Design of Polar Lattices Based on Subcode Rate Selection

By Erick Christian GARCIA ALVAREZ

A thesis submitted to
School of Information Science,
Japan Advanced Institute of Science and Technology,
in partial fulfillment of the requirements
for the degree of
Master of Information Science
Graduate Program in Information Science

Written under the direction of
Professor Brian Kurkoski

March, 2017

Design of Polar Lattices Based on Subcode Rate Selection

By Erick Christian GARCIA ALVAREZ (s1410202)

A thesis submitted to
School of Information Science,
Japan Advanced Institute of Science and Technology,
in partial fulfillment of the requirements
for the degree of
Master of Information Science
Graduate Program in Information Science

Written under the direction of
Professor Brian Kurkoski

and approved by
Professor Brian Kurkoski
Professor Tadashi Matsumoto
Professor Masahi Unoki

February, 2017 (Submitted)

I certify that I have prepared this Master's Thesis by myself without any inadmissible outside help.

Erick Christian GARCIA ALVAREZ
JAIST, February 10th 2017.

Author : _____

Date : _____

Supervisor : _____

Acknowledgments

This research could not have been done without the constant help of my supervisor Brian M. Kurkoski. He always gave me precise ideas to follow. I would like to thank him for his support in both personally and academically. As well, thank you professor Tadashi Matsumoto which showed me the dedication towards research. I am specially thankful to my senior Ryota Sekiya which helped me from the very beginning of my research studies. He was patient at any time, he helped me to solve many questions related to our research and he always looked happy to explain. I am very thankful to my seniors Ricardo A. Parrao Hernandez and Francisco J. Cuadros Romero, who pushed me towards producing elegant research, thanks to that constant help, I was able to obtain results on this research. Also, thank you to all the time spend on me, providing explanation to many topics that I did not understand clearly. And, thank you to all other members in Matsumoto lab and BITS lab, I enjoyed spending time with all of you. I am convince that having a working place with good environment provides easier and relax mind which helps to focus and also helps to produce great research.

Achievements

Research student

1 Erick C. Garcia Alvarez, Brian M. Kurkoski, Ryota Sekiya, "Write-Once Memory Codes for the AMAC with Efficient Threshold Decoding," *IEICE Technical Report*, Vol. 115, No. 214, pp. 13–18, September 2015.

2 Ryota Sekiya, Erick C. Garcia Alvarez, Brian M. Kurkoski, Hideki Yagi, "Applying Write Once Memory Codes to Asymmetric Multiple Access Channel with Errors," *SITA2014*, pp. 577–582, December 2014.

3 Erick C. Garcia Alvarez, Brian M. Kurkoski, Ryota Sekiya, "Performance of Write–Once Memory Codes Applied to Asymmetric Multiple Access Channel", *SITA2014*, pp. 632–637, December 2014.

4 Ryota Sekiya, Erick C. Garcia Alvarez, Brian M. Kurkoski, "Write-Once Memory Codes for Low-Complexity Decoding of Asymmetric Multiple Access Channel", *ISITA2014*, Melbourne, Australia, pp. 610–614, October 26-29, 2014.

Minor Research

5 Erick C. Garcia Alvarez, Shengbei Wang, Masashi Unoki, "An automatic watermarking in CELP speech codec based on formant tuning," *Proc. 2015 International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIHMSP2015)*, September 2015.

6 Erick C. Garcia Alvarez, Shengbei Wang, Masashi Unoki, "Analysis of Watermarking into CELP Speech Codec Based on Formant Tuning," *IEICE-115 Technical Report*, September 2015.

National Symposium

7 Erick C. Garcia Alvarez, Brian M. Kurkoski, "On the design of Polar Lattices: Selection of Subcode Rates," *SITA2016*, pp. 390–395, December 2016.

Poster Presentation

8 Erick C. Garcia Alvarez, Brian M. Kurkoski, "Performance comparison of multilevel Polar Lattices," *ISITA2016*, Monterey, California, USA, pp. 42, November 2^{nd}, 2016.

Abstract

This thesis proposes a design of lattices based on subcodes. In literature it is known how to construct polar lattices, however construction of optimal lattices are still an open problem. This work aims to form lattices from polar codes and study their performance.

To obtain efficient polar lattices in terms of low error rate, we selected various polar subcodes to compare which produces lower bit-error rate.

In this work, lattices are formed using construction A and construction D. Both constructions require a binary linear code $\mathcal{C}$. A lattice constructed with a single code is called *single-level lattice*. A lattice constructed with two nested codes is called *two-level lattice*. Multilevel refers to the number of nested codes chosen to form a lattice.

Construction A produces lattices which, in general, are effective in lower dimensions. Lattices Λ are obtained by using a single binary code $\mathcal{C}$. Each codeword of the binary code is assumed to be a lattice point in the real space $\mathbb{R}^N$.

