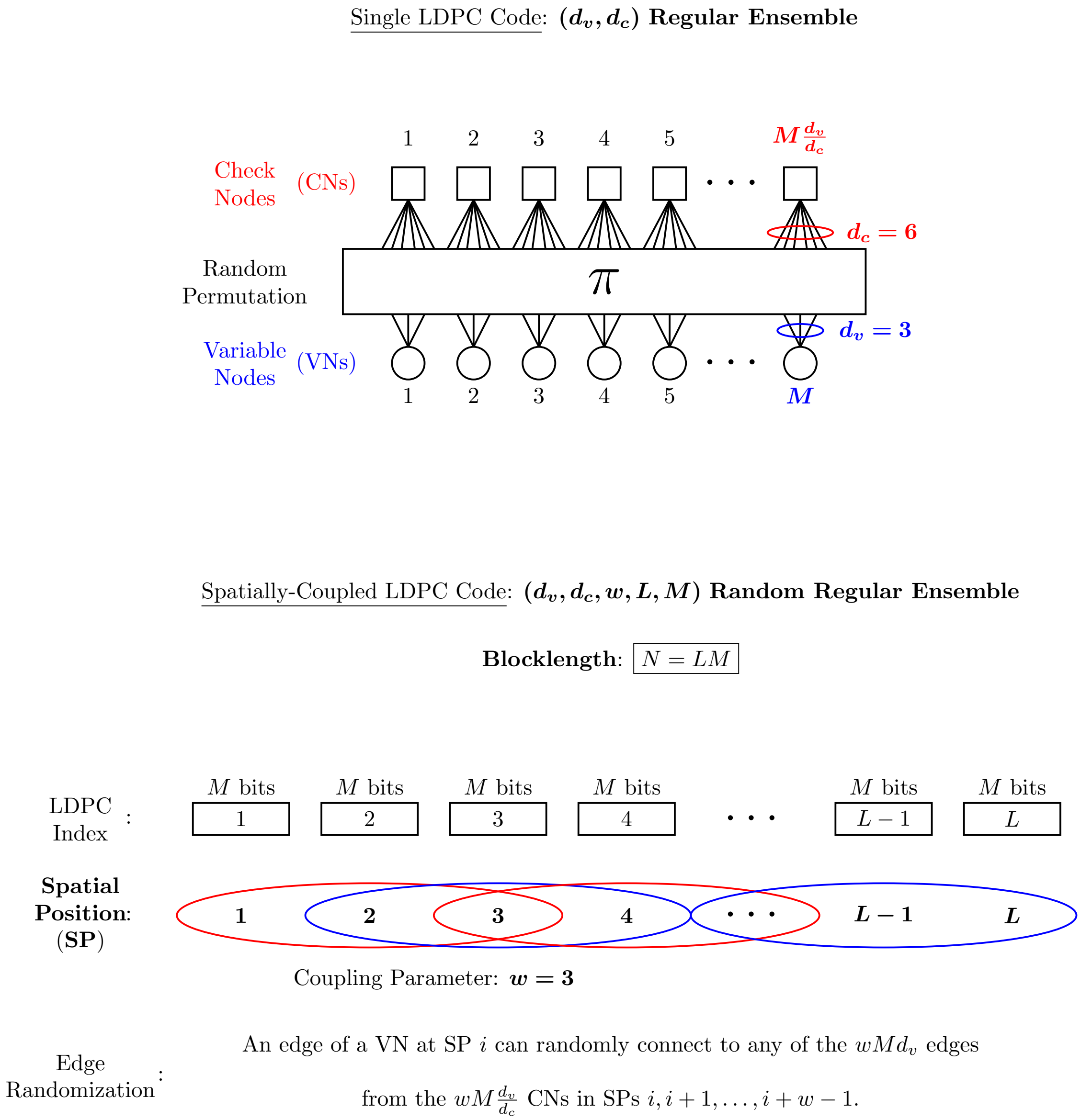

Single LDPC Code: (d_v, d_c) Regular Ensemble
1
2
3
4
5
$M\frac{d_v}{d_c}$
Check Nodes (CNs)
$d_c = 6$
Random Permutation
π
$d_v = 3$
Variable Nodes (VNs)
1
2
3
4
5
M
Spatially-Coupled LDPC Code: (d_v, d_c, w, L, M) Random Regular Ensemble
Blocklength: $\boxed{N = LM}$
LDPC Index :
M bits
1
2
3
4
$L-1$
L
Spatial Position: (SP)
1
2
3
4
$L-1$
L
Coupling Parameter: $w = 3$
Edge Randomization :
An edge of a VN at SP i can randomly connect to any of the wMd_v edges
from the $wM\frac{d_v}{d_c}$ CNs in SPs $i, i+1, \ldots, i+w-1$.

Figure 3.1: A depiction of a random regular (d_v, d_c, w, L, M) SC-LDPC ensemble constructed from a regular (d_v, d_c) LDPC ensemble.

probability of a type,

$$p(t) = \frac{|c : \tau(c) = t|}{w^{d_v}}.$$

3.2.2 Stopping Sets

A subset $\mathcal{A}$ of the set of VNs in a code is a *stopping set* if all the neighboring CNs of (the VNs in) $\mathcal{A}$ connect to $\mathcal{A}$ at least twice [25, Def. 3.137]. In such a case, if all VNs in $\mathcal{A}$ have been erased by the channel, then the peeling decoder will fail as all the neighboring CNs are connected to at least two erased VNs. Therefore, such a set will stop the decoding process and hence is called a *stopping set*. The cardinality of the set $\mathcal{A}$ is also its size. A *minimal stopping set* is one which does not contain a smaller size non-empty stopping set within itself.

3.2.3 Binary Erasure Channel

The symmetric binary erasure channel with parameter ϵ is denoted by BEC(ϵ) and its transition probabilities are defined, for $x \in \{0, 1\}$ and $y \in \{0, 1, ?\}$, by

$$W(y|x) = \begin{cases} 1 - \epsilon & \text{if } y = x \\ \epsilon & \text{if } y = ? \end{cases}$$

Hence, approximately, a fraction ϵ of the transmission (in bits) is erased randomly.

3.2.4 Single-Burst-Erasure Channel Models

We introduce two channel models for computing the burst erasure recoverability. First, the *Single Position Burst Channel* (SPBC) erases all M VNs of exactly one SP in the transmitted codeword and leaves all other bits undisturbed.

The second model is the more general *Random Burst Channel* (RBC) whose burst pattern is denoted by RBC(ℓ,s,b) where $s \in \{1, \ldots, M\}$ is the starting bit index of the burst in SP $\ell \in \{1, \ldots, L\}$, indicating the offset from the first VN of the SP ℓ, and b is the length of the burst. Note that in general $0 < b \leq (L - \ell)M - s$. As for the SPBC, all VNs in the random burst are erased while all other VNs are received correctly. We sometimes omit the SP ℓ when referring to the RBC for the following reason: neglecting boundary effects in the limit of large enough L, all SPs are structured identically. With some abuse of terminology, we will use the same notation to refer to the channel itself, rather than the burst introduced by it.

While multiple models exist for a correlated erasure channel, like the Gilbert-Elliott model used in [10], we use this model because it is sufficient to describe the scenarios that we consider: for instance, the SPBC can be used to model a slotted-ALOHA multiple access scheme where each user transmits an SC-LDPC codeword over L time slots, but one SP might be erased in the case of a collision. Additionally, long burst erasures might occur in block fading scenarios, or in optical communications which are subject to polarization dependent loss.

3.3 Error Analysis on the SPBC

Let $P_{\mathrm{B}}^{\mathrm{SPBC}}(d_v, d_c, w, L, M)$ denote the average block erasure (decoding error) probability of the (d_v, d_c, w, L, M) ensemble on the SPBC under BP decoding i.e. the probability that the iterative decoder fails to recover the codeword. For large enough M, size-2 stopping sets (each of which also form a codeword) are the dominant structures in the graph that cause the BP decoder to fail [21]. Hence, the number of size-2 stopping sets per SP, denoted $\mathbb{N}_2^{\mathrm{SP}}$, is a good starting point for analyzing the performance of the ensemble. We have

$$\begin{aligned}
P_{\mathrm{B}}^{\mathrm{SPBC}} &= \text{Prob [At least one stopping set in a SP]} \\
&\geq \text{Prob } [\mathbb{N}_2^{\mathrm{SP}} \geq 1] \\
&\overset{(a)}{\geq} \frac{\mathbb{E}[\mathbb{N}_2^{\mathrm{SP}}]^2}{\mathbb{E}[{\mathbb{N}_2^{\mathrm{SP}}}^2]} \overset{(b)}{\geq} \mathbb{E}[\mathbb{N}_2^{\mathrm{SP}}]\left(1 - \frac{M^2}{(\frac{w}{d_c}M-3)^{d_v}}\right) \\
&= \mathbb{E}[\mathbb{N}_2^{\mathrm{SP}}]\left(1 - O\left(\frac{1}{M^{d_v-2}}\right)\right) \approx \mathbb{E}[\mathbb{N}_2^{\mathrm{SP}}] \doteq \lambda_{\mathrm{SP}}, \qquad (3.1)
\end{aligned}$$

where (a) is the application of the second moment method and (b) can be shown as follows: Define $U_{ij} = 1$ if VNs i and j form a stopping set, otherwise $U_{ij} = 0$. Then $\mathbb{N}_2^{\mathrm{SP}} = \sum_{1\leq i<j\leq M} U_{ij}$ where the summation is over all $\binom{M}{2}$ pairs of VNs from a SP. We can see that $\lambda_{\mathrm{SP}} = \mathbb{E}[\mathbb{N}_2^{\mathrm{SP}}] = \binom{M}{2}p$, where $p = \mathbb{E}[U_{ij}]$ is the probability of forming a size-2 stopping set.

$$\begin{aligned}
\mathbb{E}\left[{\mathbb{N}_2^{\mathrm{SP}}}^2\right] &= \mathbb{E}\left[\left(\sum_{1\leq i<j\leq M} U_{ij}\right)^2\right] \\
&= \sum_{1\leq i<j\leq M} \mathbb{E}[U_{ij}^2] \quad + \sum_{\substack{i<j,k<l \\ (i,j)\neq(k,l)}} \mathbb{E}[U_{ij}U_{kl}],
\end{aligned}$$

where in the last step, $\sum_{1\leq i<j\leq M} \mathbb{E}[U_{ij}^2] = \binom{M}{2}p$ as $U_{ij} \in \{0,1\}$ and the second term is over the remaining $\binom{M}{2}\left(\binom{M}{2}-1\right)$ combinations. Using some combinatorial arguments, we can show that $\mathbb{E}[U_{ij}U_{kl}] = \mathbb{P}(U_{ij}=1)\mathbb{P}(U_{kl}=1|U_{ij}=1) \leq 2p/\binom{wM\frac{d_v}{d_c}-2d_v}{d_v}$. As a result, we have

$$\begin{aligned}
\mathbb{E}\left[{\mathbb{N}_2^{\mathrm{SP}}}^2\right] &< \mathbb{E}[\mathbb{N}_2^{\mathrm{SP}}]\left(1 + \frac{2\binom{M}{2}}{\binom{wM\frac{d_v}{d_c}-2d_v}{d_v}}\right) \\
&< \mathbb{E}[\mathbb{N}_2^{\mathrm{SP}}]\left(1 + \frac{M^2}{(\frac{w}{d_c}M-3)^{d_v}}\right),
\end{aligned}$$

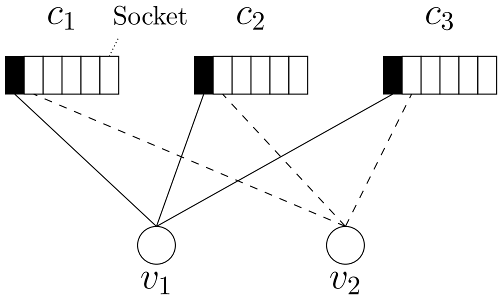


Figure 3.2: A size-2 stopping set from a $(3, 6)$ random ensemble. CNs $\{c_1, c_2, c_3\}$ and VNs $\{v_1, v_2\}$ have been labeled for convenience. CNs have been expanded to show all their $d_c = 6$ sockets. The solid edges indicate definite connections and the dashed edges complete one configuration to form a stopping set. Multiple edges are not allowed in the ensemble.

which eventually implies (3.1). Note that following standard arguments [21], [25, Appendix C], we can also approximate the bound on $P_{\text{B}}^{\text{SPBC}}$ by a Poisson distribution with mean λ_{SP}, for a large M, so that

$$P_{\text{B}}^{\text{SPBC}} \approx 1 - e^{-\lambda_{\text{SP}}} \approx \lambda_{\text{SP}}. \tag{3.2}$$

Both (3.1) and (3.2) are very tight when $w \geq d_v$ (which is a prerequisite for constructing capacity-achieving codes [14]) as otherwise, we have observed that the contribution of larger stopping sets becomes non-negligible. We use this observation later in Section 3.5 to characterize the number of size-2 stopping sets in the code, $\mathbb{N}_2^H$ (instead of one SP).

