TECHNICAL UNIVERSITY OF CRETE

Department of Electrical and Computer Engineering

Polar Codes for the Binary or Q-ary Symmetric Channels

by

Michail Nodarakis

A thesis submitted in partial fulfillment of the requirements for the Diploma degree of Electrical and Computer Engineering

Thesis Supervisor

Professor GEORGE N. KARYSTINOS

Committee Members

Professor AGGELOS BLETSAS

Professor ATHANASIOS P. LIAVAS

October 2022

Abstract

Polar coding is the first channel coding technique that provably achieves the highest rate at which information can reliably be sent over a communication channel, known as channel capacity. Polar codes utilize the effect of channel polarization to generate a set of N channels out of N independent copies of a given channel W. These channels are either perfect (symmetric capacity I(W) = 1) or extremely poor (symmetric capacity I(W) = 0).

The main quest of channel polarization is to find out which channels are perfect in order to transmit data. In this thesis, we first describe the essential concepts of polar codes and the basic polar encoder and decoder suggested by Arikan.

Then, we analyze a method for the channel selection in the case of the binary symmetric channel (BSC) and evaluate its performance.

Finally, we expand channel polarization to Q-ary input channels and evaluate a technique of channel polarization for arbitrary discrete memoryless channels. Especially for the ternary symmetric channel, we propose some modifications to the existing algorithm.

Acknowledgements

I would like to thank my family and friends for their unconditional support during my studies.
Furthermore, I would like to express my deepest gratitude to my supervisor, Professor Georgios Karystinos, who guided me throughout this thesis.
Finally, I would like to express my appreciation to my thesis committee, Professor Aggelos Bletsas and Professor Athanasios Liavas.

Table of Contents

List of Figures

Chapter 1

Introduction

Claude E. Shannon's 1948 paper [7], "A Mathematical Theory of Communication" published in the Bell Systems Technical Journal, effectively established information theory. In this paper, Shannon mathematically defined the highest rate at which information can reliably be sent over a communication channel, known as channel capacity. Furthermore, he defined the fundamental components of a communication system as in figure 1.1 which shows the process by which a message is sent by the transmitter and received by the receiver with the possible input of noise. For the next 2 decades, the field of information theory was widely accepted and applied until the 70s, when for about 20 years the interest significantly declined. In the 90s, remarkable progress was made with the development of capacity-approaching codes (Turbo codes, LDPC).

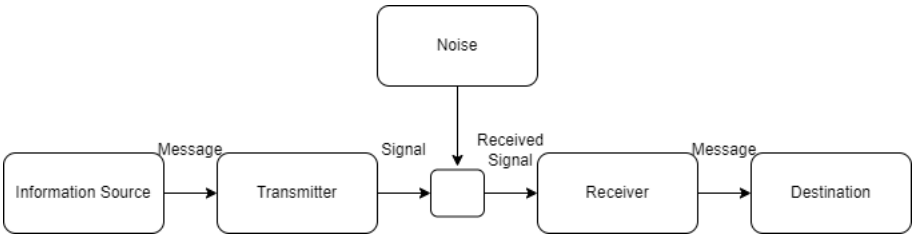


Figure 1.1: Block diagram of a communication system

Polar Codes were introduced in 2009 by Erdal Arikan and it was the first coding technique to provably achieve channel capacity for the binary input discrete memoryless channel. In addition, polar codes had lower complexity compared to previous coding techniques. Polar code utilizes the method of channel polarization, which converts channels (codeword bits) into either very good or very bad channels. As

a result, we can send information through the good channels while setting the bad channels to a specific value known to the decoder.

In this thesis, we will focus on channel polarization of Binary Discrete Memoryless channels (B-DMCs), namely on the Binary Erasure Channel (BEC) and the Binary Symmetric Channel (BSC). We will also describe a low-complexity Successive Cancellation (SC) decoder and simulate the performance results of polar coding under SC decoding for each channel at various block lengths.
Finally, we will generalize channel polarization for the case of q-ary Discrete Memoryless Channels channels and specifically the case of Ternary Symmetric Channel(TSC).

Chapter 2

Polarization of Binary-Discrete Memoryless Channels

According to Arikan [1], channel polarization is a technique for converting N independent copies of a B-DMC W into a polarized set of N binary input channels $\{W_N^{(i)} : 1 \leq i \leq N\}$ such that as N increases, there are indices near 1 that approach capacity $I(W)$, those near 0 that approach $1 - I(W)$ and a small fraction of mediocre channels.

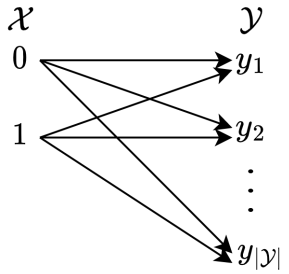


Figure 2.1: Binary Discrete Memoryless Channel

2.1 Channel Parameters

To begin with, two critical parameters must be defined to measure rate and reliability: symmetric capacity and the Bhattacharyya parameter.

2.1.1 Symmetric Capacity

Given a B-DMC $W : \mathcal{X} \to \mathcal{Y}$ with input alphabet $\mathcal{X} = \{0, 1\}$ and output alphabet $\mathcal{Y}$, the symmetric capacity is defined as:

$$I(W) \triangleq \sum_{y \in \mathcal{Y}} \sum_{x \in \mathcal{X}} \frac{1}{2} W(y|x) \log \frac{W(y|x)}{\frac{1}{2}W(y|0) + \frac{1}{2}W(y|1)} \tag{2.1}$$

It is used as a measure of the rate and it is essentially the mutual information between the input and output of the channel when the input is a uniform distribution.

Binary Symmetric Channel

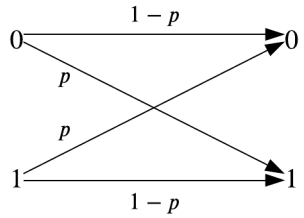
The Binary Symmetric Channel with input alphabet $\mathcal{X} = \{0, 1\}$, output alphabet $\mathcal{Y} = \{0, 1\}$ and crossover probability p has the following symmetric capacity

$$I(W) = 1 + p \log_2(p) + (1 - p) \log_2(1 - p) \tag{2.2}$$

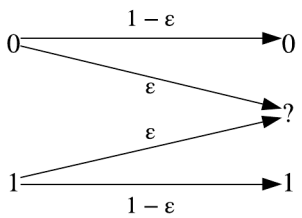
Binary Erasure Channel

The symmetric capacity of the Binary Erasure Channel with input alphabet $\mathcal{X} = \{0, 1\}$, output alphabet $\mathcal{Y} = \{0, 1, ?\}$ and erasure probability ε is

$$I(W) = 1 - \varepsilon \tag{2.3}$$



(a) BSC



(b) BEC

2.1.2 Bhattacharyya parameter

The Bhattacharyya parameter $Z(W)$ represents an upper bound on the probability of maximum-likelihood decision error and has values in the range [0 1]. Bhattacharyya parameter is defined as follows:

$$Z(W) \triangleq \sum_{y \in \mathcal{Y}} \sqrt{W(y|0)W(y|1)} \tag{2.4}$$

It is used as a measure of reliability. For the BSC, the Bhattacharyya parameter is:

$$Z(W) = 2\sqrt{p(1-p)} \tag{2.5}$$

For the BEC, the Bhattacharyya parameter is:

$$Z(W) = \varepsilon \tag{2.6}$$

The relation between the symmetric capacity I(W) and the Bhattacharyya parameter Z(W) is given by:

$$I(W) \geq \log \frac{2}{1 + Z(W)} \tag{2.7}$$

$$I(W) \leq \sqrt{1 - Z(W)^2} \tag{2.8}$$

2.2 Construction

To understand the concept of channel polarization, it is necessary to first define its two phases, which are the channel combining phase and channel splitting phase.

2.2.1 Channel Combining

Channel combining refers to the procedure where N independent copies of a given B-DMC W are combined recursively to form a vector channel $W_N = \mathcal{X}^N \rightarrow \mathcal{Y}^N$ where $N = 2^n$, $n \geq 0$.

Figure 2.3 depicts the construction of $W_4 : \mathcal{X}^2 \rightarrow \mathcal{Y}^2$ by recursively combining two independent copies of W_2, which is also constructed by combining two independent copies of W. Furthermore, block R_4 represents a permutation operation that separates odd-indexed with even-indexed input, in the case of W_4 it maps the input (s_1, s_2, s_3, s_4) to (s_1, s_3, s_2, s_4).

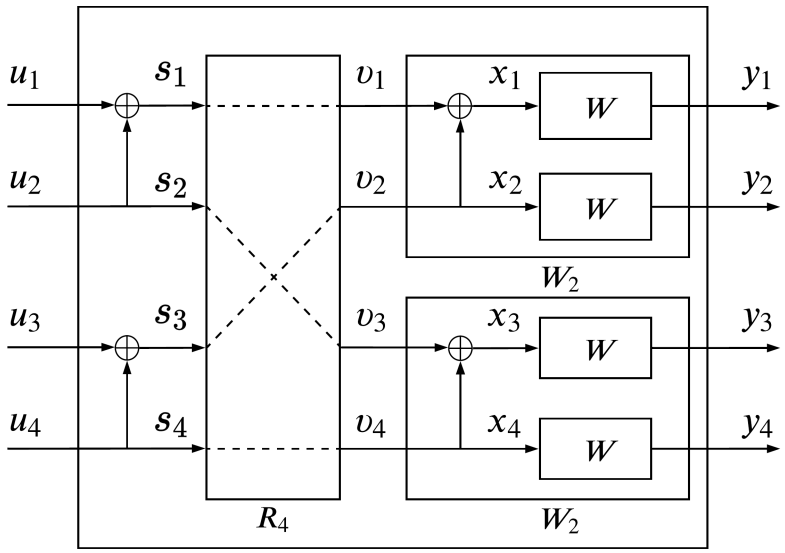


Figure 2.3: Construction of W_4

It becomes apparent that since mapping $u_1^N \mapsto v_1^N$ is linear over GF(2), the mapping of the input of the constructed channel W_N to the input of the underlying raw channels W^N is also linear and can be carried out over the binary field GF(2) in the sense that $x_1^N = u_1^N G_N$.

G_N is the generator matrix of polar codes and it is defined as $G_N = B_N F^{\otimes n}$, where B_N is the bit reversal permutation matrix $B_N = R_N(I_2 \otimes B_{N/2})$ and $F^{\otimes n}$ is the n Kronecker product of the matrix $F = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$.
For the case of W_4 where $N = 2^2$, we have

$$F^{\otimes}2 = \begin{bmatrix} 1\begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} & 0\begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \\ 1\begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} & 1\begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix} \text{ and}$$

$$G_4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

Hence, the construction of the vector channel is given from the relation

$$W_N(y_1^N|u_1^N) = W^N(y_1^N|u_1^N G_N) = W^N(y_1^N|x_1^N) \tag{2.9}$$

for $y_1^N \in \mathcal{Y}^N, u_1^N \in \mathcal{X}^N$.