Construction D is one of the various lattice constructions which produces lattices from nested binary codes. In this work, lattices Λ are formed by selecting nested polar codes $\mathcal{C}_i$.

We selected polar codes as the binary code $\mathcal{C}$ denoted by $Polar(N, K, \mathcal{F})$. Polar codes are specified by the channel transition probabilities $W(y|x)$, N is the block code length (or the lattice dimension), K is the number of information bits and the index vector $\mathcal{F}$ that has $N - K$ elements, which in literature is commonly called frozen bits.

We are interested in the generator matrix $\mathbf{G}_N$ of the polar code, because sub polar codes can be obtained by using $\mathbf{G}_N$. The full rank generator matrix $\mathbf{G}_N$ is defined by: $\mathbf{G}_N = \mathbf{R}_N(\mathbf{F} \otimes \mathbf{I}_{\frac{N}{2}}) \cdot (\mathbf{I}_2 \otimes \mathbf{G}_{\frac{N}{2}})$, where $\mathbf{R}_N$ is the reverse shuffle permutation matrix, $\mathbf{F} = \left[\begin{smallmatrix} 1 & 0 \\ 1 & 1 \end{smallmatrix}\right]$ and $\otimes$ is the Kronecker product. A given polar subcode contains some of the basis row vectors as the generator matrix $\mathbf{G}_N$.

Polar codes were chosen because polar codes have a structured construction providing easier identification of subcodes required to build up polar lattice by construction D.

Polar lattices Λ_P are specified by $\Lambda_P(N, K_i)$, where N is the dimension of the lattice and K is the dimension of the nested binary code i. In a few lines, the polar lattice

construction is explained as follows. From the generator matrix $\mathbf{G}_N$, subcodes with rate $(\frac{K_i}{N})$ are chosen, following the polar code construction, such as the selection of frozen bits by maximizing the symmetric capacity, which is explain in detail in section 3. To form polar lattices we use the polar subcodes and apply: (1) construction A requires a single polar code. Details are explained in section 4.4.1. (2) construction D requires nested polar codes. There is a restriction on the minimum distance $d_i \geq \frac{4^{(i)}}{j}$ for the nested polar codes, where j is either 1 or 2, and in this work we used 2. i is the number of nested polar codes.

To evaluate the best bit-error rate, symbol-error rate and word-error rate performance for each lattice, Monte Carlo simulation were performed for:

1. Single-level lattice construction. Polar lattices are formed by a single code with construction A and construction D. We investigated the best BER, SER and WER performance for a give polar lattice under lower volume to noise ratio (VNR). VNR is the metric for lattices which shows how dense a lattice is. We evaluate from 0dB to 5dB for the VNR.

2. Multilevel lattice construction. Polar lattices are formed by nested codes using construction D. The objective of this simulation is to identify with how many polar lattices levels the best BER, SER and WER under lower VNR (from 0dB to 5dB VNR).

For such simulations, the additive white Gaussian noise (AWGN) channel is considered. And at the receiver side, the decoding is done using a multilevel decoding and successive cancellation decoding (SCD).

Simulation results show that single-level polar lattices outperform the two-level and three-level polar lattices. Simulations also show that polar lattices with lower minimum distance has lower error rate performance, this positive result occurs because the best performance occurs when a higher amount of lattice points are packed on the same volume.

In this work we also compared the performance of polar lattices with another code. We chose BCH codes as another binary code to form lattices. BCH codes are powerful random error-correcting cyclic codes.

Simulation where performed with single-level BCH lattices by construction A. Results show that at a lower VNR (1dB), BCH lattice performed worse than polar lattices in terms of SER. However, on a little higher VNR (4dB), the best BCH lattice seems improve the performance, but still lower than some polar lattices.

Table of Contents

List of Figures

List of Tables

List of Symbols

$\mathbb{R}^N$	Euclidean space, set of real numbers, N-dimensional real domain.
N	N-dimension
K	Information vector length
$\mathcal{F}$	Set of frozen bits
$\mathcal{F}^c$	Set of information bits
Λ	Lattice
$\mathbf{u} = U^N$	Source vector message (Polar codes system)
u	Single source message bit (Polar codes system)
$\mathbf{b}$	Source vector message (Lattice system)
b	Single source message bit (Lattice system)
$\mathbf{x} = X^N$	Codeword, lattice point
x	Single codeword bit
$\mathbf{y} = Y^N$	Received noisy vector
y	Single noisy bit
$\hat{\mathbf{b}}$	Estimated information
$\hat{\mathbf{x}}$	Estimated lattice point, estimated codeword
$\mathbf{z}$	Additive white Gaussian noise vector
e	Cross over probability
$Q(\cdot)$	Error probability function
E_b	Bit energy
N_0	Double-side power spectral density