3.3.1 Calculation of p

We now calculate the probability p of finding a size-2 stopping set within an SP of a code uniformly sampled from an ensemble. As example, we randomly choose two VNs v_1 and v_2 from an SP of the $(d_v = 3, d_v = 6, w, L, M)$ ensemble. First, we connect the $d_v = 3$ edges of v_1 to randomly chosen empty sockets of d_v distinct CNs as described in Section 3.2.1. Let c_1, c_2, c_3 denote the CNs adjacent to v_1. A stopping set (and in this case, also a low-weight codeword) is formed if and only if the edges of v_2 are connected to the same CNs, i.e. c_1, c_2, c_3. This situation is shown in Fig. 3.2: once we have assigned d_v CNs to v_1, we have $d_c - 1 = 5$ free distinct sockets each for CNs c_1, c_2, c_3. Thus, the first edge of v_2 has $d_v(d_c - 1) = 15$ ways to attach to these sockets, the second edge has $(d_v - 1)(d_c - 1) = 10$ ways and the last edge has $(d_v - 2)(d_c - 1) = 5$ ways. In general, the edges of v_2 can be connected to any of the $(wMd_v - d_v)$ possible sockets.

By a counting argument, we can compute $p = \frac{T_{ss}}{T}$ where T_{ss} is the total number of combinations by which the edges of v_2 can form a stopping set with v_1 and T is the total number of combinations by which the edges of v_2 can be fit to the possible CN sockets without forming multiple edges. We have

$$\begin{aligned} T_{ss} = & \ 15 \times 10 \times 5, \\ T = & \ 15 \times 10 \times 5 \\ & + [15 \times 10 \times (3wM - 18)] \times 3 \\ & + [15 \times (3wM - 18) \times (3wM - 24)] \times 3 \\ & +(3wM - 18) \times (3wM - 24) \times (3wM - 30) \end{aligned}$$

that give

$$p \approx \frac{15 \times 10 \times 5}{(3wM - 18)(3wM - 24)(3wM - 30)}.$$

Hence, for a general (d_v, d_c, w, M) ensemble we can calculate $p = \frac{T_{ss}}{T}$ with

$$T_{ss} = \prod_{i=0}^{d_v-1} (d_v - i)(d_c - 1) = d_v!(d_c-1)^{d_v},$$

$$T = \sum_{i=0}^{d_v} \frac{(d_c-1)^i d_v!}{(d_v-i)!} \binom{d_v}{i} \left[\prod_{k=0}^{d_v-1-i} (wMd_v - (d_v+k)d_c) \right].$$

For large M, T can be well approximated by the dominating summand ($i = 0$) leading to

$$p \approx \prod_{i=0}^{d_v-1} \frac{(d_v-i)(d_c-1)}{(wMd_v - (d_v+i)d_c)} \approx \frac{d_v!(d_c-1)^{d_v}}{((wM-d_c)d_v)^{d_v}}. \tag{3.3}$$

We observe that $\lambda_{SP} = \binom{M}{2} p \sim O(M^{2-d_v})$.

3.3.1.1 Poisson Ensemble

We make note of a significant change to (3.3) when this random ensemble is slightly relaxed. Retaining the construction of the random ensemble, if there is no limit placed on the check degree then we get the so-called Poisson ensemble $\mathcal{C}_P$. For this ensemble, sockets are not distinct and therefore the calculation of p is much simpler. Let v_1 and v_2 belong to SP i. Assume that the edges of v_2 are assigned to CNs sequentially. The first edge can connect to any of the $(wM\frac{d_v}{d_c})$ CNs from SPs $i, i+1, \ldots, i+w-1$. The second edge has one CN less to choose from, the third edge has two CNs less to choose from and so on. But, there is exactly one way in which the edges can connect exactly to the same CNs as v_1. Hence the probability of v_2 forming a stopping set with v_1 is

$$p' = \left[\binom{wM\frac{d_v}{d_c}}{d_v} \right]^{-1}. \tag{3.4}$$

Let us compare this with (3.3). First, we rewrite p as

$$p \cong \left(\frac{d_c - 1}{d_c}\right)^{d_v} \times \frac{d_v!}{\prod_{i=0}^{d_v-1}\left(wM\frac{d_v}{d_c} - (d_v + i)\right)}.$$

Then, we can compare the two ensembles as below.

$$\begin{aligned} & p' - p \\ = & \frac{d_v!}{\prod_{i=0}^{d_v-1}\left(wM\frac{d_v}{d_c} - i\right)} - \left(\frac{d_c - 1}{d_c}\right)^{d_v} \frac{d_v!}{\prod_{i=0}^{d_v-1}\left(wM\frac{d_v}{d_c} - (d_v + i)\right)} \\ = & \frac{d_v!}{\prod_{i=0}^{2d_v-1}\left(wM\frac{d_v}{d_c} - i\right)} \times \prod_{i=0}^{d_v-1}\left[\left(wM\frac{d_v}{d_c} - (d_v + i)\right) - \left(\frac{d_c - 1}{d_c}\right)\left(wM\frac{d_v}{d_c} - i\right)\right]. \end{aligned}$$

Now, analyze the individual product terms as

$$a' = \left(wM\frac{d_v}{d_c} - i\right) - d_v \text{ and } a = \left(\frac{d_c - 1}{d_c}\right)\left(wM\frac{d_v}{d_c} - i\right),$$

where $i = 0, 1, \ldots, d_v - 1$. We immediately see that

$$wM\frac{d_v}{d_c} - i \geq d_v d_c \Rightarrow a' \geq a.$$

Evaluating this condition for the worst case, $i = d_v - 1$, we can conclude that

$$M \geq \frac{(d_v d_c + d_v - 1)d_c}{wd_v} \Rightarrow p' - p \geq 0.$$

Hence, the Poisson ensemble performs worse than the random ensemble under this condition.

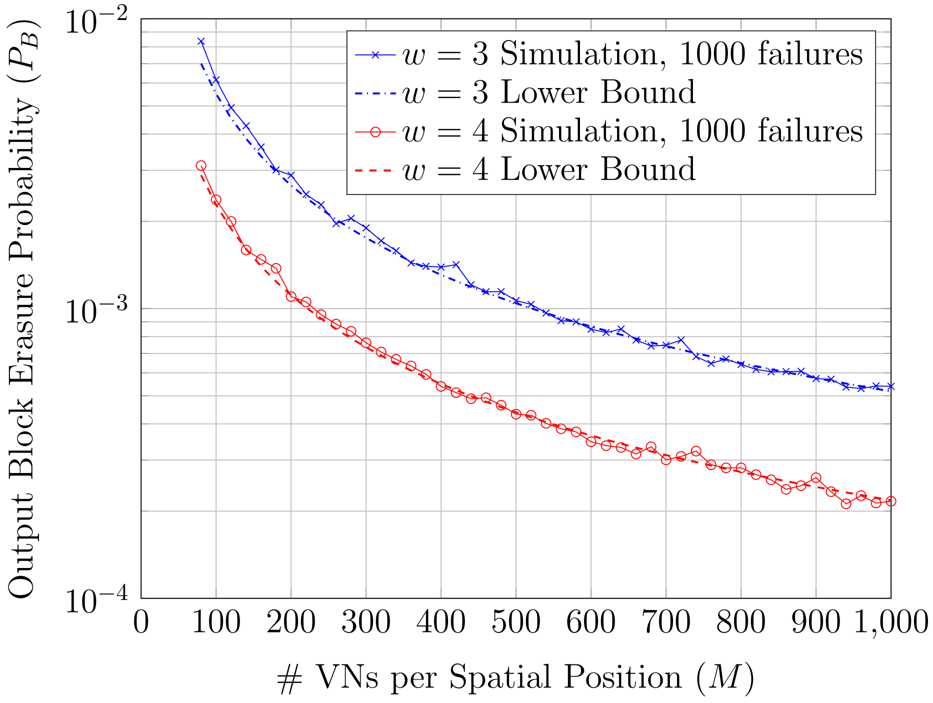


Figure 3.3: Monte Carlo simulations on the SPBC with a $(3, 6)$ random ensemble for $w = 3$ and $w = 4$, along with their respective theoretical lower bound (3.1). The bound becomes tight very quickly with M.

3.3.2 Simulations

We performed Monte-Carlo simulations where we randomly selected a spatial position from the middle of the graph (to avoid boundary effects) to be erased, for each transmitted codeword. At the receiver we performed BP decoding and averaged over the ensemble. We counted 1000 decoding failures for each M to assess the average block erasure probability $P_{\mathrm{B}}^{\mathrm{SPBC}}$. The simulation results for a $(3, 6)$ random ensemble with $w = 3$ and $w = 4$ are shown in Fig. 3.3 along with their respective lower bounds calculated using (3.1) and (3.3). We observe that the bound indeed becomes very good for large M, since large-size stopping sets (larger than 2) vanish. The simulation curve is slightly unstable because counting 1000 failures is not enough to keep the sample variance small as $P_{\mathrm{B}}^{\mathrm{SPBC}}$ decreases by $O(M^{2-d_v})$.

We performed the above experiment again for a $d_v = 3, w = 3$ Poisson ensemble and plotted the results in Fig. 3.4 along with the lower bound calculated using (3.4) as p in (3.2). For comparison, we have also plotted the lower bound from the previous simulation for a $(3, 6, w = 3)$ random ensemble. As noted in Section 3.3.1.1, the Poisson ensemble performs worse than the equivalent random ensemble since our condition for this case is $M \geq 14$. Moreover, we performed simulations for the same scenario with the random ensemble by allowing multiple edges in the graph and plotted the results in the same figure. We see that multiple edges degrade the performance even more. This gives a complete picture of the relative performances of these ensembles on the SPBC.