2.2.2 Channel Spliting

Channel splitting refers to the procedure where the vector channel W_N is split back to N channels $W_N^{(i)} : \mathcal{X} \to \mathcal{Y}^N \times \mathcal{X}^{i-1}$ and it is defined by the following transition probabilities

$$W_N^{(i)}(y_1^N, u_1^{i-1}|u_i) \triangleq \sum_{u_{i+1}^N \in \mathcal{X}^{N-i}} \frac{1}{2^{N-1}} W_N(y_1^N|u_1^N) \tag{2.10}$$

In the case of the polarization of BEC, the symmetric capacities can be easily calculated by the following recursive relations:

$$\begin{aligned} I(W_N^{(2i-1)}) &= I(W_{N/2}^{(i)})^2 \\ I(W_N^{(2i)}) &= 2I(W_{N/2}^{(i)}) - I(W_{N/2}^{(i)})^2 \end{aligned} \tag{2.11}$$

where $I(W) = 1 - \varepsilon$.

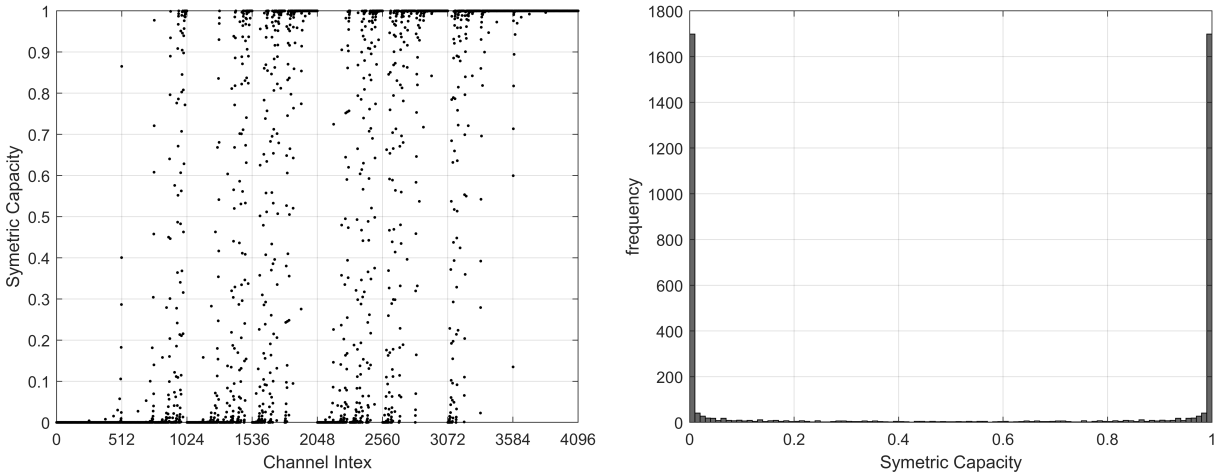


Figure 2.4: BEC($\varepsilon = \frac{1}{2}$) channel polarization for $N = 2^{12}$

The plot and histogram show that channel polarization of BEC with erasure probability e = 0.5 results in half of the channels being perfect with symmetric capacity 1, half of the channels being useless with symmetric capacity 0, and a small percentage of mediocre channels.

2.2.3 Channel, Rate and Reliability Transformations

Given a single binary-input channel $W : \mathcal{X} \to \mathcal{Y}$ we can perform a single step transformation $(W, W) \mapsto (W^-, W^+)$ and get two binary-input channels $W^- : \mathcal{X} \to \tilde{\mathcal{Y}}$ and $W^+ : \mathcal{X} \to \tilde{\mathcal{Y}} \times \mathcal{X}$, iff there exist one-to-one mapping $f : \mathcal{Y}^2 \to \tilde{\mathcal{Y}}$ such that

$$W^-(f(y_1, y_2)|u_1) = \sum_{u_2'} \frac{1}{2} W(y_1|u_1 \oplus u_2')W(y_2|u_2') \tag{2.12}$$

$$W^+(f(y_1, y_2), u_1|u_2) = \frac{1}{2} W(y_1|u_1 \oplus u_2)W(y_2|u_2) \tag{2.13}$$

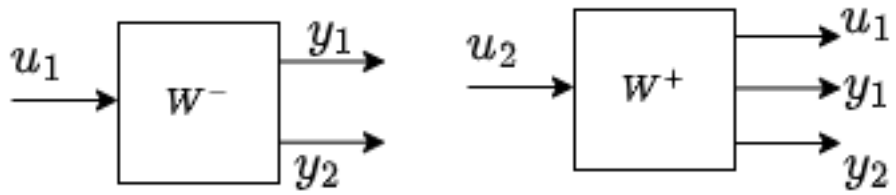


Figure 2.5: Polarized channels

and if we take f as the identity mapping, we can obtain $(W, W) \mapsto (W_2^-, W_2^+)$ such that

$$W_2^-(y_1^2|u_1) = \sum_{u_2} \frac{1}{2} W(y_1|u_1 \oplus u_2)W(y_2|u_2) \tag{2.14}$$

$$W_2^+(y_1^2, u_1|u_2) = \frac{1}{2} W(y_1|u_1 \oplus u_2)W(y_2|u_2) \tag{2.15}$$

In addition, it is important to mention the properties of rate and reliability. Given a single binary-input channel $(W, W) \mapsto (W^-, W^+)$ Then

$$I(W^-) + I(W^+) = 2I(W) \tag{2.16}$$

$$I(W^-) \leq I(W^+) \tag{2.17}$$

The equality stands for when I (W) equals 0 or 1.
Those two relations highlight the preserving properties of the symmetric capacity under single-step channel transformation.

Furthermore, given the single-step channel transform $(W, W) \mapsto (W^-, W^+)$

$$Z(W^+) = Z(W)^2 \tag{2.18}$$

$$Z(W^-) \leq 2Z(W) - Z(W)^2 \tag{2.19}$$

$$Z(W^-) \geq Z(W) \geq Z(W^+) \tag{2.20}$$

The equality in (2.19) stands for when W is a BEC. This results that under single-step channel transform, the reliability can either stay the same iff W is BEC or improve. All of the above-mentioned relations can expand for $W_N^{(i)}$

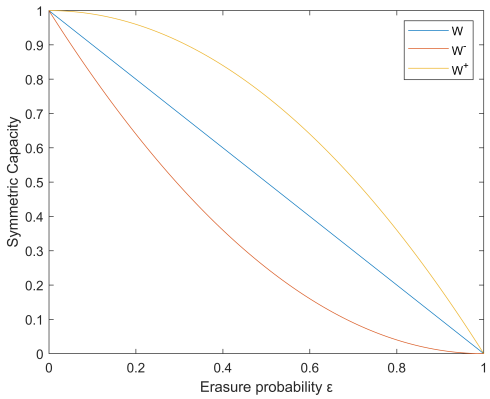


Figure 2.6: Symmetric Capacity vs Erasure probability of the original and the polarized BEC

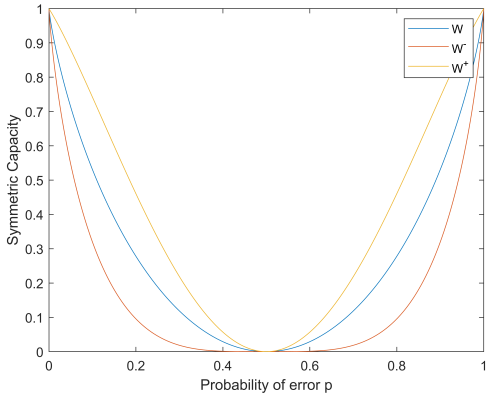


Figure 2.7: Symmetric Capacity vs Error probability of the original and the polarized BSC

It is obvious that for both the BEC and the BSC, the symmetric capacity of the degraded channel W^- is smaller than before the polarization, and the symmetric capacity of the upgraded channel W^+ is greater.

2.3 Encoding

The matrix multiplication method outlined in channel combining (2.9) is convenient for small values of N, but as N grows, so does the complexity. Another implementation of the encoder is rather straightforward, as we must construct the W_N as shown in figure 2.8.

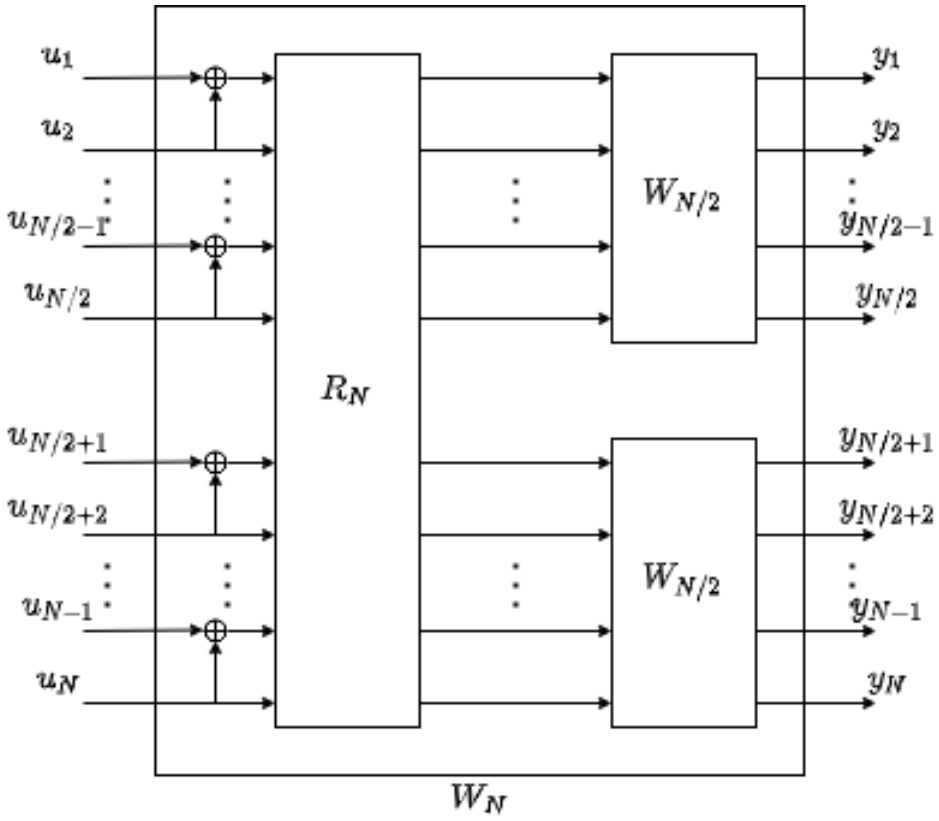


Figure 2.8: Recursive construction of W_N from two copies of $W_N/2$

Having made the selection of the information set A, we assign the message to the channels $u_A \in \mathcal{X}^K$ and the remaining channels $u_{A^c} \in \mathcal{X}^{N-K}$ are frozen and set to 0.