$W(\cdot)$	Transition probability
$W_N(\cdot)$	Transition probability vector form
W	Virtual channel
W_N	Channel combined
π	Permutation index
$I(\cdot)$	Symmetric capacity, mutual information
$Z(\cdot)$	Bhattacharyya parameter
$\mathbf{z}_N$	Bhattacharyya index vector specific length N
$\mathbf{G}^P$	Polar code generator matrix
$\mathbf{G}_N^P$	Full rank polar code generator matrix
$\mathbf{G}^\Lambda$	Lattice generator matrix
$\mathbf{G}^{\Lambda_P}$	Polar lattice generator matrix
$\mathbf{B}$	Bit-reversal permutation matrix
$\mathbf{B}_N$	Bit-reversal permutation matrix with specific length N
$\mathbf{F}$	Plotkin matrix
$\otimes$	Kronecker product
R_N	Reverse shuffle permutation matrix of length N
$\mathbf{I}$	Identity matrix
$\mathcal{O}$	Complexity
n	$\log_2(N)$
$\log$	Logarithm
L	LLR
$\mathbf{g}$	Row basis vector
$V(\Lambda)$	Voronoi region
σ^2	Variance
$\mod$	Modulo operation
M	Number of codewords
$\mathcal{C}$	Binary code
C_{ch}	Capacity of the channel

d_{min}	Minimum Hamming distance of the binary code
d_i	Minimum distance restriction of construction D
q	$q = 2(a \mod 2)$
dB	Decibel

Acronyms

AWGN	Additive White Gaussian Noise
i.i.d.	independent and identically distributed
LLR	Log Likelihood Ratio
SNR	Signal-to-Noise Ratio
DMC	Discrete Memoryless Channel
BER	Bit Error Rate
SER	Symbol Error Rate
WER	Word Error Rate
LDPC	Low-Density Parity-Check
PAM	Pulse-Amplitude Modulation
QAM	Quadrature-Amplitude Modulation
VNR	Volume to Noise Ratio
NVNR	Normalized Volume to Noise Ratio
SCD	Successive Cancellation Decoder
PSCD	Permuted Successive Cancellation Decoder

Chapter 1

Introduction

1.1 Background, Motivation and Method

For reliable data transmissions there are still several problems to be solved. This thesis focuses on wireless communication models, and specifically on the design of codes to reliably transmit information through a noisy medium. The communication medium on which data is transmitted is called a channel, and when distortion occurs on the channel, it is called a noisy channel.

In 1948 C. E. Shannon developed a theorem for communication transmissions, to transmit efficiently and reliably, that is called the the noisy channel coding theorem. In his work, Shannon presented an analysis of coding theory for noisy communication channels where he obtained an upper bound on the probability of error for an optimal code on the discrete memoryless channel (DMC). The upper bound was obtained by letting the information being as random as possible. Another important contribution that Shannon presented was an expression for the capacity of a finite state channel. He introduced the limits of reliable communications, however, he did not mention how to achieve it practically.

In 1993 Claude Berrou *et al.* introduced turbo codes [1] which are recursive systematic convolutional codes that have lower bit error rates than the best non systematic convolutional codes at any signal-to-noise ratio for high code rates. Moreover, the performance of turbo codes in terms of bit-error rate is very close to Shannon's limit.

Another efficient binary code is the so-called low-density parity-check code (LDPC). LDPC codes are error correcting codes that have been proven to be capable of approaching the Shannon's limit more closely than any other code. However, LDPC are

not optimal codes, but LDPC codes have a simple decoding scheme.

Turbo codes and LDPC codes are efficient binary coding schemes when the message is modulated in the binary domain $\{0, 1\}$. However, regarding the modulation of a real communication channel, the noise and the transmission medium are functions of the real domain $\mathbb{R}^N$ and not binary. This is one of the main motivations to use lattices. A lattice is a periodic arrangement of point in the N-dimensional real space. Using lattice codes, an original message can be modulated in various dimensions in the real space $\mathbb{R}^N$. For example pulse-amplitude modulation (PAM) or quadrature-amplitude modulation (QAM), which correspond to one-dimensional and two-dimensional lattice-like constellation, respectively. An approach to construct multi-dimensional lattice constellations is by shaping (cutting a finite section from a lattice). Cubes and spherical shapes are the most widely used.

Lattices are capable of solving geometric, coding or quantization problems with a sphere-like codebook. Spherical-like codebook are constructed by taking a sphere (sphere packing problem) and placing a codeword from a binary code in each center. The sphere packing problem is to find the densest packing of spheres in a Euclidean space $\mathbb{R}^N$.

To construct dense sphere packing Λ, error correcting codes can be used. In [2], Conway and Sloane explained how to construct spherical codes using binary error correcting codes for lower or higher dimension.

It is possible to perform packing in space by two methods. (1) Using a single error correcting code, to produce packing in lower dimension; in [2] it is explained as "taking cross-sections of the binary code". And (2) building up packing by layers, producing packing in higher dimensions; in [2] it is explained as "laminated lattices".