3.4 Error Analysis on the RBC

We now generalize our results to the RBC, where a burst can span multiple spatial positions and can be of arbitrary length. Besides the stopping sets within a single spatial position, we first have to derive an expression for stopping sets that span

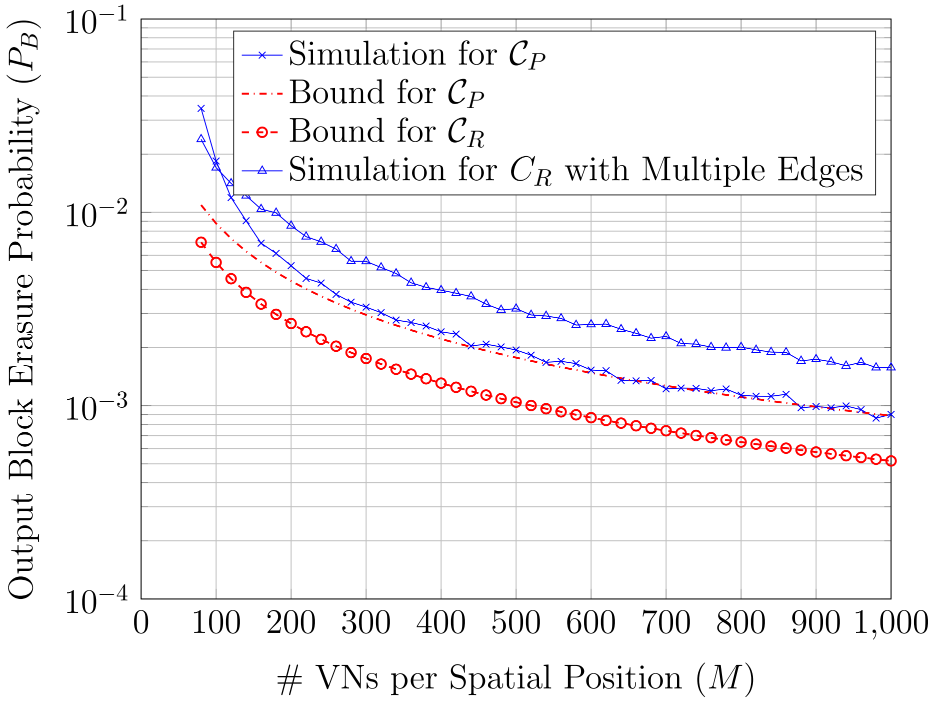


Figure 3.4: Monte Carlo simulations on the SPBC with the Poisson ensemble $\mathcal{C}_P$ for $d_v = 3, w = 3$, along with the theoretical lower bound calculated using (3.4) as p in (3.2). The bound for a $(3, 6, w = 3)$ random ensemble $\mathcal{C}_R$ and the simulation results for $\mathcal{C}_R$ with multiple edges, for the same scenario, is also plotted for comparison.

multiple SPs.

3.4.1 Size-2 Stopping Sets across Coupled SPs

The results from Section 3.3 can be extended when the channel is a RBC, i.e., the burst occurs at arbitrary location and is of arbitrary length. This means that size-2 stopping sets formed across coupled SPs will also contribute to decoding failures. Hence, we will now calculate the probability that two VNs chosen each from two coupled spatial positions form a stopping set.

Let us first consider two VNs chosen from two adjacent SPs: w.l.o.g, call them v_1 and v_2 chosen from SPs 1 and 2, respectively. We immediately notice that the check positions adjacent to v_1 are $1, 2, \ldots, w$ and to v_2 are $2, 3, \ldots, w+1$. Hence, to form a stopping set, v_1 should not have any edge connected to check position 1. This restricts the number of favorable constellations [14] for v_1 to be $(w-1)^{d_v}$. Using the same ideas as in Section 3.3.1 and restricting the constellations for v_1, we have

$$p_{(1,2)} = \frac{(w-1)^{d_v}}{w^{d_v}} p,$$

where p can be approximated by (3.3). This idea can now be extended to VNs chosen from positions $(1,3), (1,4), \ldots, (1,w)$ by restricting the number of favorable constellations for v_1. Hereafter, we will refer to these as size-2 $(1,i)$-stopping sets. Hence, a (d_v, d_c, w, L, M) ensemble can be completely characterized on erasure channels, for large enough M, by the vector

$$\begin{aligned} \underline{p}(d_v, d_c, w, L, M) &= (p_{(1,1)}, p_{(1,2)}, \ldots, p_{(1,w)}) \\ \text{with } p_{(1,i)} &= \left(\frac{w-(i-1)}{w}\right)^{d_v} p. \end{aligned} \tag{3.5}$$

The average number of size-2 stopping sets of each type, $\lambda_{(1,i)}$, can be calculated as

$$\lambda_{(1,1)} = \binom{M}{2} p_{(1,1)} = \lambda_{SP} \;\; ; \;\; \lambda_{(1,i)} = M^2 p_{(1,i)}, \tag{3.6}$$

where $i = 2, 3, \ldots, w$. Again, we see that $\lambda_{(1,i)} \sim O(M^{2-d_v})$.

3.4.2 Performance on the RBC

Now let us see the effect of RBC(s, b) on the ensemble in terms of the average block erasure probability, $P_{\text{B}}^{\text{RBC}}$. For keeping the expressions simple, let us assume in the example that $w = 3$ and $0 < b \leq 2M$. This means that the burst can span a maximum of 3 SPs. Applying the same argument as in Section 3.3 and assuming all values for s are equally likely,

$$\begin{aligned} P_{\text{B}}^{\text{RBC}} &\approx \sum_{s=1}^{M} \frac{1 - P_{(1,1)}P_{(2,2)}P_{(3,3)}P_{(1,2)}P_{(2,3)}P_{(1,3)}}{M} \;\; ; \\ P_{(k,k)} &= 1 - \binom{m_k}{2} p_{(1,1)} \quad \text{for } k = 1, 2, 3, \\ P_{(k,k+1)} &= 1 - m_k m_{k+1} p_{(1,2)} \;\; \text{for } k = 1, 2, \\ P_{(k,k+2)} &= 1 - m_k m_{k+2} p_{(1,3)} \;\; \text{for } k = 1, \end{aligned} \tag{3.7}$$

where $m_1 = (M - s), m_2 = \min(b - m_1, M), m_3 = (b - m_1 - m_2)$ are the lengths of the burst in each SP that it affects, progressing from left to right. If any of these lengths is zero, all probabilities involving that length are 1, i.e., the probability of forming no size-2 stopping sets involving the SP corresponding to this (zero) length is 1. For general w and longer bursts, this strategy can be extended for finding a very good approximation for the average block erasure probability for the ensemble.

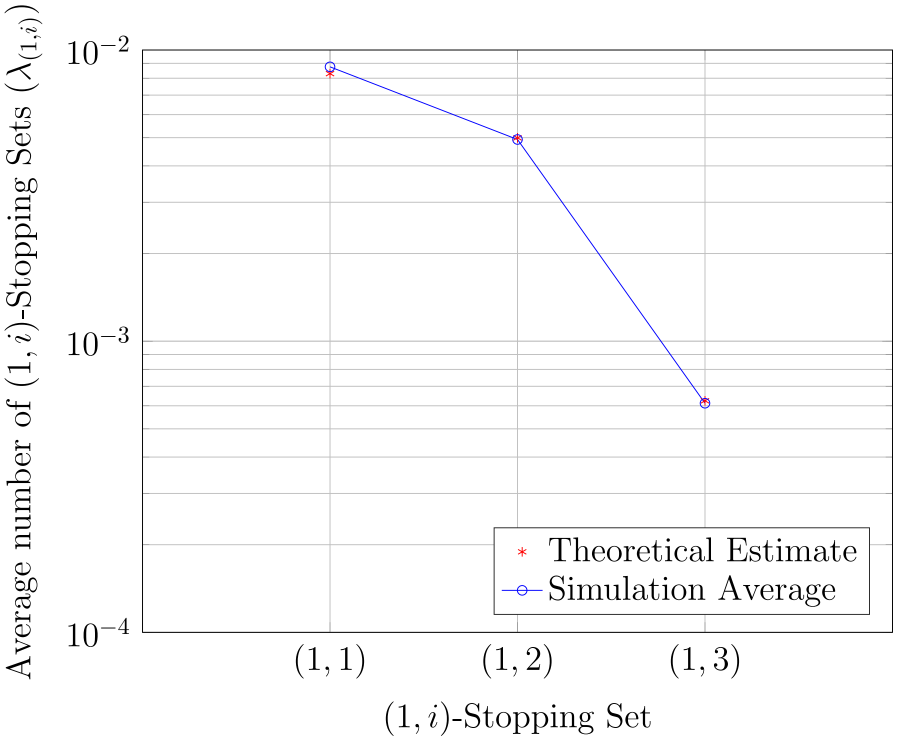


Figure 3.5: Average number of size-2 $(1,i)$-stopping sets in a code from the random $(3,6,3,100,64)$ ensemble, along with theoretical estimates calculated using (3.8).

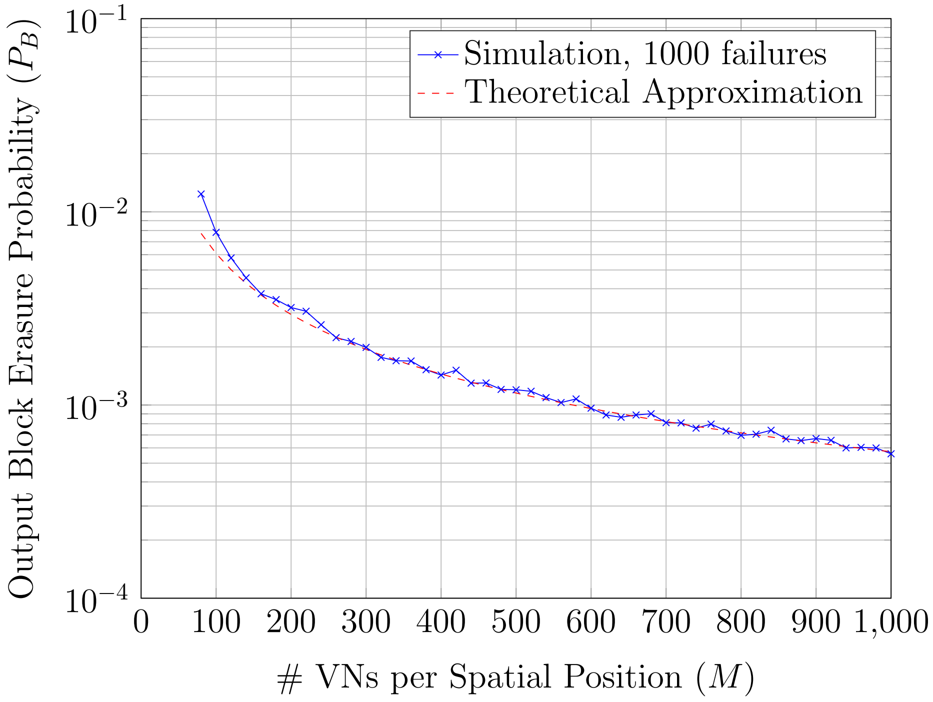


Figure 3.6: Monte Carlo simulations for a $(3, 6, 3, 20, M)$ random ensemble on the RBC with burst length $b = 1.25M$, along with the theoretical approximation (3.7).

3.4.3 Simulations

First, we show that the individual components of $\underline{p}(d_v, d_c, w, L, M)$ given in (3.5) are accurate for even a small value of $M = 64$. With $w = 3$, $L = 100$ for a $(3, 6)$ random ensemble, we estimate the average number of size-2 $(1, i)$-stopping sets corresponding to each component of $\underline{p}$ by averaging over all the SPs of 1000 codes sampled from the ensemble. The experimental histogram and the theoretical averages

$$(\lambda_{(1,1)}, \lambda_{(1,2)}, \lambda_{(1,3)}) = \left(\binom{M}{2} p_{(1,1)}, M^2 p_{(1,2)}, M^2 p_{(1,3)}\right) \tag{3.8}$$

are plotted in Fig. 3.5.

To verify the tightness of (3.7), we again performed Monte-Carlo simulations and counted 1000 decoding failures for each M to assess the average block erasure probability $P_{\mathrm{B}}^{\mathrm{RBC}}$. For the sake of example, we fixed the burst length to be $b = 1.25M$. We selected a value for s, uniformly from $\{1, \ldots, M\}$, for each codeword. The simulation results for the $(3, 6, 3, 20, M)$ ensemble are shown in Fig. 3.6 along with (3.7). We see that (3.7) is indeed a tight approximation.