The reverse shuffle operation R_N separates the odd-indexed channels, which are fed into the first copy $W_{N/2}$, from the even-indexed channels, which are fed into the second copy $W_{N/2}$. This process is implemented recursively until the channel W_2 which is comprised of two independent copies of W_1, the B-DMC which outputs y_1^N becomes the input to the decoder.

The polar coding transform graph example in Figure 2.9 shows that the encoding process has $\log_2(N)$ levels with N channels at each level, implying that the encoding complexity is $O(N \log_2 N)$.

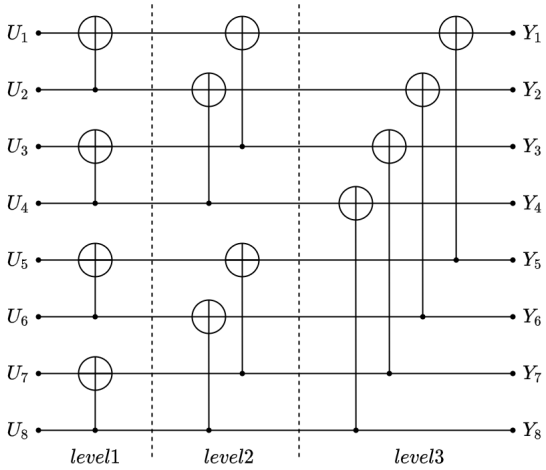
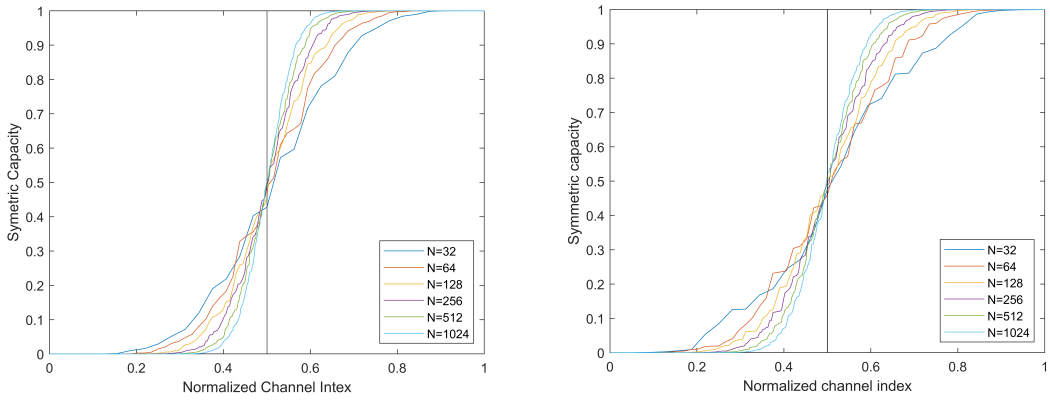


Figure 2.9: Graph of the polar encoding for N=8

As the block length increases more channels are going to polarize, thus, the probability of error is going to get smaller and as we can see in figure 2.10, more channels are either perfect with a symmetric capacity near 1 or useless with a symmetric capacity near 0.



(a) BEC (b) BSC

Figure 2.10: I(W) versus Channel index for block lengths $2^5...2^{10}$

2.4 Decoding

The technique used to decode polar codes, as suggested by Arikan in [1], is known as a successive cancellation decoder. Based on the observed channel output y_1^N and knowledge of the frozen bits u_{A^c}, this method gives an estimate $\hat{u}_1^N$ of u_1^N. The decoding is sequential in the sense that the calculation of probabilities makes use of the previously estimated bits (fig 2.11), and it is accomplished using the recursive formulae listed below.

$$W_{2N}^{(2i-1)}(y_1^{2N}, u_1^{2i-2}|u_{2i-1}) = \sum_{u_{2i}} \frac{1}{2} W_N^{(i)}(y_1^N, u_{1,o}^{2i-2} \oplus u_{1,e}^{2i-2}|u_{2i-1} \oplus u_{2i}) \cdot W_N^{(i)}(y_{N+1}^{2N}, u_{1,e}^{2i-2}|u_{2i}) \tag{2.21}$$

$$W_{2N}^{(2i)}(y_1^{2N}, u_1^{2i-1}|u_{2i}) = \frac{1}{2} W_N^{(i)}(y_1^N, u_{1,o}^{2i-2} \oplus u_{1,e}^{2i-2}|u_{2i-1} \oplus u_{2i}) \cdot W_N^{(i)}(y_{N+1}^{2N}, u_{1,e}^{2i-2}|u_{2i}) \tag{2.22}$$

and the decision is made according to the following rule

$$\hat{u}_i = \begin{cases} 0 & \text{if } u_i \in A^c \\ argmax_{x \in (0,1)} W_N^{(i)}(y_1^N, u_1^{i-1}|x) & \text{otherwise} \end{cases} \tag{2.23}$$

For the Binary input channel $W : \mathcal{X} \to \mathcal{Y}$ with $\mathcal{X} = 0, 1$ and $\mathcal{Y}$ arbitrary, we can also use the likelihood ratio

$$L_N^{(i)}(y_1^N, \hat{u}_1^{i-1}) \triangleq \frac{W_N^{(i)}(y_1^N, \hat{u}_1^{i-1}|0)}{W_N^{(i)}(y_1^N, \hat{u}_1^{i-1}|1)} \tag{2.24}$$

With decision rule

$$\hat{u}_i = \begin{cases} 0 & \text{if } L_N^{(i)}(y_1^N, \hat{u}_1^{i-1}) \geq 1 \\ 1 & \text{otherwise} \\ 0 & \text{if } u_i \in A^c \end{cases} \tag{2.25}$$

The following recursive formulas can be used to estimate the $L_N^{(i)}(y_1^N, \hat{u}_1^{i-1})$

$$L_N^{(2i-1)}(y_1^N, \hat{u}_1^{2i-2}) = \frac{L_{N/2}^{(i)}(y_1^{N/2}, \hat{u}_{1,o}^{2i-2} \oplus \hat{u}_{1,e}^{2i-2}) L_{N/2}^{(i)}(y_{N/2+1}^N, \hat{u}_{1,e}^{2i-2}) + 1}{L_{N/2}^{(i)}(y_1^{N/2}, \hat{u}_{1,o}^{2i-2} \oplus \hat{u}_{1,e}^{2i-2}) L_{N/2}^{(i)}(y_{N/2+1}^N, \hat{u}_{1,e}^{2i-2})} \tag{2.26}$$

$$L_N^{(2i)}(y_1^N, \hat{u}_1^{2i-1}) = \left[L_{N/2}^{(i)}(y_1^{N/2}, \hat{u}_{1,o}^{2i-2} \oplus \hat{u}_{1,e}^{2i-2}) \right]^{1-2\hat{u}_{2i-1}} \cdot L_{N/2}^{(i)}(y_{N/2+1}^N, \hat{u}_{1,e}^{2i-2}) \tag{2.27}$$

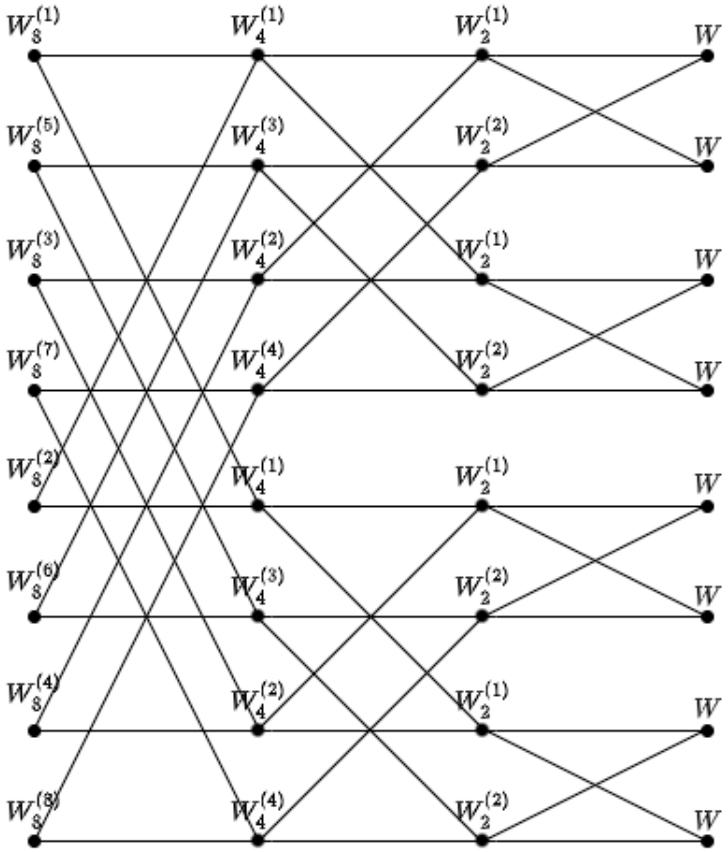


Figure 2.11: Channel transformation for N=8

We can see from the recursive formulas that some calculations are repeated. For example, once we've calculated formula (2.12), we're ready to calculate formula (2.13) too. To obtain a better approach and decrease complexity, we store every calculation in two matrices of size $N \times (\log_2 N + 1)$ and reuse them rather than computing them again.

2.5 BEC simulation

To determine the performance of polar coding on the BEC under SC decoding, we define channel W as in BEC figure . For rate = 1/2 and block lengths ranging from 2^5 to 2^9, we run 5000 experiments for each value of erasure probability e in order to obtain the BER in relation to the erasure probability e.