Several techniques are known to construct lattices, such techniques are construction A, construction B, construction C, Construction D, etc. In this work, we focus on construction A and construction D using polar codes to evaluate its performance and then to compare both constructions.

In order to form lattices by construction A, a cross-section packing of the error correcting codes is performed. And, in order to use construction D, lattices are formed by building up layer by layer, where a set of nested binary codes is required.

A very powerful code which has an easy identification of its nested codes is the well-known Polar code. In [3], Arikan showed how to create a polar generator matrix by removing some rows of the Plotkin generator matrix. With this method, the polar

code generator matrix has a triangular form, which provides an easy construction of lattices. Lattices formed by polar codes are called polar latices which are defined in 5 on detail.

Polar codes are known to have an explicit construction with low encoding and decoding complexity [4]. Moreover, polar codes have a recursive structure that is suitable for hardware implementation [5]. Research on polar codes has demonstrated that their decoding with successive cancellation and short block lengths do not perform as efficiently as low density parity check codes (LDPC). In [6], Ido Tal and Alexander Vardy showed two possible causes of this performance variation. The first possible cause might be because of the existence of a performance gap between successive cancellation and maximum likelihood decoding. Or another possible cause might be simple because polar codes themselves are weak at short block lengths.

It is well known that using long block length codes, it is possible to increase the performance of polar codes. And also the performance of lattice codes. Combining them seems to be an appealing idea regarding to efficiency, however, we are aware that increasing the lattice dimension and so the length of the polar code leads to increase the decoding complexity of the system.

1.2 Summary of Contributions

This thesis aims to provide a guideline for beginner researchers who are interested in learning polar codes, lattices and polar lattices. This research provides a guideline to design lattices from polar codes by choosing subcode rates of the polar code. Polar lattices are formed by construction A and construction D with specific subcode rates. Evaluation of these construction are provided in section 6. Additionally, numerical results in terms of symbol error rate (SER), bit error rate (BER) and word error rate (WER) are provided in section 6.

The achievements of this dissertation can be summarized as follows:

- Polar lattices are formed by two different methods: construction A for single codes, and construction D for nested codes.
- It was found by computational simulation that polar lattices perform better on the BER than polar codes itself.
- For construction A, it was obtained that the best SER performance for polar

lattices is close a rate of ($\frac{K}{N} = \frac{1}{2}$) for the polar code design-rate.

- For construction D, it was observed that increasing the minimum distance between the nested polar codes decrease the SER performance of the polar lattice.
- Single-level lattices provides better SER performance than multilevel lattices.
- Polar lattices outperform BCH-Lattices on the lower VNR.

Chapter 2

Preliminaries

2.1 Source Coding and Channel Coding

The process of data transmission over a noisy channel is described by source coding and channel coding. A sender, receiver and medium are the three basic elements in this process. The sender is a source that produces encoded information vectors. The receiver decodes the received sequences to obtain the same information vector that was transmitted over the medium.

In source coding, the source generates symbols that are either discrete or for the idealistic case real symbols. Let consider a source which produces only two possible strings as outputs "No" and "Yes". These two strings can be mapped into 0's for No and 1's for Yes. Assuming noiseless reception of a transmitted bit, the receiver can easily reconstruct the original message given that the receiver knows $\{No \rightarrow 0, Yes \rightarrow 1\}$. In this easy example, the information strings are mapped to binary bits to represent the message. Accordingly, a relevant question would be, how to map efficiently?

Shannon asked himself this question already. *"Given a noiseless channel, what is the most efficient way to communicate the source message to the receiver?"*. Source codes try to answer this question mathematically. The fundamental idea is that highly probable symbols are assigned to the shortest coded transmissions. This ideal leads to convey maximum and by sending shorter coded message reduce the information being transmitted. Through reducing information (compression) and conveying maximum, this approach tries to reach the goal of source coding.

On the other hand, we know that communication channels are noisy. And we also know that noise can be modeled by using conditional probability distribution. Where

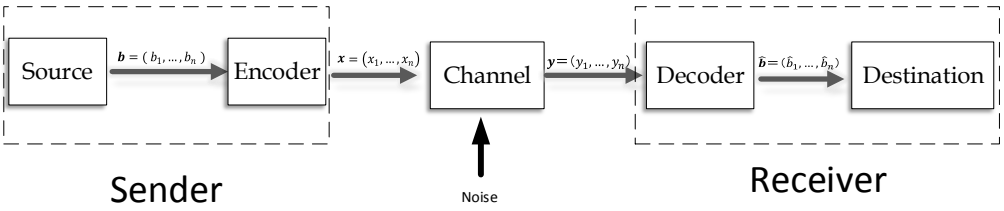

Figure 2.1. Basic communication system model.

the conditional probability distribution can be seen as a set of chances of getting particular output given a particular input with a certain random noise that is typically known.