3.5 Error Floor on BEC

3.5.1 Distribution of $\mathbb{N}_2^H$

The approach described here is from [21]. We know that stopping sets of size larger than 2 vanish for large enough M. This means that with random erasures on BEC(ϵ), size-2 stopping sets in the code are, with high probability, the cause of decoder failures. As mentioned earlier, the ensemble is completely characterized by the vector $\underline{p}(d_v, d_c, w, L, M)$, as given in (3.5). Using these we also know the average number of size-2 stopping sets of each type in the code, which has been expressed in (3.6). Therefore, the average number of size-2 stopping sets in a code is

$$\lambda = \mathbb{E}[\mathbb{N}_2^H] = L\left[\lambda_{(1,1)} + \sum_{i=2}^{w} \lambda_{(1,i)}\right].$$

and we observe that $\lambda \propto LM^{2-d_v}$. More carefully, if we take into account the boundary effects, we calculate this as

$$\lambda = \mathbb{E}[\mathbb{N}_2^H] = L\lambda_{(1,1)} + (L-w+1)\sum_{i=2}^{w} \lambda_{(1,i)} + \sum_{j=1}^{w-2}\sum_{i=j+1}^{w-1} \lambda_{(1,w-i+1)}. \tag{3.9}$$

Since each of the $(1,i)$-stopping sets form a Poisson distribution with mean $\lambda_{(1,i)}$ and the correlation between them is negligible, we conjecture that $\mathbb{N}_2^H \sim \text{Poisson}(\lambda)$.

Given that size-2 stopping sets are dominantly responsible for decoder failures on BEC(ϵ), the expected error floor for a (d_v, d_c, w, L, M) random ensemble is given by

$$P_{\text{b}}(d_v, d_c, w, L, M) = \frac{2\lambda\epsilon^2}{LM}, \tag{3.10}$$

where P_{b} is the average bit error rate for the ensemble.

3.5.2 Simulations

First, we show in Fig. 3.7 that $\mathbb{N}_2^H \sim \text{Poisson}(\lambda)$ through the simulation histogram averaged over 10^5 code blocks from the $(3,6,3,10,64)$ random ensemble. Equation (3.9) is used to calculate the theoretical Poisson distribution.

We verified the error floor calculation through standard simulations on the BEC for a $(3,6,3,L,M)$ random ensemble with $M = 128, 256, 512$ and $L = M/2$. The results and the predicted error floor (3.10) are plotted in Fig. 3.8. In comparison with the observations in [21], the error floor of the random ensemble seems to be slightly worse than the protograph-based ensemble since the latter is more structured.

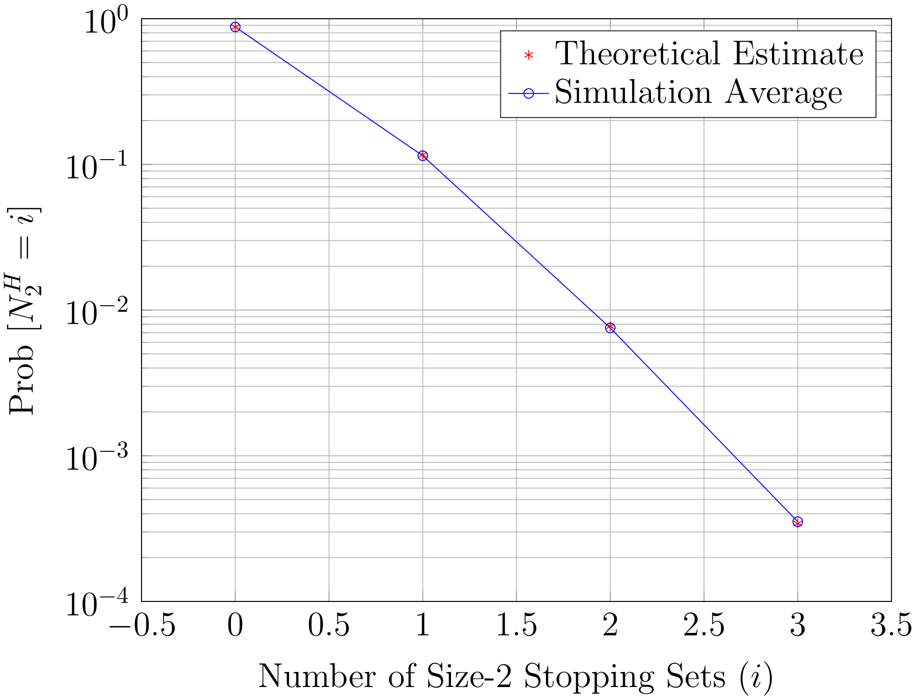


Figure 3.7: Poisson distribution of $\mathbb{N}_2^H$ for a $(3, 6, 3, 10, 64)$ ensemble, with theoretical estimates calculated using (3.9).

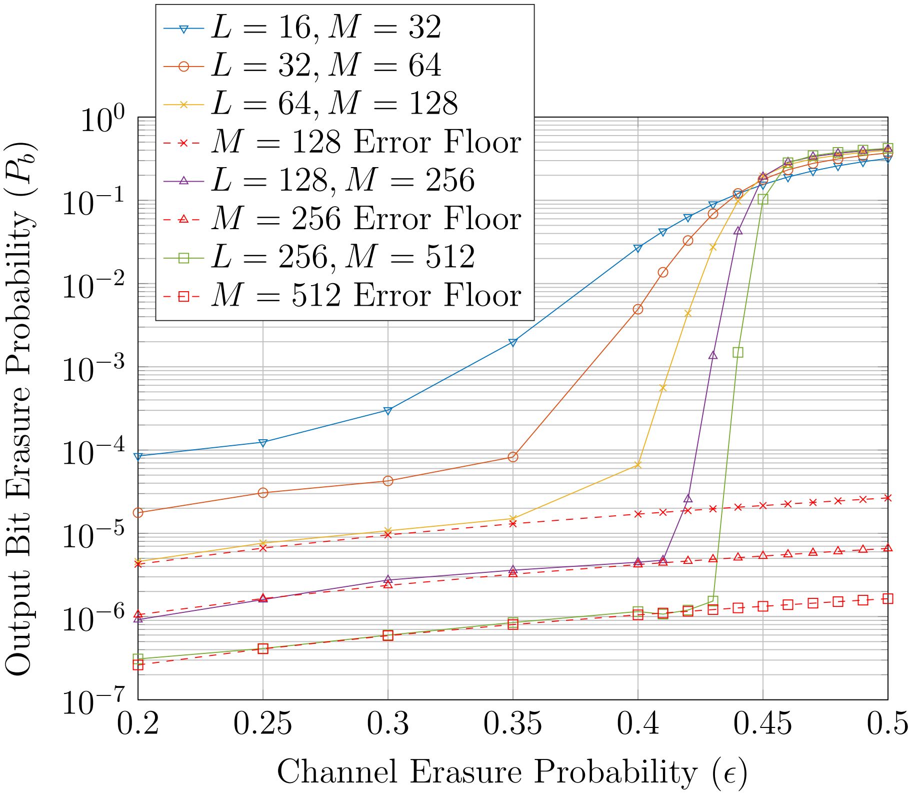


Figure 3.8: The expected error floors for a $(3, 6, 3, L, M)$ random ensemble on the BEC.

Also, it is worth noting again that the performance of a typical code from the random ensemble is concentrated around the ensemble average and hence, this is the expected behavior for a code uniformly sampled from this ensemble.

3.6 Effects of Expurgation

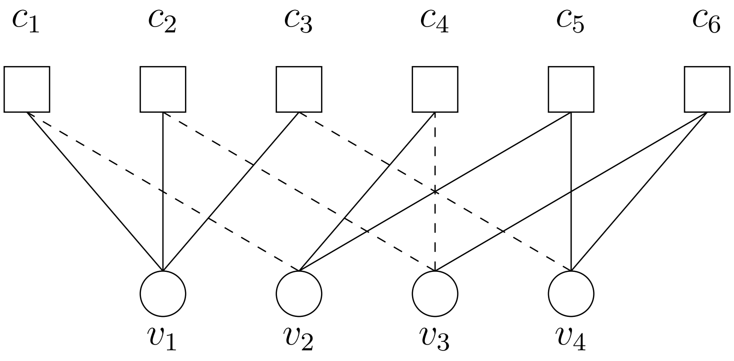
3.6.1 Minimal Stopping Set Size

As the performance is mainly dominated by size-2 stopping sets, we can improve the burst erasure correction capability by expurgating the ensemble and thereby removing all small stopping sets. Observing that a size-2 stopping set, as shown in Fig. 3.2, is built around 4-cycles, we can reduce the size of the minimal stopping sets by removing small cycles from the graph. For example, increasing the girth of the graph to 6 leads to minimal stopping sets of size $s_{\min} = d_v + 1$ [22].

We give a simple construction which we will use to find the probability of a size $(d_v + 1)$ stopping set in a SP of a SC-LDPC code. Let us consider a $(3, 6)$ random ensemble as an example. We immediately notice that size-3 stopping sets vanish once *girth* $= 6$. A size-4 stopping set is shown in Fig. 3.9 along with its bi-adjacency matrix that describes the neighbors of each VN in the corresponding row. We can notice a pattern in this matrix that can be generalized to get a $(d_v{+}1)\times(\frac{s_{\min}d_v}{2})$ matrix for a (d_v, d_c) LDPC (or SC-LDPC) ensemble. The pattern has been highlighted using dashed lines in the matrix: row $i \in \{1, 2, \ldots, d_v\}$ has one subset of $(d_v - (i - 1))$ columns with all 1s and an identity matrix I_{d_v-i+1} spanning these columns starting from row $i + 1$. Such a construction always corresponds to a minimal stopping set of size $(d_v + 1)$ and involves exactly $\frac{s_{min}d_v}{2} = \frac{(d_v+1)d_v}{2}$ neighboring CNs.