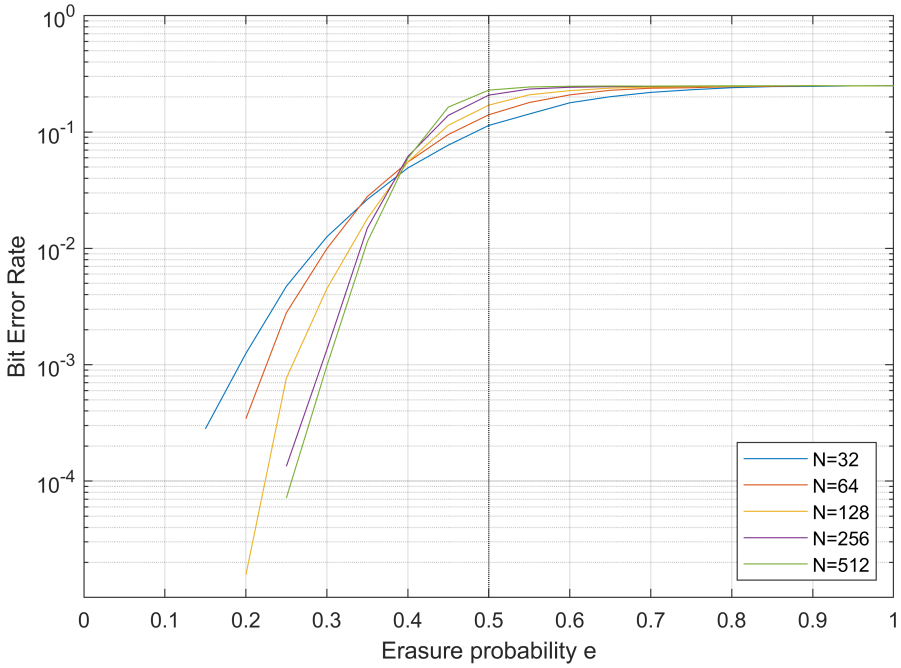


Figure 2.12: erasure probability versus BER for polar coding and SC decoding at block lengths $2^5 ... 2^9$ on a BEC with rate 1/2.

Figure 2.12 indicates, as expected, that as the block length increases, the BER decreases since polar codes approach capacity when the block length increases to infinity.

Chapter 3

Polar Codes Construction

We have already described the nature of polar codes and the creation of a low-complexity encoder and decoder. However, deciding on the information set A is critical in determining if the construction is also efficient. In the case of the binary erasure channel (BEC), choosing the information set is straightforward since we can easily compute the symmetric capacity I(W) of each channel using (2.11), and then choose the K channels with the largest capacity as the information set A. In the case of the binary symmetric channel (BSC), however, it is rather challenging since there are no recursive relations to calculate the symmetric capacity I(W).

One of the first attempts at an efficient construction of polar codes was made by Mori and Tanaka [3] with linear complexity in the blocklength, making this approach unfeasible. In their paper [8], Tal and Vardy proposed an efficient method to determine the information set and thus construct polar codes. This method uses two approximations to obtain a upper bound (degrading approximation) and an lower bound (upgrading approximation) on the probability of error for each bit-channel. These approximations make use of a parameter μ, which specifies the accuracy of the approximation since it is the size of the output alphabet. It is evident that for bigger values of μ, the bounds are closer to the actual probability of error, but the complexity of the approximation of n bit-channels grows. $O(n \cdot \mu^2 log\mu)$.

3.1 Preliminaries

In order to fully understand the proposed solution, we must first define the stochastically degraded and stochastically upgraded channels.
According to [8]

Assuming two channels $\mathcal{Q} : \mathcal{X} \to \mathcal{Z}$ and $\mathcal{W} : \mathcal{X} \to \mathcal{Y}$, we say that channel $\mathcal{Q}$ is stochastically degraded with respect to $\mathcal{W}$, $\mathcal{Q} \preccurlyeq \mathcal{W}$, if there exist a channel $\mathcal{P} : \mathcal{Y} \to \mathcal{Z}$ such that

$$\mathcal{Q}(z|x) = \sum_{y \in \mathcal{Y}} \mathcal{W}(y|x)\mathcal{P}(z|y) \tag{3.1}$$

for all $z \in \mathcal{Z}$ and $x \in \mathcal{X}$.
Similarly,

Assuming two channels $\mathcal{Q}' : \mathcal{X} \to \mathcal{Z}'$ and $\mathcal{W} : \mathcal{X} \to \mathcal{Y}$, we say that channel $\mathcal{Q}$ is stochastically upgraded with respect to $\mathcal{W}$, $\mathcal{Q}' \succcurlyeq \mathcal{W}$, if there exist a channel $\mathcal{P} : \mathcal{Z}' \to \mathcal{Y}$ such that

$$\mathcal{W}(y|x) = \sum_{z' \in \mathcal{Z}'} \mathcal{Q}(z'|x)\mathcal{P}(y|z') \tag{3.2}$$

for all $z' \in \mathcal{Z}'$ and $x \in \mathcal{X}$.
Apparently,

$$\mathcal{Q}' \succcurlyeq \mathcal{W} \text{ iff } \mathcal{W} \preccurlyeq \mathcal{Q}' \tag{3.3}$$

Furthermore, it can be seen that if $\mathcal{W}'$ is both degraded and upgraded with respect to $\mathcal{W}$, the $\mathcal{W}$ and $\mathcal{W}'$ are equivalent, $\mathcal{W} \equiv \mathcal{W}'$, and since $\preccurlyeq$ and $\succcurlyeq$ are reflexive relations

$$\mathcal{W} \preccurlyeq \mathcal{W} \text{ and } \mathcal{W} \succcurlyeq \mathcal{W} \tag{3.4}$$

and transitive relations

$$\text{if } \mathcal{W} \preccurlyeq \mathcal{W}' \text{ and } \mathcal{W}' \preccurlyeq \mathcal{W}'', \text{ then } \mathcal{W} \preccurlyeq \mathcal{W}'' \tag{3.5}$$

Therefore, $\equiv$ is also a transitive relation, according to (3.3) it is a symmetric relation and according to (3.4) it is reflexive as well.

In addition, assuming a binary input memoryless symmetric channel (BMS) $\mathcal{W} : \mathcal{X} \to \mathcal{Y}$, and a channel $\mathcal{Q} : \mathcal{X} \to \mathcal{Z}$ that is degraded with respect to $\mathcal{W}$ then according to [8]

$$P_e(\mathcal{Q}) \geq P_e(\mathcal{W}) \tag{3.6}$$

$$Z(\mathcal{Q}) \geq Z(\mathcal{W}) \tag{3.7}$$

$$I(\mathcal{Q}) \leq I(\mathcal{W}) \tag{3.8}$$

If the inequalities are reversed, the above relations hold for the upgraded channel.
It is also known that the probability of error of a BMS channel under a maximum-likelihood decision is

$$P_e(\mathcal{W}) = \frac{1}{2} \sum_{y \in \mathcal{Y}} min\{\mathcal{W}(y|0), \mathcal{W}(y|1)\} \tag{3.9}$$

The proof is provided in the Appendix.

Finally, assuming a binary input memoryless symmetric channels (BMS) $\mathcal{W} : \mathcal{X} \to \mathcal{Y}$, the likelihood ratio of an output symbol $y \in \mathcal{Y}$ is given by

$$LR_{\mathcal{W}}(y) = \frac{\mathcal{W}(y|0)}{\mathcal{W}(y|1)} = \frac{\mathcal{W}(y|0)}{\mathcal{W}(\tilde{y}|0)} \tag{3.10}$$

and at least one of the probabilities $\mathcal{W}(y|0) = \mathcal{W}(\tilde{y}|1)$ or $\mathcal{W}(y|1) = \mathcal{W}(\tilde{y}|0)$ is greater than zero.
If the denominator of the likelihood ratio is zero then $LR_{\mathcal{W}}(y) = \infty$.

3.2 Bit-Channel approximations

According to [8], to obtain the degrading or upgrading approximation of each bit-channel, we provide the underlying BMS channel, the fidelity parameter, and the binary representation of the channel index as input to an algorithm that performs Arikans' recursive transformations:

$$\mathcal{W}^{-}(y_1, y_2|u_1) = \frac{1}{2} \sum_{u_2 \in \mathcal{X}} \mathcal{W}(y_1|u_1 \oplus u_2)\mathcal{W}(y_2|u_2) \tag{3.11}$$

$$\mathcal{W}^{+}(y_1, y_2, u_1|u_2) = \frac{1}{2}\mathcal{W}(y_1|u_1 \oplus u_2)\mathcal{W}(y_2|u_2) \tag{3.12}$$

One issue with using recursive transformations is the massive increase in the output alphabet size. Thus, after each iteration, we call the degrading merge or upgrading merge function in order to reduce the output alphabet size to at most μ. The algorithm proposed in [8] for the Bit-channel degrading procedure is the following

Given a BMS channel $W : \mathcal{X} \to \mathcal{Y}$ with output alphabet $\mathcal{Y} = \{y_1, ..., y_L, \tilde{y_1}, ..., \tilde{y_L}\}$, we choose from each pair $(y, \tilde{y})$ a representative such that

$$1 \leq LR(y_1) \leq LR(y_2) \leq ... \leq LR(y_L). \tag{3.13}$$

Algorithm 1: Bit-channel degrading procedure

Input: An underlying BMS channel W, a bound μ=2ν on the output alphabet size, a code length $N = 2^m$, and an index i with binary representation $i = \langle b_1, b_2, ..., b_m \rangle_2$.

Output: An upper bound on the probability of error of W_i $P_e(W_i)$.

$\mathbf{Z} \leftarrow Z(W)$

$\mathcal{Q} \leftarrow$ degrading_merge(W, μ)

for *j = 1,2,...,m* **do**

  if $b_j = 0$ **then**

    $\mathcal{W} \leftarrow \mathcal{Q}^-$

    $\mathbf{Z} \leftarrow min\{Z(\mathcal{W}), 2\mathbf{Z} - \mathbf{Z}^2\}$

  else

    $\mathcal{W} \leftarrow \mathcal{Q}^+$

    $\mathbf{Z} \leftarrow \mathbf{Z}^2$

  $\mathcal{Q} \leftarrow$ degrading_merge$(\mathcal{W}, \mu)$

return $min\{P_e(\mathcal{Q}), \boldsymbol{Z}\}$

Where Z(W) is the Bhattachayya parameter of the BMS channel W and it is obtained by

$$Z(W) = \sum_{y \in \mathcal{Y}} \sqrt{W(y|0)W(y|1)}. \tag{3.14}$$

The same algorithm holds for the Bit-channel upgrading procedure if we change *degrading_merge* with *upgrading_merge*.

It is evident that if the *degrading_merge* and *upgrading_merge* have a time complexity of τ=τ(μ) then due to the for loop, this algorithm has a complexity of O(mτ). In order to approximate all N bit-channels the time complexity would be O(Nmτ), but as noted in [8] many transformations can be avoided because they are shared between bit-channels.