Given the fact that communication channels are noisy, Shannon also asked the question. *"What is the maximum rate at which we can convey information with small chance of getting an error through this probabilistic channel?"*. The idea is to map the source coded information symbols through the noisy channel such that communication happens at the maximum possible rate as well as low error probabilities. This idea is called channel coding.

2.2 System Model

The general block diagram of a communication system is shown in Figure 2.1. The source message **b** is encoded by adding some redundancy to the original message to obtain codewords **x**. The sender transmits the codeword **x** through a noisy channel where error could occur due to noise in the channel. Generally the noise is modeled for example with a Gaussian distribution. The noisy vector **y** is received so that the decoder can make use of the redundancy on the message to detect and correct errors. The estimate vector $\hat{\mathbf{b}}$ is then sent to the destination.

2.3 AWGN Channel

The noise in a physical wireless channels is commonly modeled as additive white Gaussian noise (AWGN).

$$\mathbf{y} = \mathbf{x} + \mathbf{z} \tag{2.1}$$

The Gaussian random variables in $\mathbf{z}$ are i.i.d. with zero-mean, with a variance of $\sigma^2 = \frac{N_0}{2}$.

The cross over probability e can be calculated as follows:

$$e = Q\left(\sqrt{\frac{2E_b}{N_0}}\right) \tag{2.2}$$

where $Q(\cdot)$ is the error probability function, E_b is the bit energy and $N_0 = 2\sigma^2$ is called the double-side power spectral density. The ratio of the transmitted energy and the power spectral density of noise $\frac{E_b}{N_0}$ is called the signal-to-noise ratio (SNR).

2.4 Performance Measurements

Typically on digital communications systems the end-to-end performance measurement is the bit-error rate (BER), which quantifies the reliability of the entire communication system from how many "bits are in" to "bits are out," of the system.

The BER is defined by:

$$BER = \frac{\text{Errors}}{\text{Total number of bits}} \tag{2.3}$$

Another measurements for the communication systems are the symbol-error rate (SER) which is defined as the number of symbol changes for waveform changes, or signaling events, across the transmission medium. In this thesis we defined as:

$$SER = \frac{\text{Errors}}{\text{Total number of symbols}} \tag{2.4}$$

And, when measuring the entire codeword that is transmitted the word-error rate (WER) is used. We defined the WER as:

$$WER = \text{any error on the codeword} \tag{2.5}$$

These computational performance measurements require to send a data sequence through the system and compare the output with the input execution over an infinitely long period of time, and further more, it is essential to assume that a data transmission

is a random process. However, it is impossible to wait forever to make a BER, SER or WER measurement. Instead of that a "pseudorandom" data process is used. This process is called pseudorandom since it is not possible to truly randomize it, however it is possible to use deterministic methods.

Chapter 3

Polar Codes

Polar codes were introduced by Arikan [7] in 2009. Recently, polar codes have gained great interest due to its structured construction and its capability to achieve the capacity of symmetric binary-input of discrete memoryless channels. In [8] Huijun Li and Jinhong Yuan introduced a method to calculate the Bhattacharyya parameter and a construction of polar codes based on a Gaussian approximation. They evaluated the code performance showing that the code using a Gaussian approximation have better code performance with lower computational complexity than the code introduced by Arikan.

Several researchers have great interest in using polar codes, not only in wireless communications but also in other communication fields. From the field of wireless physical layer security, in [9] Andersson *et al.* obtained the capacity of wiretap channels for an specific setting. They used subcodes of the polar code to produce nested codes. Their coding scheme achieved the capacity of the physically degraded relay channel.

In this work, we use polar codes to form lattices. Firstly, we use the Bhattacharyya parameter to obtain a polar code and several subcodes. And secondly, we use the subcodes to shape lattices.

3.1 Polar Codes Definition

A polar code is specified by $Polar(N, K, \mathcal{F})$, where N is the code length in bits, K is the number of information bits and $\mathcal{F}$ is a index vector which in literature is commonly called frozen bits. $\mathcal{F}$ is a vector of length $N-K$. Occasionally we use $\mathcal{F}^c$ to represent the K information bits indexing. The input to the channel is $\mathbf{x} \in \mathcal{X}$ and the output of