3.6.2 Performance on the SPBC

We can use the same approach as in Section 3.3.1 to calculate the probability of occurrence of the stopping set shown in Fig. 3.9 within a spatial position of a code



$$\begin{array}{c|cccccc|}
 & c_1 & c_2 & c_3 & c_4 & c_5 & c_6 \\
v_1 & 1 & 1 & 1 & 0 & 0 & 0 \\
v_2 & 1 & 0 & 0 & 1 & 1 & 0 \\
v_3 & 0 & 1 & 0 & 1 & 0 & 1 \\
v_4 & 0 & 0 & 1 & 0 & 1 & 1
\end{array}$$

Figure 3.9: A size-4 stopping set from an expurgated $(3, 6, w, L, M)$ random ensemble. CNs $\{c_1, c_2, c_3, c_4, c_5, c_6\}$ and VNs $\{v_1, v_2, v_3, v_4\}$ have been labeled for convenience. The solid edges indicate definite connections and the dashed edges complete one configuration to form a stopping set. Multiple edges are not allowed in the ensemble. The bi-adjacency matrix is also shown with its pattern highlighted.

sampled uniformly from the ensemble. Once again we have $p = \frac{T_{ss}}{T}$, where T_{ss} is the total number of combinations of the edges of v_1, v_2, v_3, v_4 that form a stopping set and T is the total number of combinations by which these edges can fit to the available CN sockets. For an expurgated $(3, 6, w, L, M)$ random ensemble, we have

$$\begin{aligned} T_{ss} = \; & [(1)] \times [(3wM)(3wM-6)(3wM-12)] \times \frac{3!}{0! \times 3!} \\ & \times [(15)] \times [(3wM-18)(3wM-24)] \times \frac{3!}{1! \times 2!} \\ & \times [(20)(10)] \times [(3wM-30)] \times \frac{3!}{2! \times 1!} \\ & \times [(15)(10)(5)] \times [(1)] \times \frac{3!}{3! \times 0!}. \end{aligned}$$

Since T is the total number of combinations in which the edges of $(d_v + 1)$ VNs can be assigned to sockets ensuring no 4-cycles, we can again approximate it by its dominant term as

$$T \approx \prod_{j=0}^{d_v(d_v+1)-1} (wMd_v - jd_c).$$

For a general (d_v, d_c, w, L, M) random ensemble, the expression for T_{ss} can be calculated as

$$\begin{aligned} T_{ss} = \; & \prod_{i=0}^{d_v} \left[\prod_{j=1}^{i} j(d_c - 1)(d_v - i + 1) \right] \\ & \times \left[\prod_{k=\sum_{m=0}^{i-1}(d_v-m)}^{\sum_{m=0}^{i-1}(d_v-m)+(d_v-i-1)} (wMd_v - kd_c) \right] \binom{d_v}{i}. \end{aligned}$$

It can be verified that the last value for k in the above expression is $k = \frac{d_v(d_v+1)}{2} - 1$. Then, we can simplify and rearrange the expression as

$$\begin{aligned}
T_{ss} = & \left[\prod_{k=0}^{\frac{d_v(d_v+1)}{2}-1} (wMd_v - kd_c) \right] \\
& \times \prod_{i=0}^{d_v} \left[\prod_{j=1}^{i} j \right] \left[\prod_{j=1}^{i} (d_c-1)(d_v-i+1) \right] \\
& \times \frac{d_v!}{i! \times (d_v-i)!} \\
= & \, T_{1/2} \times \prod_{i=1}^{d_v} i! \left[(d_c-1)(d_v-i+1) \right]^i \times \frac{d_v!}{i! \times (d_v-i)!} \\
T_{ss} = & \, T_{1/2} \times \prod_{i=1}^{d_v} \left[(d_c-1)(d_v-i+1) \right]^i \times \frac{d_v!}{(d_v-i)!},
\end{aligned}$$

where $T_{1/2} = \prod_{k=0}^{\frac{d_v(d_v+1)}{2}-1} (wMd_v - kd_c)$ is the first half of the products in T which can be canceled while calculating p, so that

$$\frac{T}{T_{1/2}} \cong \prod_{j=\frac{d_v(d_v+1)}{2}}^{d_v(d_v+1)-1} (wMd_v - jd_c).$$

For a general (d_v, d_c, w, M) ensemble, the probability of forming such a minimal stopping set of size (d_v+1) can be shown to be

$$p = \frac{T_{ss}}{T} \approx \frac{\prod_{i=1}^{d_v} \left[(d_c-1)(d_v-i+1) \right]^i \times \frac{d_v!}{(d_v-i)!}}{\prod_{j=\frac{d_v(d_v+1)}{2}}^{d_v(d_v+1)-1} (wMd_v - jd_c)}, \tag{3.11}$$

which means the expected number of such stopping sets within a SP of the code is $\lambda_{SP} = \binom{M}{d_v+1} p$. Using similar arguments as in Section 3.3, we have

$$\mathbb{N}_{d_v+1}^{SP} \sim \text{Poisson}(\lambda_{SP}).$$

A tight approximation for the average block erasure probability on the SPBC, $P_{\mathrm{B}}^{\mathrm{SPBC}}$, can be calculated as

$$P_{B,\exp}^{\mathrm{SPBC}} \approx 1 - e^{-\lambda_{\mathrm{SP}}} \approx \lambda_{\mathrm{SP}}. \tag{3.12}$$

3.6.3 Simulations

We performed Monte-Carlo simulations for an expurgated $(3, 6)$ random ensemble with $w = 3$ and counted 100 decoding failures on the SPBC. The simulation averages for varying M and their respective lower bounds calculated using (3.11) and (3.12) are plotted in Fig. 3.10. It is evident that the bound becomes tight very quickly which reassures that the decoder performance is indeed dominated by minimal stopping sets.

3.7 Finite Length Observations

We now compare the average performance of different SC-LDPC ensembles on the SPBC. We fix the asymptotic code rate as $\frac{1}{2}$, the smoothing parameter as $w = d_v$ and plot the (tight) approximations on P_B^{SPBC} of three ensembles, namely $(3, 6)$, $(4, 8)$ and $(5, 10)$, for both the unexpurgated and the expurgated cases in Fig. 3.11.

- For the unexpurgated case, the average block erasure probability varies as

  $$P_{\mathrm{B}}^{\mathrm{SPBC}} \sim O(M^{2-d_v}).$$

  Hence, linearly increasing d_v, for a constant rate $\frac{1}{2}$, keeps improving the performance by multiples of $1/M$.

- When the ensemble is expurgated so that *girth* $= 6$, the improvement is by an

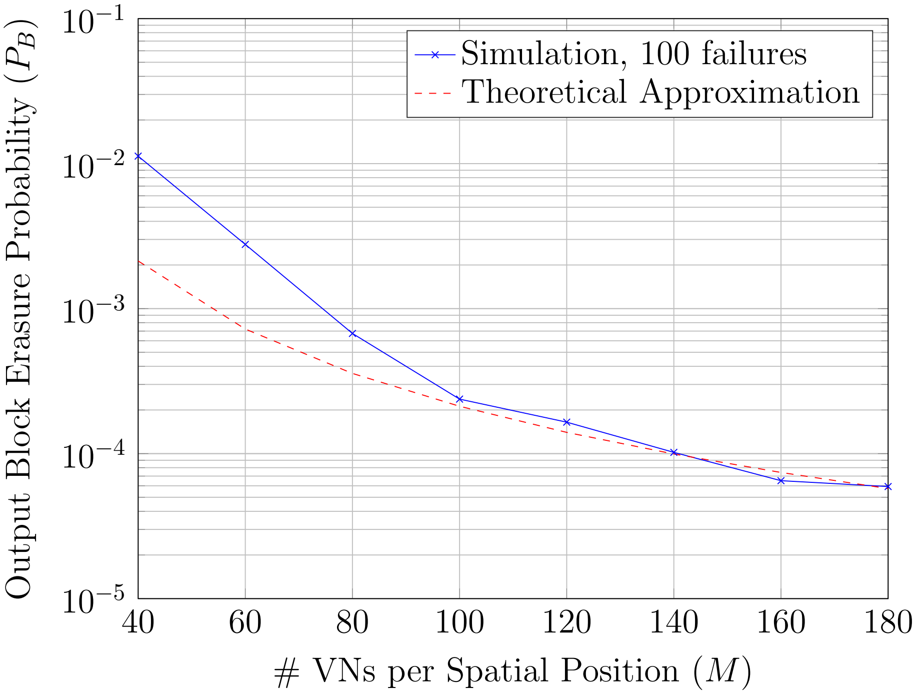


Figure 3.10: Monte Carlo simulations on the SPBC with an expurgated $(3,6)$ random ensemble for $w = 3$ along with the theoretical approximation. The approximation becomes tight very quickly with M.

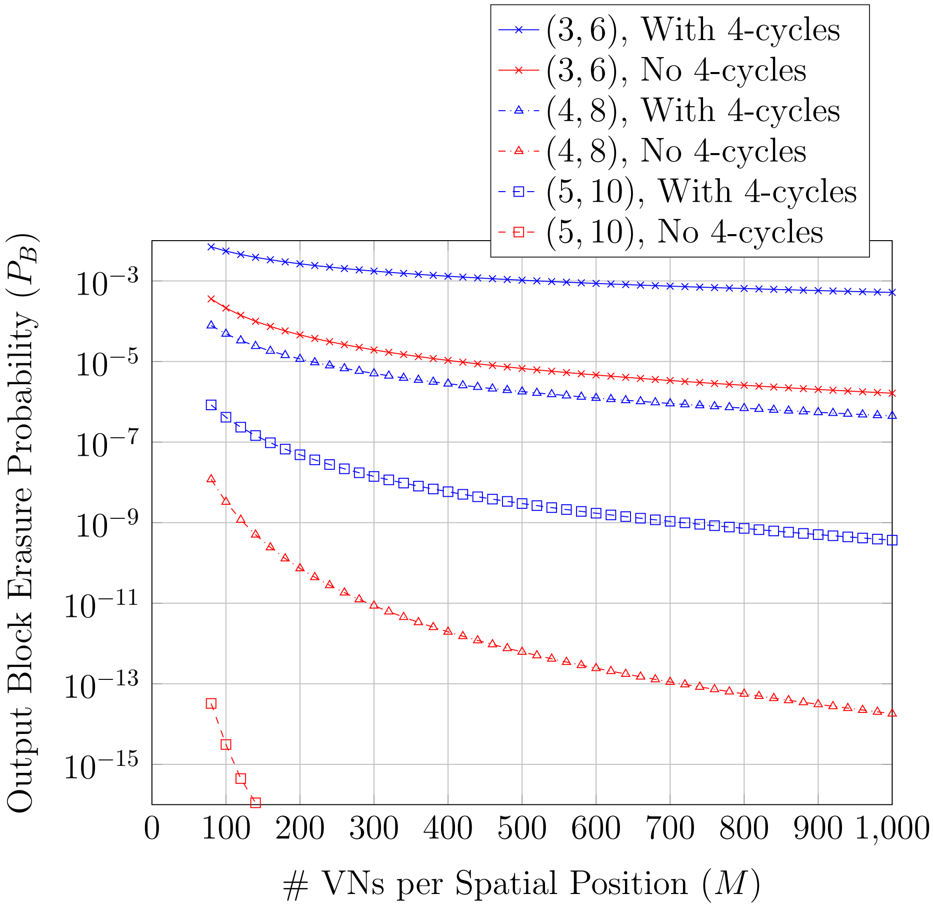


Figure 3.11: The theoretical approximations on P_B^{SPBC} for various ensembles in both the unexpurgated and expurgated scenarios.

order of $\frac{d_v+1}{2}$ in M. Now, we have

$$P_{\mathrm{B}}^{\mathrm{SPBC}} \sim O(M^{(d_v+1)(2-d_v)/2}).$$

Therefore, for a fixed rate $\frac{1}{2}$, a unit increase in d_v improves the performance by a factor of about M^{-d_v}.

- As d_v is increased, it was observed that the performance is worse if w is kept constant. This is because higher size stopping sets dominated when $w < d_v$. All the bounds presented in this work are tight only when $w \geq d_v$.

3.8 Conclusion

We have analyzed random SC-LDPC ensembles on the burst erasure channel and provided insights into improving the block erasure probability through increased VN degree and expurgation. The expected error floor for the ensemble has been characterized and verified on the BEC. We have shown through these results that the vector in (3.5) completely characterizes the ensemble performance on the erasure channel.

There is more work to be done to arrive at tighter bounds for the block erasure channel. We also need to analyze the expurgated ensemble on the random burst channel. One method to do that would be to find the vector in (3.5) for the expurgated ensemble. Since that is very tedious, the main challenge in this direction is finding a simpler way of characterizing the performance. Also, we have observed that higher size stopping sets dominate when $w < d_v$. An explicit proof for this could be insightful.

4. CONCLUSIONS

In Section 2, this work introduces a method to construct cyclic polar codes over $\mathbb{F}_q$ for any blocklength N satisfying $N|(q-1)$. For the QEC, these codes can be decoded efficiently using Forney's algebraic decoder to decode the intermediate blocks. In our simulations, they outperform standard polar codes. For the case of $N = 2^n$, a soft-decision SC decoder was also implemented and tested on the q-ary symmetric channel. Under SC decoding, cyclic polar codes clearly outperform RS codes of the same rate and blocklength.