For example, given N = 16, the bit-channels with index ⟨0000⟩ and ⟨0001⟩ have three transformations $\mathcal{Q}^-$ that are shared, or ⟨0000⟩and ⟨0010⟩ have two transformations $\mathcal{Q}^-$ that are shared.

Upon observing this, we decided to save the transformation result for every combination of indexes b_j with $j = 1, .., m$ when it first occurs and then recall it from memory

when it reoccurs rather than calculating it again. The time complexity of the smart implementation is O(Nτ).

3.2.1 Degrading merge

As proved in [8], given a BSM channel $\mathcal{W} : \mathcal{X} \to \mathcal{Y}$ and let the symbols that we want to merge be $y_1, y_2 \in \mathcal{Y}$. We define $\mathcal{Q} : \mathcal{X} \to \mathcal{Z}$ with the output alphabet

$$\mathcal{Z} = \mathcal{Y} \setminus \{y_1, \tilde{y}_1, y_2, \tilde{y}_2\} \cup \{z_{1,2}, \tilde{z}_{1,2}\}$$

for all $x \in \mathcal{X}$ and $z \in \mathcal{Z}$.

Then the degraded $\mathcal{Q}$ with respect to $\mathcal{W}$, $\mathcal{Q} \preccurlyeq \mathcal{W}$, is given by

$$\mathcal{Q}(z|x) = \begin{cases} \mathcal{W}(z|x) & \text{if } z \notin \{z_{1,2}, \tilde{z}_{1,2}\}, \\ \mathcal{W}(y_1|x) + \mathcal{W}(y_2|x) & \text{if } z = z_{1,2}, \\ \mathcal{W}(\tilde{y}_1|x) + \mathcal{W}(\tilde{y}_2|x) & \text{if } z = \tilde{z}_{1,2}. \end{cases} \tag{3.15}$$

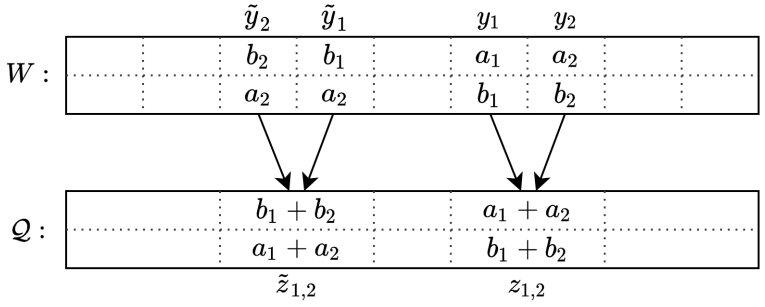


Figure 3.1: The degrading merge operation

It is also worth noting that when y_i and y_{i+1} are merged, the LR order is preserved, since $LR(y_i) \leq LR(z) \leq LR(y_{i+1})$. The output alphabet size of the degraded channel $\mathcal{Q}$ decreases by 2 after each iteration of the degrading merge. Following that, the degrading merge algorithm proposed in [8] is defined as follows:

Algorithm 2: The degrading_merge function

Input: A BMS channel $\mathcal{W} : \mathcal{X} \leftarrow \mathcal{Y}$ where $|\mathcal{Y}| = 2L$, a bound μ=2ν on the output alphabet size.

Output: A degraded channel $\mathcal{Q} : \mathcal{X} \rightarrow \mathcal{Y}'$, where $|\mathcal{Y}'| \leq \mu$.

Assume $1 \leq LR(y_1) \leq LR(y_1) \leq ... \leq LR(y_L)$

if $L \leq \nu$ **then**
  return $\mathcal{W}$

for *i = 1,2,...,L-1* **do**
  $d \leftarrow$ new data elements
  $d.a \leftarrow \mathcal{W}(y_i|0), \quad d.b \leftarrow \mathcal{W}(\tilde{y}_i|0)$
  $d.a' \leftarrow \mathcal{W}(y_{i+1}|0), \quad d.b' \leftarrow \mathcal{W}(\tilde{y}_{i+1}|0)$
  $d.deltaI \rightarrow$ calcDeltaI$(d.a, d.b, d.a', d.b')$
  insertRightmost(d)

$l = L$

while $l > \nu$ **do**
  $d \leftarrow$ getMin()
  $a^+ = d.a + d.a', \quad b^+ = d.b + d.b'$
  dLeft $\rightarrow$ d.left
  dRight $\rightarrow$ d.right
  removeMin()
  $l \rightarrow l - 1$
  if $dLeft \neq null$ **then**
    dLeft.$a' = a^+$
    dLeft.$b' = b^+$
    dLeft.deltaI = calcDeltaI(dLeft.a, dLeft.b, a^+, b^+)
    valueUpdated(dLeft)
  if $dRight \neq null$ **then**
    dRight.$a = a^+$
    dRight.$b = b^+$
    dRight.deltaI = calcDeltaI(, a^+, b^+, dRight.a', dRight.b')
    valueUpdated(dRight)

Construct $\mathcal{Q}$ according to the probabilities in the data structure **return** $\mathcal{Q}$

The degrading_merge is implemented using a data structure that stores

a, b, a', b', deltaI, dleft, dRight, h

a , b, a', b' store the probabilities $\mathcal{W}(y_i|0), \mathcal{W}(\tilde{y}_i|0), \mathcal{W}(y_{i+1}|0), \mathcal{W}(\tilde{y}_{i+1}|0)$, deltaI store the return value of the function calcDeltaI, dLeft is a pointer to the previous pair (y_{i-1}, y_i) and dRight is a pointer to the next pair (y_{i+1}, y_{i+2}).

Effectively, we are looking for a pair of y_i, y_{i+1} that upon applying the degrading merge, the difference in capacity is the minimum among all pairs (getMin). The resulting difference in capacity is obtained by

$$\textit{calcDeltaI(a,b,a',b')} = \textit{C(a,b)} + \textit{C(a',b')} - C(a^+, b^+) \tag{3.16}$$

where

$$\textit{C(a,b)} = -(a+b)\log_2((a+b)/2) + a\log_2(a) + b\log_2(b) \tag{3.17}$$

with $0 \log_2 0$ defined as 0.

The function removeMin() removes the element of the pair returned by getMin(), from the data structure.

Obviously, after a merge is accomplished, the parameter deltaI of the pairs (y_{i-1}, y_i) and (y_{i+1}, y_{i+2}) must be recalculated with the parameters of the new symbol z instead of y_i and y_{i+1}. Thus, we check if the previous element in the array is not null to update its deltaI and its field of *a', b', dRight* with valueUpdated(dLeft). Also, if the next element in the array is not null, we update its deltaI and its field of *a, b, dLeft* accordingly, with valueUpdated(dRight).

The complexity of the merge function is $O(L \log L)$ and since transformation (3.12) grows the alphabet size to at most $2\mu^2$ and $|y| = 2L$, the complexity is translated to $O(\mu,^2 \log \mu^2)$.

3.2.2 Upgrading merge

As in the case of the degrading_merge, where we merged pairs of symbols to obtain a channel that is degraded with respect to the original channel. We can also merge pairs of symbols with the upgrading_merge function and obtain a channel that is upgraded with respect to the original channel. As a result, we get a lower bound (the upgraded channel) and an upper bound (the degraded channel) on the probability of error for each bit-channel. In reality, even for tiny values of the fidelity parameter μ, the two boundaries are typically very close.

Method 1: As proved in [8], given a BSM channel $\mathcal{W} : \mathcal{X} \to \mathcal{Y}$ and let the symbols that we want to merge be $y_2, y_1 \in \mathcal{Y}$. We denote $\lambda_2 = LR(y_2)$ and $\lambda_1 = LR(y_1)$, assuming that

$$1 \leq \lambda_1 \leq \lambda_2.$$

Also, let $a_1 = \mathcal{W}(y_1|0)$ and $b_1 = \mathcal{W}(\tilde{y}_1|0)$. We define α_2 and β_2 as follows.

If $\lambda_2 < \infty$

$$\alpha_2 = \lambda_2 \frac{a_1 + b_1}{\lambda_2 + 1}, \quad \beta_2 = \frac{a_1 + b_1}{\lambda_2 + 1}$$

Otherwise $\lambda_2 = \infty$ and thus,

$$\alpha_2 = a_1 + b_1, \quad \beta_2 = 0$$

For α, β real numbers and $x \in \mathcal{X}$

$$t(\alpha, \beta|x) = \begin{cases} \alpha & \text{if } x = 0 \\ \beta & \text{if } x = 1 \end{cases} \tag{3.18}$$

We define $\mathcal{Q}' : \mathcal{X} \to \mathcal{Z}'$ with the output alphabet

$$\mathcal{Z}' = \mathcal{Y} \setminus \{y_2, \tilde{y}_2, y_1, \tilde{y}_1\} \cup \{z_2, \tilde{z}_2\}$$

for all $x \in \mathcal{X}$ and $z \in \mathcal{Z}'$.

Then the upgraded $\mathcal{Q}'$ with respect to $\mathcal{W}$, $\mathcal{Q}' \succcurlyeq \mathcal{W}$, is given by

$$\mathcal{Q}'(z|x) = \begin{cases} \mathcal{W}(z|x) & \text{if } z \notin \{z_2, \tilde{z}_2\}, \\ \mathcal{W}(y_2|x) + t(\alpha_2, \beta_2|x) & \text{if } z = z_2, \\ \mathcal{W}(\tilde{y}_2|x) + t(\beta_2, \alpha_2|x) & \text{if } z = \tilde{z}_2. \end{cases} \tag{3.19}$$

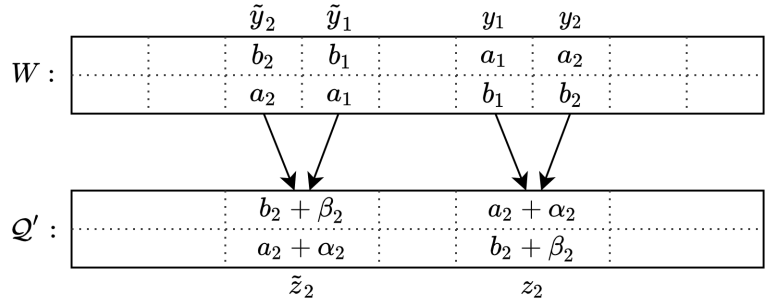


Figure 3.2: The upgrading merge operation **Method 1**

Method 2: As proved in [8], given a BSM channel $\mathcal{W}: \mathcal{X} \to \mathcal{Y}$ and let the symbols that we want to merge be $y_1, y_2, y_3 \in \mathcal{Y}$. We denote $\lambda_1 = LR(y_1)$, $\lambda_2 = LR(y_2)$ and $\lambda_3 = LR(y_3)$, assuming that

$$1 \leq \lambda_1 < \lambda_2 < \lambda_3.$$

Also, let $a_2 = \mathcal{W}(y_2|0)$ and $b_2 = \mathcal{W}(\tilde{y}_2|0)$. We define $\alpha_1, \beta_1, \alpha_3, \beta_3$ as follows.