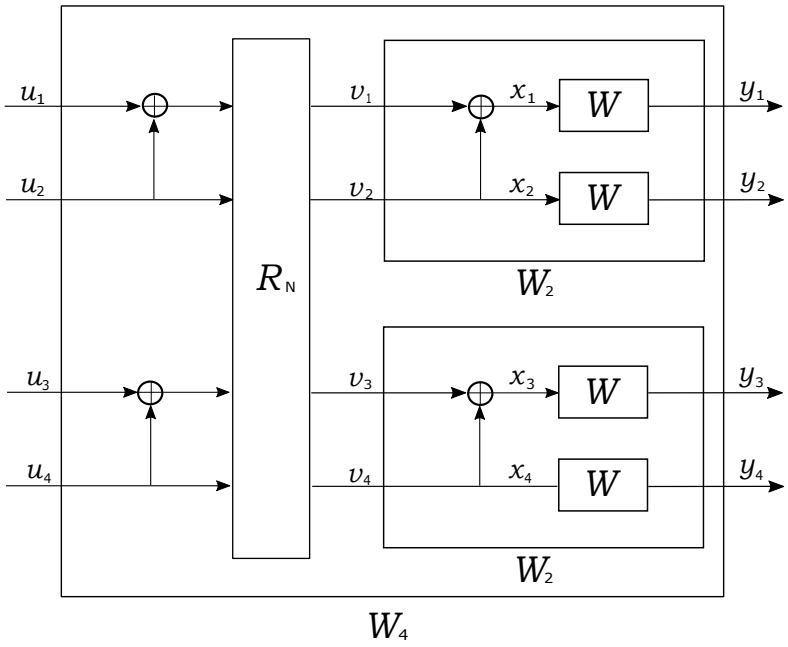

Figure 3.1. Four independent W copies are combined to produce a channel W_4.

the channel is $\mathbf{y} \in \mathcal{Y}$. The information vector $\mathbf{u} = U^N$ is the original information, while $\mathbf{x} = X^N$ is the input to the channel when N-channels are used to encode a vector message $\mathbf{u}$, and the output of the channel is also denoted by $\mathbf{y} = Y^N$. Polar codes are modeled by transition probabilities $W(y|x)$ or $W_N(\mathbf{y}|\mathbf{x})$ in vector form, which are a set of outputs conditioned on a set of inputs. The transition probabilities $W_N(\mathbf{y}|\mathbf{x})$ represents what are called virtual channels denoted by W.

Definition 3.1.1. A symmetric binary discrete memoryless channel (B-DMC) W_N has binary input $\mathcal{X} = \{0, 1\}$. W_N is assumed to be symmetric if and only if there exists a permutation over the outputs of the channel $\pi : \mathcal{Y} \longrightarrow \mathcal{Y}$ such that $\pi = \pi^{-1}$ and $W(y|0) = W(\pi(y)|1)$.

The idea of polar codes is to create $W_N^{(i)}$ from N-independent W-channels , where $1 \leq i \leq N$, such that N goes through a linear transformation which leads the channels to polarize. Figure 3.1 depicts a W_4 channel, where four independent W copies are combined $(W_4^{(1)}, W_4^{(2)}, W_4^{(3)}, W_4^{(4)})$ to produce the channel W_4.

3.2 Channel Polarization

Channel polarization consists of two operations, the channel combination phase and the channel splitting phase.

Combination Two independent copies of $W_{\frac{N}{2}}$ are combined to produce the channel W_N, which is represented by

$$W_N : X^N \to Y^N, \tag{3.1}$$

exemplified on Figure 3.1.

Splitting W_N is split back into a set of N binary-input coordinate channels, represented by

$$W_N^{(i)} : X^N \to Y^N \times X^{i-1} \tag{3.2}$$

When N grows large, the channels are polarized and so the symmetric capacity $I(W)$ is close to either 0 or 1. The symmetric capacity is the highest rate at which reliable communication is possible across W. Figure 3.2 is a plot of the symmetric capacity of independent virtual channels W, where it is shown that the fraction of indexes i for which $I(W_N^{(i)})$ is close to 1 is $I(W)$ and close to 0 is $1 - I(W)$.

Definition 3.2.1. Frozen bits are the indices for which $I(W_N^{(i)})$ are closer to 0. Frozen bits are the set of $N - K$ elements.

Definition 3.2.2. The symmetric capacity or mutual information of a B-DMC with input $\mathcal{X} = \{0, 1\}$ is defined by

$$I(W) \doteq \sum_{y \in \mathcal{X}} \sum_{x \in \mathcal{X}} \frac{1}{2} W(y|x) \log \frac{W(y|x)}{\frac{1}{2}W(y|0) + \frac{1}{2}W(y|1)} \tag{3.3}$$

where for reliable communications over a symmetric B-DMC at any rate is up to $I(W)$.