An algebraic errors and erasures decoding strategy was also considered for the intermediate block codes. Preliminary results show that this approach is suboptimal when compared to hard decision decoding of a RS code with the same rate and blocklength. In future work, we plan to consider APP decoding of the intermediate blocks for small lengths while retaining a hard-decision decoder at larger blocks typically placed close to inputs in the graph. We will also consider the rate of polarization for these codes based on similar work for standard polar codes [13, 20]. The programs developed for this work can be accessed at `https://github.com/nrenga/cyclic_polar`.

In Section 3, we have analyzed random SC-LDPC ensembles on the burst erasure channel and provided insights into improving the block erasure probability through increased VN degree and expurgation. The expected error floor for the ensemble has been characterized and verified on the BEC. We have shown through these results that the vector in (3.5) completely characterizes the ensemble performance on the erasure channel.

There is more work to be done to arrive at tighter bounds for the block erasure channel. We also need to analyze the expurgated ensemble on the random burst channel. One method to do that would be to find the vector in (3.5) for the expurgated ensemble. Since that is very tedious, the main challenge in this direction is finding a simpler way of characterizing the performance. Also, we have observed that higher size stopping sets dominate when $w < d_v$. An explicit proof for this could be insightful.

REFERENCES

[1] Erdal Arıkan. Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels. *IEEE Trans. Inform. Theory*, 55(7):3051–3073, July 2009.

[2] Erdal Arıkan. Systematic polar coding. *IEEE Commun. Letters*, 15(8):860–862, 2011.

[3] Richard E. Blahut. *Fast Algorithms for Digital Signal Processing.* Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1st edition, 1985.

[4] Richard E. Blahut. *Algebraic Codes for Data Transmission.* Cambridge University Press, 2003.

[5] Kai Lai Chung. *A course in probability theory.* New York: Academic Press, 1974.

[6] James Cooley and John Tukey. An algorithm for the machine calculation of complex Fourier series. *Mathematics of Computation*, 19(90):297–301, 1965.

[7] E. Şaşoğlu, I.E. Telatar, and E. Arıkan. Polarization for arbitrary discrete memoryless channels. In *Proc. IEEE Inform. Theory Workshop*, pages 144–148, Oct. 2009.

[8] J. Felström and K. S. Zigangirov. Time-varying periodic convolutional codes with low-density parity-check matrix. *IEEE Trans. Inform. Theory*, 45(6):2181–2191, Sept. 1999.

[9] G.D. Forney. On decoding BCH codes. *IEEE Trans. Inform. Theory*, 11(4):549–557, Oct. 1965.

[10] A.R. Iyengar, M. Papaleo, Gianluigi Liva, P.H. Siegel, J.K. Wolf, and G.E. Corazza. Protograph-based ldpc convolutional codes for correlated erasure channels. In *Proc. IEEE Int. Conf. Commun.*, pages 1–6, May 2010.

[11] George Johnson. Claude Shannon, mathematician, dies at 84. *New York Times*, 27 Feb 2001. [Online]. Available: http://www.nytimes.com/2001/02/27/nyregion/claude-shannon-mathematician-dies-at-84.html.

[12] A. Jule and I. Andriyanova. Performance bounds for spatially-coupled ldpc codes over the block erasure channel. In *Proc. IEEE Int. Symp. Inform. Theory*, pages 1879–1883, July 2013.

[13] Satish Babu Korada, Eren Şaşoğlu, and Rüdiger Urbanke. Polar codes: Characterization of exponent, bounds, and constructions. *IEEE Trans. Inform. Theory*, 56(12):6253–6264, 2010.

[14] S. Kudekar, T. J. Richardson, and R. L. Urbanke. Threshold saturation via spatial coupling: Why convolutional LDPC ensembles perform so well over the BEC. *IEEE Trans. Inform. Theory*, 57(2):803–834, Feb. 2011.

[15] S. Kudekar, T.J. Richardson, and R.L. Urbanke. Spatially coupled ensembles universally achieve capacity under belief propagation. *IEEE Trans. Inform. Theory*, 59(12):7761–7813, Dec. 2013.

[16] Michael Lentmaier, Gerhard P Fettweis, Kamil Sh Zigangirov, and Daniel J Costello. Approaching capacity with asymptotically regular LDPC codes. In *Proc. Annual Workshop on Inform. Theory and its Appl.*, pages 173–177, 2009.

[17] Hiroki Mori and Tadashi Wadayama. Band splitting permutations for spatially coupled LDPC codes enhancing burst erasure immunity. *CoRR*, abs/1501.04394,

2015. [Online]. Available: http://arxiv.org/abs/1501.04394.

[18] Ryuhei Mori and Toshiyuki Tanaka. Performance and construction of polar codes on symmetric binary-input memoryless channels. In *Proc. IEEE Int. Symp. Inform. Theory*, pages 1496–1500, 2009.

[19] Ryuhei Mori and Toshiyuki Tanaka. Non-binary polar codes using Reed-Solomon codes and algebraic geometry codes. In *Proc. IEEE Inform. Theory Workshop*, pages 1–5, Aug. 2010.

[20] Ryuhei Mori and Toshiyuki Tanaka. Source and channel polarization over finite fields and Reed-Solomon matrices. *IEEE Trans. Inform. Theory*, 60(5):2720–2736, 2014.

[21] P.M. Olmos and R.L. Urbanke. Scaling behavior of convolutional ldpc ensembles over the bec. In *Proc. IEEE Int. Symp. Inform. Theory*, pages 1816–1820, July 2011.

[22] A. Orlitsky, R. Urbanke, K. Viswanathan, and J. Zhang. Stopping sets and the girth of tanner graphs. In *Proc. IEEE Int. Symp. Inform. Theory*, pages 2–, 2002.

[23] H.D. Pfister. A brief introduction to polar codes. 21 April 2014. [Online]. Available: http://pfister.ee.duke.edu/courses/ecen655/polar.pdf.

[24] N. Rengaswamy and H.D. Pfister. Cyclic Polar Codes. In *Proc. IEEE Int. Symp. Inform. Theory*, pages 1287–1291, June 2015.

[25] T.J. Richardson and R.L. Urbanke. *Modern Coding Theory.* Cambridge University Press, 2008.

[26] C.E. Shannon. A symbolic analysis of relay and switching circuits. *Trans. of the American Institute of Electrical Engineers*, 57(12):713–723, Dec 1938.

[27] C.E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27(3):379–423, July 1948.

[28] N. ul Hassan, I. Andriyanova, M. Lentmaier, and G. P. Fettweis. Protograph design for spatially-coupled codes to attain an arbitrary diversity order. In *Proc. IEEE Inform. Theory Workshop*, Jeju City, South Korea, Oct. 2015.

[29] N. ul Hassan, M. Lentmaier, I. Andriyanova, and G.P. Fettweis. Improving code diversity on block-fading channels by spatial coupling. In *Proc. IEEE Int. Symp. Inform. Theory*, pages 2311–2315, June 2014.

APPENDIX A

COOLEY-TUKEY FORMULA

A.1 Discussion

In this section, we will discuss the details of the Cooley-Tukey fast Fourier transform [3, 6] and derive the Kronecker product formulation of the same as given in Lemma 3.

Consider two vectors $\underline{u}$ and $\underline{v}$ such that $\underline{u}$ is the Fourier transform of $\underline{v}$ and let $\ell = ab$ be the length of the vectors, where a and b are positive integers. The Fourier transform is given by

$$u_i = [F_\ell \underline{v}]_i = \sum_{j=0}^{\ell-1} \omega_\ell^{ij} v_j,$$

where the matrix F_ℓ is defined by $[F_\ell]_{i,j} \triangleq \omega_\ell^{ij}$. Now, express each of the indices with a coarse index and vernier index as

$$j = j' + bj'' \quad ; \quad i = ai' + i''$$

where $i', j' = 0, 1, \ldots, b-1$ and $i'', j'' = 0, 1, \ldots, a-1$. By making these substitutions we get

$$u_{ai'+i''} = \sum_{j''=0}^{a-1} \sum_{j'=0}^{b-1} \omega_\ell^{(j'+bj'')(ai'+i'')} v_{j'+bj''}.$$

Now define $\gamma = \omega_\ell^b$ and $\beta = \omega_\ell^a$ so that they have multiplicative orders a and b, respectively, in $\mathbb{F}_q$. Since ω_ℓ has a multiplicative order of $\ell = ab$, $\omega_\ell^{abj''i'} = 1$ in the expansion of the above formula. Rearranging the expression gives us the following

$$\begin{bmatrix} v_0\\ v_1\\ v_2\\ v_3\\ v_4\\ v_5 \end{bmatrix} \xrightarrow{(i)} \begin{bmatrix} v'_0 & v'_2 & v'_4 \\ v'_1 & v'_3 & v'_5 \end{bmatrix} \xrightarrow{(ii)} \begin{bmatrix} u'_0 & u'_1 & u'_2 \\ u'_3 & u'_4 & u'_5 \end{bmatrix} \xrightarrow{(iii)} \begin{bmatrix} u_0\\ u_1\\ u_2\\ u_3\\ u_4\\ u_5 \end{bmatrix}$$

Figure A.1: Sequence of operations in the Cooley-Tukey fast Fourier transform for the case $\ell = 6, a = 3, b = 2$.

convenient form of the Cooley-Tukey fast Fourier transform.

$$u_{ai'+i''} = \sum_{j'=0}^{b-1} \beta^{j'i'} \left[\omega_\ell^{j'i''} \left(\sum_{j''=0}^{a-1} \gamma^{j''i''} v_{j'+bj''} \right) \right].$$

This form allows us to fragment the Fourier transform into a sequence of operations which reduces the overall complexity as we will see next. The indices i and j are expressed in two different combinations of their coarse and vernier indices to arrive at this convenient form of the transform.

Next, we will see the sequence of operations in computing the transform. The above expression of the Cooley-Tukey FFT indicates that the computation is closely related to a two-dimensional (2-D) Fourier transform. The input vector $\underline{v}$, of length ℓ, is rearranged column-wise into a 2-D matrix of dimensions $b \times a$. Fig. A.1 shows an example demonstrating the following sequence of operations.

(i) Firstly, the inner summation is the 1-D FFT,

$$v'_{j'+bi''} = \sum_{j''=0}^{a-1} \gamma^{j''i''} v_{j'+bj''},$$

of each row of this matrix; for each value of j', compute a length-a Fourier transform of the vector $\underline{v}_{j'} = (v_{j'+bj''})$ that outputs the vector $\underline{v}'_{j'} = (v'_{j'+bi''})$ which are the set of summations for $i'' = 0, 1, \ldots, a-1$. Hence this intermediate output vector can be indexed with $i = bi'' + j'$ so that $i'' = \lfloor i/b \rfloor$ and $j' = i \bmod b$.

(ii) Then, all elements are multiplied by $\omega_\ell^{j'i''}$ and reshuffled to give

$$u'_{aj'+i''} = \omega_\ell^{j'i''} v'_{j'+bi''},$$

where j' and i'' vary across the rows and columns of the matrix, respectively. The different expressions for i and j in terms of their coarse and vernier indices explain the need for reshuffling in this step. In a Fourier transform, all indices need to be involved in computing each output coefficient but step (i) has only involved interleaved indices. Hence, in the next step, adjacent indices should be involved to complete the transform.

(iii) Finally, the outer summation is the 1-D FFT,

$$u_{ai'+i''} = \sum_{j'=0}^{b-1} \beta^{j'i'} u'_{aj'+i''},$$

of each column of the resultant matrix obtained after the multiplication step; for each value of i'', compute a length-b Fourier transform whose inputs are indexed by j' and the output $\underline{u}$ is indexed by $i' = 0, 1, \ldots, b-1$, clearly indicating an output interleaved by a. Hence, the (output) indexing implies that the output vector is to be read row-wise from the matrix after the last (row-FFT) step.