If $\lambda_3 < \infty$

$$\alpha_1 = \lambda_1 \frac{\lambda_3 b_2 - a_2}{\lambda_3 - \lambda_1}, \quad \beta_1 = \frac{\lambda_3 b_2 - a_2}{\lambda_3 - \lambda_1}$$
$$\alpha_3 = \lambda_3 \frac{a_2 - \lambda_1 b_2}{\lambda_3 - \lambda_1}, \quad \beta_1 = \frac{a_2 - \lambda_1 b_2}{\lambda_3 - \lambda_1}$$

Otherwise $\lambda_3 = \infty$ and thus,

$$\alpha_1 = \lambda_1 b_2, \quad \beta_1 = b_2$$
$$\alpha_3 = a_2 - \lambda_1 b_2, \quad \beta_3 = 0$$

Let $t(\alpha, \beta|x)$ be the same as in **Method 1**.

We define $\mathcal{Q}': \mathcal{X} \to \mathcal{Z}'$ with the output alphabet

$$\mathcal{Z}' = \mathcal{Y} \setminus \{y_1, \tilde{y}_1, y_2, \tilde{y}_2, y_3, \tilde{y}_3\} \cup \{z_1, \tilde{z}_1, z_3, \tilde{z}_3\}$$

for all $x \in \mathcal{X}$ and $z \in \mathcal{Z}'$.

Then the upgraded $\mathcal{Q}'$ with respect to $\mathcal{W}$, $\mathcal{Q}' \succcurlyeq \mathcal{W}$, is given by

$$\mathcal{Q}'(z|x) = \begin{cases} \mathcal{W}(z|x) & \text{if } z \notin \{z_1, \tilde{z}_1, z_3, \tilde{z}_3\}, \\ \mathcal{W}(y_1|x) + t(\alpha_1, \beta_1|x) & \text{if } z = z_1, \\ \mathcal{W}(\tilde{y_1}|x) + t(\beta_1, \alpha_1|x) & \text{if } z = \tilde{z}_1. \\ \mathcal{W}(y_3|x) + t(\alpha_3, \beta_3|x) & \text{if } z = z_3, \\ \mathcal{W}(\tilde{y_3}|x) + t(\beta_3, \alpha_3|x) & \text{if } z = \tilde{z}_3. \end{cases} \tag{3.20}$$

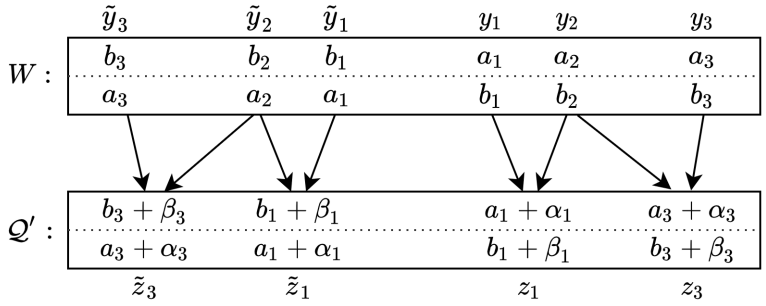


Figure 3.3: The upgrading merge operation **Method 2**

Method 2 has several advantages over Method 1, the only disadvantage of Method 2 is that if λ_1 and λ_3 are very close, the denominator of $\alpha_1, \beta_1, \alpha_3, \beta_3$ will cause numerical instabilities.

Thus to construct the upgrading_merge algorithm, we define a threshold ε and check if any neighboring λ_s are closer than this threshold, if so we employ **Method 1** otherwise we use **Method 2** to have a better merging operation. Having said that, and having the algorithm for the construction of the degrading_merge at hand, the algorithm for the upgrading_merge is quite straightforward. As in the case of the degrading merge, the output alphabet size of the upgraded channel $\mathcal{Q}'$ decreases by 2 after each iteration of the upgrading_merge. Similarly to the degrading_merge, the complexity of the upgrading_merge algorithm is $O(\mu^2 \log \mu)$.

3.3 Performance of the approximations on the BSC under SC decoding

To determine the performance of polar coding on the BSC under SC decoding, we define the transition probabilities for the BSC channel W and find the information set as we described earlier. For symmetric capacity I(W)=0.5 (p=0.11) and block lengths ranging from 2^5 to 2^9, we run 4000 experiments for rate R from 1 to 0 in order to obtain the BER in relation to the rate R.

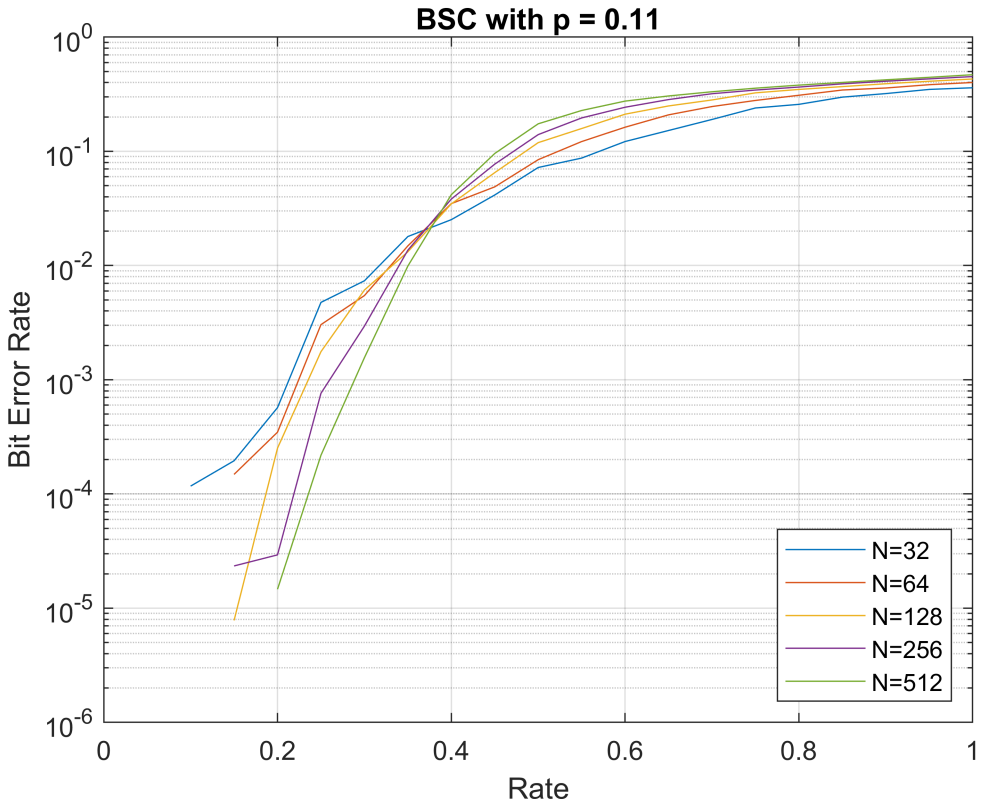


Figure 3.4: Performance of polar coding on the BSC with a probability of error p=0.11.

Figure 3.4 indicates, as expected, that as the block length increases, the BER decreases since polar codes approach capacity when the block length increases to infinity.

Chapter 4

Polarization of q-ary Discrete Memoryless Channels

The channel polarization method introduced by Arikan in [1] was initially proposed for binary-input Memoryless channels. Later on, [2], [6], [4], [5] set the groundwork for expanding channel polarization for q-ary input channels.

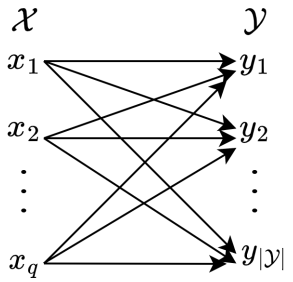


Figure 4.1: Q-ary Discrete Memoryless Channel

4.1 Channel Parameters

In order to polarize q-ary channels, we must first generalize the relations for the rate, reliability, and channel transformations for q-ary input channels.

4.1.1 Symmetric Capacity

According to [9], given a channel $W : \mathcal{X} \to \mathcal{Y}$ with input alphabet $\mathcal{X} = \{0, 1, .., q-1\}$, the symmetric capacity is defined as:

$$I(W) \triangleq \sum_{y \in \mathcal{Y}} \sum_{x \in \mathcal{X}} \frac{1}{q} W(y|x) \log_q \frac{W(y|x)}{\sum_{x' \in \mathcal{X}} \frac{1}{q} W(y|x')} \tag{4.1}$$

Since the base of the logarithm equals the alphabet size q, the symmetric capacity has values in

$$0 \leq I(W) \leq 1$$

.

4.1.2 Bhattacharrya parameter

The Bhattacharyya distance between a pair of input symbols x,x' is defined as follows

$$Z(W_{\{x,x'\}}) = \sum_{y \in \mathcal{Y}} \sqrt{W(y|x) W(y|x')} \tag{4.2}$$

According to [9], the average Bhattacharyya distance of W defines an upper bound on the probability of error of uncoded transmission

$$Z(W) = \sum_{x,x' \in \mathcal{X}, x \neq x'} \frac{1}{q(q-1)} Z(W_{\{x,x'\}}) \tag{4.3}$$

The relation between the error probability and the Bhattacharyya parameter is defined as

$$Pe \leq (q-1) Z(W) \tag{4.4}$$

Finally, the relation between the symmetric capacity I(W) and the Bhattacharyya parameter Z(W) is given by

$$I(W) \geq \log \frac{q}{1 + (q-1) Z(W)} \tag{4.5}$$

$$I(W) \leq \log(\frac{q}{2}) + (\log 2) \sqrt{1 - Z(W)^2} \tag{4.6}$$

$$I(W) \leq 2(q-1)(\log e) \sqrt{1 - Z(W)^2} \tag{4.7}$$

4.2 Channel Transformations

The channel transformations for q-ary input channels are similar to those for the polarization of binary-input channels, with the exception that in the case of the q-ary channels the operations are over GF(q). Given a channel $W : \mathcal{X} \to \mathcal{Y}$ with input alphabet $\mathcal{X} = \{0, 1, .., q-1\}$, where q is a prime number, it is proven in [9] that the single step transformations are defined as follows:

$$W^{-}(y_1, y_2|u_1) = \sum_{u_2 \in \mathcal{X}} \frac{1}{q} W(y_1|u_1 + u_2) W(y_2|u_2) \tag{4.8}$$

$$W^{+}(y_1, y_2, u_1|u_2) = \frac{1}{q} W(y_1|u_1 + u_2) W(y_2|u_2) \tag{4.9}$$

Also, it is known that

$$W_2(y_1, y_2|u_1, u_2) = W(y_1|u_1 + u_2) W(y_2|u_2) \tag{4.10}$$

4.3 Ternary Symmetric Channel

The Ternary Symmetric Channel depicted in figure 4.3 with input alphabet $\mathcal{X} = \{0, 1, 2\}$ and output alphabet $\mathcal{Y} = \{0, 1, 2\}$ has the following symmetric capacity

$$I(W) = 1 + p \log_3(p/2) + (1-p) \log_3(1-p) \tag{4.11}$$

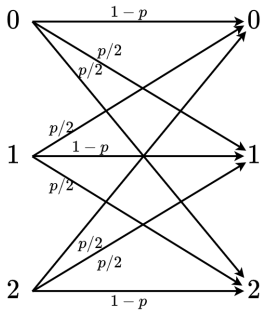


Figure 4.2: Ternary symmetric channel (TSC)