Definition 3.2.3. The Bhattacharyya parameter is defined by

$$Z(W) \doteq \sum_{y \in \mathcal{Y}} \sqrt{W(y|0)W(y|1)} \tag{3.4}$$

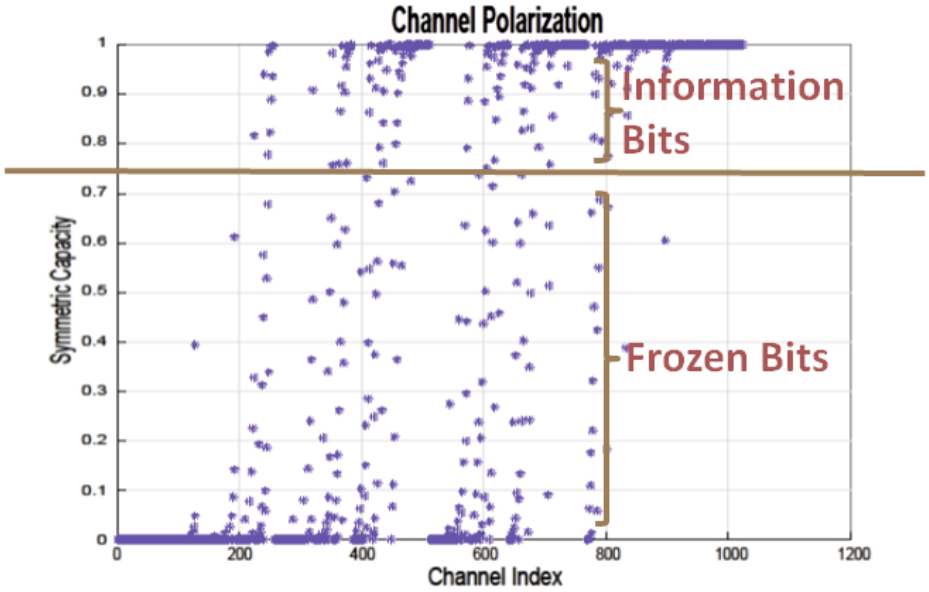

Figure 3.2. Plot of $I(W_N^{(i)})$ for a BEC.

The Bhattacharyya parameter is an upper bound on the probability of maximum-likelihood decision error when a channel W is used to transmit a single symbol 0 or 1.

To compute the Bhattacharyya vector z_N for a $Polar(N, K, \mathcal{F})$ through recursion is found by:

$$z_{2k,j} = \begin{cases} 2z_{k,j} - z_{k,j}^2 & : \quad x \leq j \leq k \\ z_{k,j-k}^2 & : \quad k+1 \leq j \leq 2k \end{cases}$$

for a polar code of block length $N = 2^n$, $1 \leq K \leq N$ where $k = 1, 2, 2^2, ..., 2^{n-1}$ and initializing with $z_{1,1} = \frac{1}{2}$, where $z_{1,1}$ is the so-called SNR-design or the "target block error rate".

By indexing each of the elements of the vector $z_N = \{z_1,\ z_2, \ldots, z_N\}$, and by arranging in ascending order according with the Bhattacharyya parameter value we obtained a permuted vector called π_N.

For example, when $N = 16$, the Bhattacharyya parameter provides a vector $z_{16} = \{1,\ 0.8999,\ 0.9634, 0.2275,\ 0.9853, 0.3462, 0.5327,\ 0.0078, 0.9922,\ 0.4673,\ 0.6538,\ 0.0147,\ 0.7725,\ 0.0366, 0.1001, 0\}$. Now, index each of the elements of the vector $z_{16} = \{z_1,\ z_2, z_3,\ z_4,\ z_5, z_6, z_7,\ z_8, z_9, z_{10}, z_{11}, z_{12},\ z_{13}, z_{14}, z_{15}, z_{16}\}$, and arrange

in ascending order according with the Bhattacharyya parameter. A permuted vector $\pi_{16} = \{z_{16}, z_8, z_{12}, z_{14}, z_{15}, z_4, z_6, z_{10}, z_7, z_{11}, z_{13}, z_2, z_3, z_5, z_9, z_1\}$ is obtained.

There exists a relation between the symmetric capacity and the Bhattacharayya parameter which is: $Z(W) \cong 1 - I(W)$.

3.3 Polar Encoder

For a $Polar(N, K, \mathcal{F})$ polar code, in this section we describe the encoding operation for a vector of information bits $\mathbf{u}$ of length K. The rate of the code is $R = \frac{K}{N}$, and we define n as $n = \log_2(N)$.

3.3.1 Generator Matrix of Polar Codes

By using Plotkin construction, we are interested in obtaining a generator matrix $\mathbf{G}^P$ for polar codes defined by:

$$\mathbf{G}^P = \mathbf{B}\mathbf{F}^{\otimes n}. \tag{3.5}$$

To clearly state the dimension of the matrix we write in the form $\mathbf{G}_N^P$ and $\mathbf{B}_N$, where N is the block length. Occasionally, we write $\mathbf{G}_{N,K}^P$ to specify the block length N and the information vector length K.