Naïve implementations of the b length-a transforms and the a length-b transforms would require a complexity of $O(ba^2)$ and $O(ab^2)$, respectively. Therefore, the total complexity of the Fourier transform is now reduced from $O(\ell^2)$ to $O(\ell(a+b))$.

A.2 Proof of Lemma 3

Consider $\underline{v} = (v_0, v_1, \ldots, v_{ab-1})^T$ and $\underline{u} = (u_0, u_1, \ldots, u_{ab-1})^T$ to be the input and output vectors of the transform, respectively. We follow the sequence of operations described above to translate the summations into equivalent matrix operations.

(i) First, we perform b length-a Fourier transforms on b interleaved blocks as the matrix-vector product

$$\underline{v}' = (F_a \otimes I_b)\underline{v},$$

where $A \otimes B$ denotes the Kronecker product of matrix A with matrix B as given in Definition 1. It is important to note that both the input and output vectors have no shuffling in the indices of their elements.

(ii) Next, we multiply each element of $\underline{v}'$, indexed by $i = bi'' + j'$, by the twiddle factor $\omega_{ab}^{(\lfloor i/b \rfloor)(i \bmod b)}$. If we construct a diagonal matrix $D_{a,b}$ with these factors as its main-diagonal elements, then this step can be expressed as

$$\underline{u}' = D_{a,b}\underline{v}'.$$

(iii) Finally, we perform the a length-b Fourier transforms on a adjacent blocks of $\underline{u}'$ as the matrix-vector product

$$\underline{u} = (I_a \otimes F_b)\underline{u}'.$$

Since this output vector has its indices shuffled, we also need to deinterleave it. Hence, the final expression for the transform is given by

$$\underline{u} = S_{b,a}(I_a \otimes F_b)D_{a,b}(F_a \otimes I_b)\underline{v},$$

where $S_{b,a}$ is the perfect-shuffle permutation matrix introduced in Definition 2. □

A.3 Proof of Lemma 4

From Lemma 3 we have, for $N = a \times b$,

$$\begin{aligned} F_{ab} &= S_{b,a}(I_a \otimes F_b)D_{a,b}(F_a \otimes I_b) \\ &= (F_b \otimes I_a)S_{b,a}D_{a,b}(F_a \otimes I_b) \\ &= [(S_{1,b}D_{b,1} \otimes I_a)(F_b \otimes I_a)] \times [(S_{b,a}D_{a,b} \otimes I_1)(F_a \otimes I_b)]. \end{aligned}$$

Now, let us see the extension for $N = a \times bc$.

$$\begin{aligned} F_N &= (F_{bc} \otimes I_a)S_{bc,a}D_{a,bc}(F_a \otimes I_{bc}) \\ &= [((F_c \otimes I_b)S_{c,b}D_{b,c}(F_b \otimes I_c) \otimes I_a)(F_b \otimes I_a)] \times [(S_{bc,a}D_{a,bc} \otimes I_1)(F_a \otimes I_{bc})] \\ &= [F_c \otimes I_b \otimes I_a] \times [S_{c,b}D_{b,c}(F_b \otimes I_c) \otimes I_a] \times [(S_{bc,a}D_{a,bc} \otimes I_1)(F_a \otimes I_{bc})] \\ &= [(S_{1,c}D_{c,1} \otimes I_{ba})(F_c \otimes I_{ba})] \times [(S_{c,b}D_{b,c} \otimes I_a)(F_b \otimes I_c \otimes I_a)] \\ &\quad \times [(S_{bc,a}D_{a,bc} \otimes I_1)(F_a \otimes I_{bc})] \\ &= [(S_{N/abc,c}D_{c,N/abc} \otimes I_{abc/c})(F_c \otimes I_{N/c})] \times [(S_{N/ab,b}D_{b,N/ab} \otimes I_{ab/b})(F_b \otimes I_{N/b})] \\ &\quad \times [(S_{N/a,a}D_{a,N/a} \otimes I_{a/a})(F_a \otimes I_{N/a})]. \end{aligned}$$

We see a pattern in the recurison which can be generalized for length $N = \prod_{m=1}^{n} \ell_m$. Define $p_j = \prod_{m=1}^{j} \ell_j$ and

$$U_m = (S_{N/p_m,\ell_m} D_{\ell_m,N/p_m} \otimes I_{p_m/\ell_m})(F_{\ell_m} \otimes I_{N/\ell_m}).$$

Then, the Fourier transform can be expressed as

$$F_N = U_n U_{n-1} \cdots U_1. \qquad \square$$

APPENDIX B

CHANNEL POLARIZATION

In this section, we prove the polarization theorem, stated in Section 2.5.1 Theorem 5, for the cyclic polar code construction. First, we prove Lemma 6 so that the result can be used to prove the theorem.

B.1 Proof of Lemma 6

Observing that every term in the summation of (2.11) is positive, we have the following:

(i)

$$\begin{aligned}\frac{1}{\ell}\sum_{j=0}^{\ell-1}\psi(\ell,j,\epsilon') &= \frac{1}{\ell}\sum_{j=0}^{\ell-1}\sum_{i=0}^{(\ell-1)-j}\binom{\ell}{i}(1-\epsilon')^i(\epsilon')^{\ell-i}\\ &= \sum_{i=0}^{\ell-1}\frac{(\ell-i)}{\ell}\binom{\ell}{i}(1-\epsilon')^i(\epsilon')^{\ell-i}\\ &= \epsilon'\sum_{i=0}^{\ell-1}\binom{\ell-1}{i}(1-\epsilon')^i(\epsilon')^{\ell-1-i}\\ &= \epsilon'\end{aligned}$$

(ii) Given $\epsilon' \in (0,1)$, we have

$$\psi(\ell,\ell-1,\epsilon') = \binom{\ell}{0}(1-\epsilon')^0(\epsilon')^\ell = (\epsilon')^\ell < \epsilon'$$

and

$$\psi(\ell,0,\epsilon') = 1-\binom{\ell}{\ell}(1-\epsilon')^\ell(\epsilon')^0 = 1-(1-\epsilon')^\ell.$$

Now, consider $\psi(\ell, 0, \epsilon') - \epsilon'$. We have

$$\begin{aligned}\psi(\ell, 0, \epsilon') - \epsilon' &= 1 - (1-\epsilon')^{\ell} - \epsilon' \\ &= (1-\epsilon') - (1-\epsilon')^{\ell} \\ &> 0.\end{aligned}$$

Hence, $\psi(\ell, \ell-1, \epsilon') < \epsilon' < \psi(\ell, 0, \epsilon')$. □

B.2 Proof of Theorem 5

We use the same strategy as Arıkan used in [1] but we slightly generalize the channel evolution tree and the mathematical framework to re-formulate the problem in our scenario. The primary requirement for this is that, at every stage of the transform, each channel splits into multiple channels in our case whereas in the original polar code construction, each channel split into exactly two channels at every stage.

The root node of the tree is associated with the underlying QEC W. At level 1, W evolves into ℓ_1 channels, namely $W_{\ell_1}^{(1)}, W_{\ell_1}^{(2)}, \ldots, W_{\ell_1}^{(\ell_1)}$. We have ℓ_1 nodes corresponding to ℓ_1 channels at level 1. At level 2, every channel from level 1 gives birth to ℓ_2 channels. Hence, we have the channels $W_{\ell_1\ell_2}^{(1)}, W_{\ell_1\ell_2}^{(2)}, \ldots, W_{\ell_1\ell_2}^{(\ell_1\ell_2)}$, and so on. The i^{th} channel from the top at level n will be denoted by $W_{\ell_1\ell_2\cdots\ell_n}^{(i)}$.

Since this is not a binary tree, the channels have to be indexed by ℓ_m-ary symbols s_m's, for $m = 1, 2, \ldots$. Define $L_m \triangleq \{0, 1, \ldots, \ell_m - 1\}$. The root node is indexed with a null sequence. The nodes at level 1 are indexed with symbol $s_1 \in L_1$. Given a node at level m with the symbol sequence $s_1 s_2 \cdots s_m$, the child nodes at the next level will have indices $s_1 s_2 \cdots s_m 0, s_1 s_2 \cdots s_m 1, \ldots,$ $s_1 s_2 \cdots s_m(\ell_m - 1)$. According to this labeling, the channel $W_{\ell_1\ell_2\cdots\ell_m}^{(i)}$ is situated at the node $s_1 s_2 \cdots s_m$ with $i = 1 + \sum_{j=1}^{m} s_j \ell_j^{m-j}$. Alternatively, we denote this channel as $W_{s_1 s_2 \cdots s_m}$.

We redefine the random tree process $\{K_m; m \geq 0\}$. The process begins at the root node with $K_0 = W$. At level 1, the process takes the value $K_1 = W_{s_1}$, where all values for s_1 are equally likely. In general, if $K_m = W_{s_1 s_2 \cdots s_m}$, then $K_{m+1} = W_{s_1 s_2 \cdots s_m j}$ for any $j \in L_{m+1}$ with probability $1/\ell_{m+1}$ each. We need to associate the channel obtained as the value of the process at each stage with its reliability parameter, i.e. the Bhattacharyya parameter, in order to track the evolution of the erasure rates after each step of polarization. Since the rate and reliability parameters have a complementary relation for the erasure channel, it is not necessary to also associate the rate parameter with the random tree process. Hence, we define the reliability random process $\{Z_m; m \geq 0\}$ as $Z_m = Z(K_m)$.

Now, consider the probability space $(\Omega, \mathcal{F}, P)$ where Ω is the space of all sequences $(s_1, s_2, \ldots) \in L_1 \times L_2 \times \cdots$, $\mathcal{F}$ is the Borel field generated by the cylinder sets $S(s_1, s_2, \ldots, s_n) \triangleq \{\omega \in \Omega : \omega_1 = s_1, \ldots, \omega_n = s_n\}, n \geq 1, s_m \in L_m$, P is the probability measure defined on $\mathcal{F}$ such that $P(S(s_1, \ldots, s_n)) = 1/\prod_{m=1}^{n} \ell_m$. For each $n \geq 1$, we define $\mathcal{F}_n$ as the Borel field generated by the cylinder sets $S(s_1, s_2, \ldots, s_m), 1 \leq m \leq n, s_i \in L_i$. We define $\mathcal{F}_0$ as the trivial field consisting only of the null set and Ω. Clearly, $\mathcal{F}_0 \subset \mathcal{F}_1 \subset \cdots \subset \mathcal{F}$.

Then, we can define the random processes as follows. For $\omega = (\omega_1, \omega_2, \ldots) \in \Omega$ and $n \geq 1$, define $K_n(\omega) = W_{s_1 s_2 \cdots s_n}$ and $Z_n(\omega) = Z(K_n(\omega))$. For $n = 0$, define $K_0 = W, Z_0 = Z(W)$. Hence, for any fixed $n \geq 0$, the RVs K_n and Z_n are measurable with respect to $\mathcal{F}_n$.