We can see the effect of the basic polarization step on the symmetric capacity of the Ternary Symmetric channel bellow

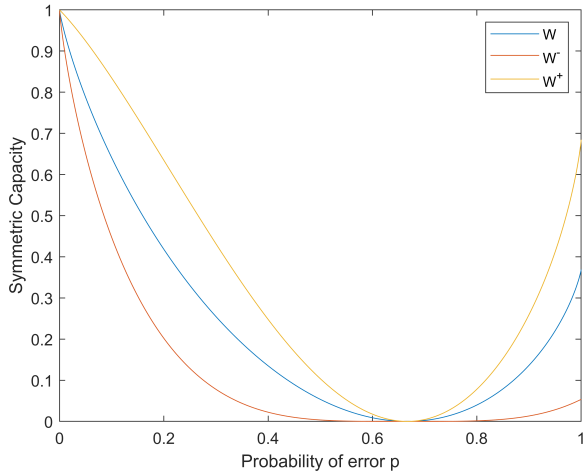


Figure 4.3: Symmetric Capacity vs Error probability of the original and the polarized TSC

As expected, the symmetric capacity of the degraded channel W^- is smaller than the original channel before polarization, and the symmetric capacity of the upgraded channel W^+ is greater than the original channel W.

4.4 Encoding

The construction of the encoder when q is a prime number, is similar to that for binary-input channels, using Arikan's transform $F = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$ and the generator matrix G_N over GF(q).

$$G_N = B_N F^{\otimes n} \text{ over GF(q).} \tag{4.12}$$

where B_N is the bit reversal permutation matrix $B_N = R_N(I_2 \otimes B_{N/2})$ and $F^{\otimes n}$ is the n kronecker product of the matrix $F = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$.

Hence, the encoding operation is given by:

$$x_1^N = (u_1^N G_N) \mod 3.$$

4.5 Decoding

Similar to the Binary case, we use a successive cancellation decoder, which is based on the following recursive formulas:

$$W_{2N}^{(2i-1)}(y_1^{2N}, u_1^{2i-2}|u_{2i-1}) = \sum_{u_{2i}} \frac{1}{q} W_N^{(i)}(y_1^N, u_{1,o}^{2i-2} \oplus u_{1,e}^{2i-2}|u_{2i-1} \oplus u_{2i}) \cdot W_N^{(i)}(y_{N+1}^{2N}, u_{1,e}^{2i-2}|u_{2i}) \tag{4.13}$$

$$W_{2N}^{(2i)}(y_1^{2N}, u_1^{2i-1}|u_{2i}) = \frac{1}{q} W_N^{(i)}(y_1^N, u_{1,o}^{2i-2} \oplus u_{1,e}^{2i-2}|u_{2i-1} \oplus u_{2i}) \cdot W_N^{(i)}(y_{N+1}^{2N}, u_{1,e}^{2i-2}|u_{2i}) \tag{4.14}$$

and the decision is made according to the following decision rule

$$\hat{u}_i = \begin{cases} u_i & \text{if } u_i \in A^c \\ argmax_{x \in (0,1,\ldots,q-1)} W_N^{(i)}(y_1^N, u_1^{i-1}|x) & \text{otherwise} \end{cases} \tag{4.15}$$

4.6 Channel Approximations

The construction of the degraded subchannels is accomplished similarly to the case of the BMS channel. We recursively apply the channel transformations based on the channel index and then implement the degrading merge function in order to reduce the output alphabet size to at most μ. The algorithm proposed in [2] for the degrading procedure is the following

and it is called, greedy mass merging. In its general form, it is very inefficient for non-binary input alphabets since we have to check every pair of symbols, thus the complexity is at most $O(N\mu^4 \log \mu)$. To minimize the complexity and make it feasible to implement, we use a technique called "No-loss alphabet reduction".

Algorithm 3: q-ary Channel Degrading

Input: DMC channel W, a bound μ on the output alphabet size, code length $N = 2^n$, and an index i with binary representation $i = \langle b_1, b_2, ..., b_n \rangle_2$.

Output: A DMC obtained from the subchannel $W_N^{(i)}$.

$T_N^{(i)} \leftarrow$ Degrading_merge_qary(W, μ)

for *j = 1,2,...,n* **do**

  if $b_j = 0$ **then**

    $T_N^{(i)} \leftarrow W^-$

  else

    $T_N^{(i)} \leftarrow W^+$

  $T_N^{(i)} \leftarrow$ Degrading_merge_qary(T, μ)

return $T_N^{(i)}$

4.6.1 No-loss Alphabet Reduction

Given a DMC $W : X \to Y$, we calculate the joint PMF P_{XY} on $X \times Y$ by assuming prior on X using the following relations:

$$P_W(x|y) = \frac{W(y|x)}{\sum_{x_0 \in X} W(y|x_0)} \tag{4.16}$$

$$P_Y(y) = \frac{1}{q} \sum_{x_0 \in X} W(y|x_0) \tag{4.17}$$

for every $x \in X$ and $y \in Y$.

It is also known that for q-ary polar codes with $q \geq 2$, the symmetric capacity $I(W) = \log q - H(X|Y)$ and the conditional entropy $H(X|Y) = E(-\log P_{X|Y}(X|Y))$. Thus we define the channel transformations in terms of the reverse channel $P_{X|Y}$ over GF(q).

$$\begin{aligned} P^+_{Y^+}(u, y_i, y_j) &= P_Y(y_i) P_Y(y_j) \sum_{x \in \mathcal{X}} P_{X|Y}(u + x|y_i) P_{X|Y}(x|y_j) \\ P^+_{X|Y^+}(x|u, y_i, y_j) &= \frac{P_{X|Y}(u + x|y_i) P_{X|Y}(x|y_j)}{\sum_{x_0 \in \mathcal{X}} P_{X|Y}(u + x_0|y_i) P_{X|Y}(x_0|y_j)} \\ P^+_{X^+}(x) &= \sum_{u \in \mathcal{X}, y_i, y_j \in \mathcal{Y}} P^+_{X|Y^+}(x|u, y_i, y_j) P^+_{Y^+}(u, y_i, y_j) \end{aligned} \tag{4.18}$$

and,

$$
\begin{aligned}
P^{-}_{Y^-}(y_i, y_j) &= P_Y(y_i) P_Y(y_j) \\
P^{-}_{X|Y^-}(x|y_i, y_j) &= \sum_{u_2 \in \mathcal{X}} P_{X|Y}(x + u_2|y_i) \times P_{X|Y}(u_2|y_j) \\
P^{-}_{X^-}(x) &= \sum_{y_i, y_j \in \mathcal{Y}} P^{-}_{X|Y^-}(x|y_i, y_j) P^{-}_{Y^-}(y_i, y_j)
\end{aligned}
\tag{4.19}
$$

On the above transformations, the operator + is modulo q addition. If P_X is uniform, P^+_X and P^-_X are also uniform. Thus, $P^-_{X|Y^-}$ and $P^+_{X|Y^+}$ are equal to the posterior distributions induced by Arikan's transformations. The transformations W^- and W^+ and the q-ary Channel Degrading function will be replaced by the above-described transformations of probability distribution.

4.6.2 Cyclic Unification

According to [2], given a distribution P_{XY} on $\mathcal{X} \times \mathcal{Y}$, we define the equivalence relation on $\mathcal{Y}$, $y_1 \sim y_2$, such that if $x_1 \in \mathcal{X}$ then $P_{X|Y}(x + x_1|y_1) = P_{X|Y}(x|y_2)$ for every $x \in \mathcal{X}$. Thus, if $y_1 \sim y_2$ we can merge y_1 and y_2 without rate loss.

Proposition As it is mentioned in [2], we can prove the above claim accordingly: Given a distribution P_{XY} on $\mathcal{X} \times \mathcal{Y}$. For every $y_1, y_2 \in \mathcal{Y}$ and any $u_1, u \in \mathcal{X}$, it holds that

$$
\begin{aligned}
P^+_{X|Y^+}(u|(u_1, y_1, y_2)) =& \frac{P_{X|Y}(u_1 + u|y_1) P_{X|Y}(u|y_2)}{\sum_{x_0 \in \mathcal{X}} P_{X|Y}(u_1 + x_0|y_1) P_{X|Y}(x_0|y_2)} \\
=& \frac{P(-u_1 + (u + u_1)|y_2) P(u_1 + u|y_1)}{\sum_{x_0 \in \mathcal{X}} P_{X|Y}(-u_1 + (u_1 + x_0)|y_2) P_{X|Y}(u_1 + x_0|y_1)} \\
=& P^+_{X|Y^+}(u + u_1|(-u_1, y_2, y_1))
\end{aligned}
\tag{4.20}
$$

Thus, for every $(u, y_1, y_2) \in \mathcal{X} \times \mathcal{Y}^2$ it is true that

$$
(u, y_1, y_2) \overset{P^+}{\sim} (-u, y_2, y_1)
$$

where if $y_1 = y_2$ then $u \neq 0$.

4.6.3 Proposed Greedy Mass Merging Algorithm

Given a DMC $W : \mathcal{X} \to \mathcal{Y}$, we recursively calculate $P^s_{XY^s}$ and after each step, we proceed as follows:

- If the last step in s is +, then we first use the function *merge_pair* to merge the symbols (u_1, y_1, y_2) and $(-u_1, y_1, y_2)$ for all u_1, y_1, y_2 and then we use the function degrade on $P^s_{XY^s}$
- If the last step in s is -, then we use the function Degrading_merge_qary on $P^s_{XY^s}$.