Lemma 7. *The sequence of random variables and Borel fields $\{Z_n, \mathcal{F}_n; n \geq 0\}$ is a* *martingale, i.e.,*

$$\mathcal{F}_n \subset \mathcal{F}_{n+1} \textit{ and } Z_n \textit{ is } \mathcal{F}_n\textit{-measurable}, \tag{B.1}$$

$$E[|Z_n|] < \infty, \tag{B.2}$$

$$Z_n = E[Z_{n+1}|\mathcal{F}_n]. \tag{B.3}$$

Proof. Condition (B.1) is satisfied just by construction and (B.2) is given by the fact that $0 \leq Z_n \leq 1$. To prove (B.3), consider a cylinder set $S(s_1, s_2, \ldots, s_n) \in \mathcal{F}_n$ and set $Z(W_{s_1 \cdots s_n}) = \epsilon', Z(W_{s_1 \cdots s_n j}) = \psi(\ell_n, j, \epsilon')$ in the result of Lemma 6 to write

$$\begin{aligned} E[Z_{n+1}|S(s_1, s_2, \ldots, s_n)] &= \frac{1}{\ell_{n+1}} \sum_{j=0}^{\ell_{n+1}-1} Z(W_{s_1 \cdots s_n j}) \\ &= Z(W_{s_1 \cdots s_n}) \end{aligned}$$

Since $Z(W_{s_1 \cdots s_n})$ is the value of Z_n on $S(s_1, s_2, \ldots, s_n)$, (B.3) follows. This completes the proof that $\{Z_n, \mathcal{F}_n\}$ is a martingale. □

Lemma 8. *The sequence $\{Z_n; n \geq 0\}$ converges a.e. to a random variable Z_∞ such that*

$$E[Z_\infty] = Z_0, \tag{B.4}$$

$$Z_\infty \in \{0, 1\} \textit{ a.e.} \tag{B.5}$$

Proof. Since $\{Z_n, \mathcal{F}_n\}$ is uniformly integrable, (B.4) follows from standard convergence results about such martingales (see, e.g., [5, Theorem 9.4.6]). From Lemma 6, we see that the individual channel erasure rates $\psi(\ell, j, \epsilon')$ polarize away from the input channel erasure rate ϵ', while the mean is preserved to be ϵ'. Since the erasure rate ψ is bounded in $[0, 1]$, the polarization will recur until it reaches either of the fixed points in the set $\{0, 1\}$. (B.5) follows automatically. □

From the above results, we have

$$E[Z_\infty] = 1 \cdot P(Z_\infty = 1) + 0 \cdot P(Z_\infty = 0) = Z_0.$$

Conditioning that we start with a channel of erasure rate $Z_0 = \epsilon$, the theorem follows. This completes the proof of Theorem 5. □

APPENDIX C

FORNEY'S DECODER FOR SMALL BLOCKS

In 1965, Forney described a simplified algorithm for the decoding of RS and BCH codes [9]. The algorithm is suitable for errors and erasures decoding. In this description, we focus on the case where:

- The locations of errors and/or erasures are known and are given by the erasure locator polynomial,
$$\Lambda\left(x\right) = \prod_{l=1}^{\nu}(1 - X_l x)$$
where, X_l denotes the location of the l-th erasure and ν is the actual number of erasures.

- Syndromes can be computed based on known values in the codeword spectrum.

For the QEC, the erased positions are known at the receiver and hence, the erasure locator polynomial $\Lambda\left(x\right)$ can be easily computed. To verify the second condition, consider the Fourier transform pair $u(x)$ and $v(x)$ defined by

$$u(x) = \sum_{i=0}^{\ell-1} u_i x^i = \sum_{i=0}^{\ell-1} v(\omega_\ell^i) x^i$$

and

$$v(x) = \sum_{i=0}^{\ell-1} v_i x^i = \sum_{i=0}^{\ell-1} \left(\ell^{-1} u(\omega_\ell^{-i})\right) x^i.$$

These are GFFT and inverse GFFT equations associated with the cyclic polar code construction.

Now, we assume that the information polynomial $u(x)$ has r consecutive known values (not necessarily zeroes) starting from index b. Thus, the value $u_i = v(\omega_\ell^i)$ is known for

$$i \in \mathcal{B} = \{b + j \bmod (\ell - 1) \,|\, j \in \mathbb{Z}, 0 \leq j \leq r - 1\}, \tag{C.1}$$

where $b \in \{0, 1, \ldots, \ell - 1\}$. These known values are available at both the transmitter and receiver and allow us to satisfy second condition above.

The Decoder

Assume that the information polynomial $u(x)$ is encoded into $v(x)$ and transmitted via ℓ consecutive uses of QEC(ϵ). Let the received polynomial be $y(x) = v(x) + e(x)$, where $e(x)$ is a "error" polynomial that changes the coefficients of $y(x)$ to be zero at all erasure locations. To compute the syndromes $S_j = e(\omega_\ell^j)$, we note that $e(\omega_\ell^j) = y(\omega_\ell^j) - u_j$ for $j \in \mathcal{B}$. These are computable at the receiver because $y(x)$ is known (except for erasures) and u_j is known for $j \in \mathcal{B}$. Now, we restrict the discussion to the case of $b = 0$, which implies $\mathcal{B} = \{0, 1, 2, \ldots, r - 1\}$.

Assume the ν erasures occurred at positions i_l for $l = 1, 2, 3, \ldots, \nu$ and proceed as follows. Let the erasure and syndrome polynomials be

$$\begin{aligned} e(x) &= \sum_{l=1}^{\nu} e_{i_l} x^{i_l} = \sum_{l=1}^{\nu} (-v_{i_l})\, x^{i_l} \\ S(x) &= \sum_{j=0}^{r-1} S_j x^j, \end{aligned}$$

where

$$S_j = e(\omega^j) = \sum_{l=1}^{\nu} e_{i_l} X_l^j$$

and $X_l \triangleq \omega_\ell^{i_l}$ is the location of the l-th erasure. The erasure evaluator polynomial is defined as

$$\begin{aligned}
\Omega(x) &= S(x)\Lambda(x) && \pmod{x^r} \\
&= \left[\sum_{j=0}^{r-1}\left(\sum_{l=1}^{\nu} e_{i_l}X_l^j\right)x^j\right]\left[\prod_{m=1}^{\nu}(1-X_m x)\right] && \pmod{x^r} \\
&= \sum_{l=1}^{\nu} e_{i_l}\sum_{j=0}^{r-1}(X_l x)^j \prod_{m=1}^{\nu}(1-X_m x) && \pmod{x^r} \\
&= \sum_{l=1}^{\nu} e_{i_l}\underbrace{(1-X_l x)\sum_{j=0}^{r-1}(X_l x)^j}_{=1-(X_l x)^r}\prod_{m\neq l}^{\nu}(1-X_m x). && \pmod{x^r}
\end{aligned}$$

Since $(X_l x)^r \bmod x^r = 0$, we find that

$$\Omega(x) = \sum_{l=1}^{\nu} e_{i_l}\prod_{m\neq l}^{\nu}(1-X_m x)$$

for $\nu - 1 < r$. Substituting $x = X_k^{-1}$ we get

$$e_{i_k} = -\frac{X_k\Omega\left(X_k^{-1}\right)}{\Lambda'\left(X_k^{-1}\right)},$$

where i_k is the index of the k-th erasure. Once the erased values are obtained, $v(x)$ and $u(x)$ can be obtained from the erasure polynomial.

APPENDIX D

CAPACITY OF QSCE

Let the input alphabet be $\mathcal{X} = \{0, 1, 2, \ldots, q-1\}$, where each element is a representation of a unique q-ary symbol, with a probability distribution $p(X) = (p_0, \ldots, p_{q-1})$. Then, the output alphabet will be $\mathcal{Y} = \mathcal{X} \cup \{?\}$.

The capacity of the channel is defined as,

$$C = \max_{p(X)} I(X;Y) = \max_{p(X)} \left(H(Y) - H(Y|X)\right) \tag{D.1}$$

where, X and Y are random variables representing the input and output alphabet, respectively. Since, we have a symmetric channel, $H(Y|X)$ is independent of $p(X)$ and is given as

$$H(Y|X) = -\left[(1-\epsilon-\beta)\,\log_q(1-\epsilon-\beta) + \alpha\,\log_q(\alpha) + \beta\,\log_q\!\left(\frac{\beta}{q-1}\right)\right]. \tag{D.2}$$

The probabilities for the output symbols in $\mathcal{Y}$ are,

$$\begin{aligned} P(Y=i) &= (p_i)(1-\epsilon-\beta) + \sum_{\substack{j=0 \\ j\neq i}}^{q-1} (p_j)\left(\frac{\beta}{q-1}\right) \\ &= (p_i)(1-\epsilon-\beta) + (1-p_i)\left(\frac{\beta}{q-1}\right), \end{aligned}$$

$$P(Y=?) = \sum_{j=0}^{q-1} (p_j)(\epsilon) = \epsilon$$

for $i = 0, 1, 2, \ldots, q-1$. Hence,

$$\begin{aligned} H(Y) &= -\sum_{y\in\mathcal{Y}} p(y)\ \log_q p(y) \\ &= -\Bigg[\sum_{i=0}^{q-1} P(Y=i)\ \log_q(P(Y=i)) + \epsilon\ \log_q\epsilon\Bigg]. \end{aligned}$$

Since $H(Y|X)$ is independent of $p(X)$, $H(Y)$ has to be maximized in order to maximize capacity. Therefore,

$$\frac{\partial H(Y)}{\partial p_i} = 0$$

which implies

$$\begin{aligned} &\left(1-\epsilon-\beta-\frac{\beta}{q-1}\right)\left[1+\ \log_q\left(p_i(1-\epsilon-\beta)+(1-p_i)\left(\frac{\beta}{q-1}\right)\right)\right] = 0 \\ \Rightarrow\qquad & 1+\ \log_q\left(p_i(1-\epsilon-\beta)+(1-p_i)\left(\frac{\beta}{q-1}\right)\right) = 0 \\ \Rightarrow\qquad & p_i(1-\epsilon-\beta)+(1-p_i)\left(\frac{\beta}{q-1}\right) = \frac{1}{q}. \end{aligned}$$

Now, sum the other $(q-1)$ equations for $j \neq i$ and equate that to $(q-1)$ times the left hand side of the above equation since both of their values evaluate to $\left(\frac{q-1}{q}\right)$:

$$\begin{aligned} &\Rightarrow (q-1)\left[p_i(1-\epsilon-\beta)+(1-p_i)\ \left(\frac{\beta}{q-1}\right)\right] \\ &\qquad\qquad = \sum_{j\neq i}\left(p_i(1-\epsilon-\beta)+(1-p_i)\left(\frac{\beta}{q-1}\right)\right) \\ \Rightarrow\qquad & (q-1)p_i(1-\epsilon)+\beta-qp_i\beta = (1-p_i)(1-\epsilon)+\beta-(1-p_i)\left(\frac{q\beta}{q-1}\right) \\ \Rightarrow\qquad & qp_i(1-\epsilon)-\left(qp_i\beta+\frac{qp_i\beta}{q-1}\right) = (1-\epsilon)-\frac{q\beta}{q-1} \\ \Rightarrow\qquad & (qp_i-1)\left(1-\epsilon-\frac{q\beta}{q-1}\right) = 0. \end{aligned}$$

Thus, $p_i = \frac{1}{q}$ maximizes $H(Y)$ to give

$$\begin{aligned} H(Y) &= -\left[(1-\epsilon-\beta+\beta)\,\log_q\!\left(\frac{1-\epsilon-\beta+\beta}{q}\right)+\epsilon\,\log_q\epsilon\right] \\ &= (1-\epsilon)+h_q(\epsilon), \end{aligned} \tag{D.3}$$

where

$$h_q(\epsilon) = -[\epsilon\,\log_q(\epsilon)+(1-\epsilon)\,\log_q(1-\epsilon)].$$

Substituting values of $H(Y|X)$ and $H(Y)$ obtained in (D.2) and (D.3), respectively, into (D.1), we get the capacity of QSCE as

$$C = (1-\epsilon)+(1-\epsilon)\,\log_q\!\left(\frac{1-\epsilon-\beta}{1-\epsilon}\right)-\beta\,\log_q\!\left(\frac{1-\alpha-\beta}{\beta}\right)-\beta\,\log_q(q-1). \tag{D.4}$$