The function **Degrading_merge_qary** is defined as follows:

Algorithm 4: Degrading_merge_qary

Input: distribution P_{X,Y_0} over $\mathcal{X} \times \mathcal{Y}_0$, bound μ on the output alphabet size.

Output: distribution $\mathcal{Q}_{X,Y}$ over $\mathcal{X} \times \mathcal{Y}$, where $|\mathcal{Y}| \leq \mu$

$\mathcal{Q} \leftarrow P$

$l \leftarrow |\mathcal{Y}|$

while $l > \mu$ **do**

  $(y_1, y_2, u) \leftarrow \text{choose}(\mathcal{Q})$

  $\mathcal{Q} \leftarrow \text{merge\_pair}(\mathcal{Q}(y_1, y_2, u))$

  $l \leftarrow l - 1$

return $\mathcal{Q}$

The function $choose(Q)$ finds y_1, y_2, u that when merged with $merge\_pair(Q, (y_1, y_2, u))$ the change of conditional entropy $H_Q(X|Y)$

$$\Delta(H) \triangleq Q_{\tilde{Y}}(\tilde{y}) H(X|\tilde{Y} = \tilde{y}) - \sum_{i=1}^{2} Q_Y(y_i) H(X|Y = y_i)$$

is the smallest among all $(y_i, y_j, u) \in \mathcal{Y}^2 \times \mathcal{X}$.

The function $merge\_pair(Q, (y_1, y_2, u))$ merges two symbols into one as follows:

$$\tilde{y} = y \backslash \{y_1, y_2\} \cup \{\tilde{y}\}$$

where $Q_{\tilde{Y}}(y) = Q_Y(y), Q_{X|\tilde{Y}}(x|y) = Q_{X|Y}(x|y)$ for all $x \in \mathcal{X}$ and $y \in \tilde{y} \backslash \{\tilde{y}\}$ and

$$Q_{\tilde{Y}}(y) = Q_Y(y_1) + Q_Y(y_2)$$

$$Q_{X|\tilde{Y}}(x|\tilde{y}) = \frac{Q_Y(y_1) Q_{X|Y}(x|y_1) + Q_Y(y_2) Q_{X|Y}(x + u|y_2)}{Q_{\tilde{Y}}(y)}$$

4.6.4 Cyclic Unification Algorithm

After we calculate the joint P_{XY} on $\mathcal{X} \times \mathcal{Y}$, we then recursively calculate $P^s_{XY^s}$ and reduce the output alphabet size by assigning one symbol to the whole class A such that:

$$P^s_{Y^s}(A) = \sum_{y \in A} P^s_{Y^s}(y) \tag{4.21}$$

$$P^s_{X|Y^s}(x|A) = P^s_{X|Y^s}(x|y^*) \tag{4.22}$$

Where y^* is an arbitrarily chosen $y \in A$ because $P^s_{X|Y^s}(\cdot|y), y \in A$ are cyclically shifted.

4.6.5 Modifications to the proposed Algorithm

Firstly, it is quite obvious that in function choose, the exhaustive search on the smallest change of conditional entropy between all the triples $(y_i, y_j, u) \in \mathcal{Y}^2 \times \mathcal{X}$, is very inefficient in practice. To overcome this problem, we set a threshold ε and select the first triple (y_i, y_j, u) that has a rate loss smaller than the threshold.

In addition, there is an ambiguity between the use of the proposition of cyclic unification and the cyclic unification algorithm when it comes to merging cyclically shifted vectors. While the greedy mass merging algorithm proposes that the cyclic unification is accomplished with the use of the proposition, we saw that using the cyclic unification algorithm instead of the proposition, is faster and has a smaller output alphabet size, since we can merge not only a pair of symbols but large sets of cyclically shifted vectors each time.

Algorithm 5: New_degrading_merge_qary

Input: distribution $P_{X,Y}$ over $\mathcal{X} \times \mathcal{Y}$

Output: distribution $\mathcal{Q}_{X,Y}$ over $\mathcal{X} \times \mathcal{Y}$

$u \leftarrow$ unique vectors in $P_{X,Y}$

$s \leftarrow |u|$

for *i=1...s* **do**

  $\mathcal{Q}_Y(i) \leftarrow \sum_{y \in A} P_Y(y)$

  $\mathcal{Q}_{Y|Y}(i) \leftarrow P_{X|Y}(x|y^*)$

return $\mathcal{Q}$

4.7 TSC simulation

To determine the performance of polar coding on the TSC under SC decoding, For symmetric capacity I(W)=0.5 (p=0.16) and block lengths ranging from 2^5 to 2^8, we run 2000 experiments for rate R from 1 to 0 in order to obtain the SER in relation to the rate R and saw the improvement on the SER as the block length increases.

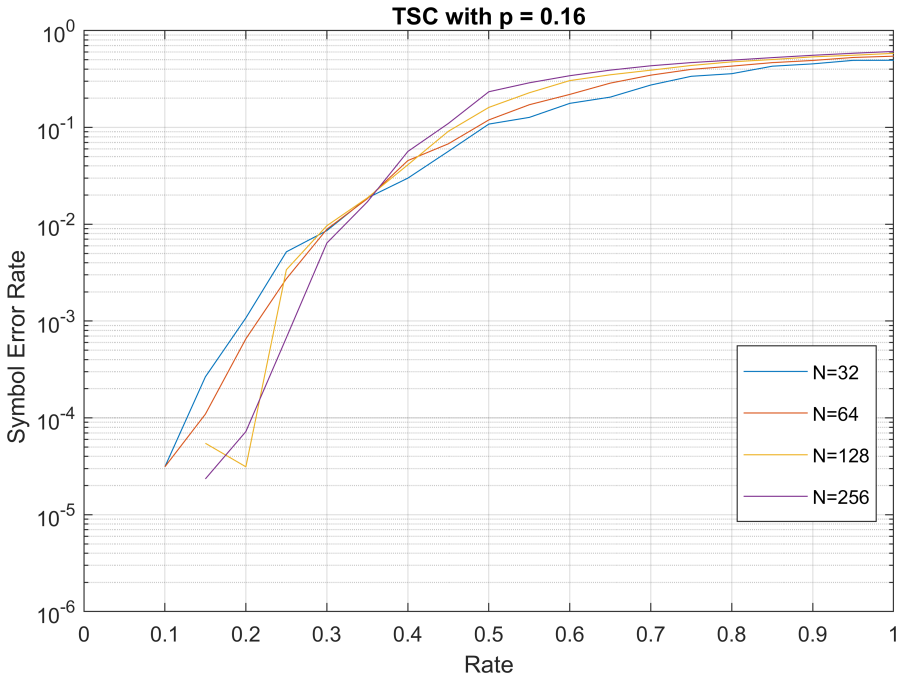


Figure 4.4: Performance of polar coding on the TSC with a probability of error p=0.16

Appendix

Proof of relation(3.9)

$$P_e(\mathcal{W}) = \frac{1}{2}\sum_{y\in\mathcal{Y}} min\{\mathcal{W}(y|0), \mathcal{W}(y|1)\} \tag{4.23}$$

Given a BMS channel with input alphabet $\mathcal{X} = 0, 1$ and arbitrary output, under a maximum-likelihood decision, the probability of error is

$$P(E) = \sum_{y\in\mathcal{Y}} P(Y = y)P(E|Y = y) \tag{4.24}$$

where

$$P(Y = y) = \frac{1}{2}W(Y = y|X = 0) + \frac{1}{2}W(Y = y|X = 1) \tag{4.25}$$

and

$$\begin{aligned} P(E|Y = y) =& \sum_{x\in\mathbf{X}} P(X = x|Y = y) + P(E|Y = y, X = x) \\ =& P(X = 0|Y = y) + P(E|Y = y, X = 0) \\ &+ P(X = 1|Y = y) + P(E|Y = y, X = 1) \end{aligned} \tag{4.26}$$

for any given $y \in Y$ one of the probabilities $P(X = 0|Y = y)$ and $P(X = 1|Y = y)$ will be equal to 0 and the other will be equal to 1

Also according to Bayes' rule

$$\begin{aligned} P(X = x|Y = y) =& \frac{W(Y = y|X = x)P(X = x)}{P(Y = y)} \\ =& \frac{\frac{1}{2}W(Y = y|X = x)}{P(Y = y)} \end{aligned} \tag{4.27}$$

Considering this

$$P(E|Y=y) = \frac{\frac{1}{2}W(Y=y|X=x)}{P(Y=y)} \tag{4.28}$$

Thus, substituting (4.25) and (4.28) on (4.24) we get that

$$P(E) = \sum_{y\in\mathcal{Y}} P(X=x)W(Y=y|X=x) = \sum_{y\in\mathcal{Y}} \frac{1}{2}W(Y=y|X=x) \tag{4.29}$$

and since we use ML decision rule

$$W(Y=y|X=x) = \mathbf{min}\{W(Y=y|X=0), W(Y=y|X=1)\} \tag{4.30}$$

$$P(E) = \frac{1}{2}\sum_{y\in\mathcal{Y}} min\{\mathcal{W}(y|0), \mathcal{W}(y|1)\} \tag{4.31}$$

References

[1] Erdal Arikan. Channel polarization: A method for constructing capacity-achieving codes. In *2008 IEEE International Symposium on Information Theory*, pages 1173–1177, 2008.

[2] Talha Cihad Gulcu, Min Ye, and Alexander Barg. Construction of polar codes for arbitrary discrete memoryless channels. *IEEE Transactions on Information Theory*, 64(1):309–321, 2018.

[3] Ryuhei Mori and Toshiyuki Tanaka. Performance and construction of polar codes on symmetric binary-input memoryless channels. In *2009 IEEE International Symposium on Information Theory*, pages 1496–1500, 2009.

[4] Ryuhei Mori and Toshiyuki Tanaka. Channel polarization on q-ary discrete memoryless channels by arbitrary kernels. In *2010 IEEE International Symposium on Information Theory*, pages 894–898, 2010.

[5] Rajai Nasser and Emre Telatar. Polar codes for arbitrary dmcs and arbitrary macs. *IEEE Transactions on Information Theory*, 62(6):2917–2936, 2016.

[6] Woomyoung Park and Alexander Barg. Polar codes for q-ary channels, $q = 2^r$. *IEEE Transactions on Information Theory*, 59(2):955–969, 2013.

[7] C. E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27(4):623–656, 1948.

[8] Ido Tal and Alexander Vardy. How to construct polar codes. *IEEE Transactions on Information Theory*, 59(10):6562–6582, 2013.

[9] Eren Şaşoğlu, Emre Telatar, and Erdal Arikan. Polarization for arbitrary discrete memoryless channels. In *2009 IEEE Information Theory Workshop*, pages 144–148, 2009.