$$\mathbf{G}_N^P = \mathbf{B}_N\mathbf{F}^{\otimes n} \tag{3.6}$$

where $\mathbf{F}^{\otimes n}$ is the Plotkin matrix, $\mathbf{F} = \left[\begin{smallmatrix} 1 & 0 \\ 1 & 1 \end{smallmatrix}\right]$, $N = 2^n$, $\otimes$ is the Kronecker product and $\mathbf{B}$ is called the permutation matrix that has relation with the permutation matrix:

$$\mathbf{B}_N = \mathbf{R}_N(\mathbf{I}_2 \otimes B_{\frac{N}{2}}). \tag{3.7}$$

where $\mathbf{I}_2$ is the 2 by 2 identity matrix and $\mathbf{R}_N$ is the reverse shuffle permutation matrix. The reverse shuffle permutation matrix $\mathbf{R}_N$ is obtained by a shuffle on the matrix $S_{p,q}$ derived in Appendix A. $\mathbf{G}_N^P$ can be expanded as

$$\mathbf{G}_N^P = \mathbf{R}_N(\mathbf{F} \otimes \mathbf{I}_{\frac{N}{2}}) \cdot (\mathbf{I}_2 \otimes \mathbf{G}_{\frac{N}{2}}^P), \tag{3.8}$$

where the $\mathbf{I}$ is the identity matrix. The polar code that uses the generator matrix has complexity of $\mathcal{O}(N \log N)$.

3.3.2 Triangular Form

To obtain the generator matrix $\mathbf{G}_N^P$ by equation (3.6) two steps are required. First step is to use the Plotkin construction to get $\mathbf{F}^{\otimes n}$. Secondly, perform the bit reversal $\mathbf{B}_N$, which is a transformation the virtual channels that is produced by the channel combination phase.

Before applying the transformation of the bit reversal, it is possible to obtain a polar code in triangular form by simple using the Plotkin construction [3].

The following example shows a generator matrix in triangular form and followed by polar code generator matrix. To obtain a polar code $Polar(8, K, \mathcal{F})$ with block length $N = 8$, first a triangular generator matrix is obtained by constructing the matrix $\mathbf{F}^{\otimes \log_2(8)}$:

$$\mathbf{F}^{\otimes 3} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}. \tag{3.9}$$

Then $\mathbf{F}^{\otimes 3}$ is multiplied by the permutation matrix $\mathbf{B}_N$. The generator matrix for $Polar(8, K, \mathcal{F})$ is

$$G_8^P = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \tag{3.10}$$

A full rank polar code generator matrix G_8^P is obtained. This generator matrix implies that non frozen bits $\mathcal{F} = \{\emptyset\}$ are used for this polar code.

3.3.3 Subcodes by Plotkin Construction

By using the full rank polar code generator matrix and the Bhattacharyya parameter, it is possible to obtain another subcode of the polar code. From the full rank generator matrix, it is enough to remove as many rows as frozen bits $\mathcal{F}$ are desired to construct a sub-polar code. The rows to be removed are obtained by selecting the Bhattacharyya indexes z that have lower value $\mathcal{F}$, a total of $K - N$ elements are taken form the vector z.

For example, let us select a polar code $Polar(N, K, \mathcal{F})$ of $N = 16$ with 10 frozen virtual channels, and a total of $K = 6$ information bits. A full rank generator matrix $\mathbf{F}^{\otimes 4}$ is obtained, which is shown in (3.13), this matrix is in a lower triangular matrix form:

$$\mathbf{F}^{\otimes 4} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}. \tag{3.11}$$

Finally, the polar subcode with $N = 16$ and $K = 6$ is obtained by removing some rows of the full rank matrix. The indexes of the vector $\mathbf{z}_{16}$ arranged in ascending order are $\pi_{16} = \{z_{16}, z_8,\ z_{12}, z_{14}, z_{15},\ z_4, z_6, z_{10}, z_7, z_{11}, z_{13},\ z_2, z_3, z_5, z_9, z_1\ \}$. Where the frozen bits $\mathcal{F}$ are $\mathcal{F} = \{z_{16}, z_8,\ z_{12}, z_{14}, z_{15},\ z_4\}$. The generator matrix $\mathbf{G}^{P}_{16,6}$ of the subcode $Polar(16, 6, \mathcal{F})$ is

$$\mathbf{G}_{16,6}^{P} = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{matrix} z_4 \\ z_8 \\ z_{12} \\ z_{14} \\ z_{15} \\ z_{16} \end{matrix} \tag{3.12}$$

The generator matrix for the full rank polar code is

$$\mathbf{G}_{16}^{P} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \tag{3.13}$$

by equation (3.6).

To obtain another generator matrix $G_{16,6}^{P}$ of the polar code $Polar(16, 6, \mathcal{F})$, it is possible to apply the permutation $\mathbf{R}_N$:

$$\mathbf{G}_{16,6}^{P} = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}. \tag{3.14}$$

Both generator matrix $\mathbf{G}_{16,6}^{P}$, shown by equation (3.12), and (3.14) are valid generator matrix for the polar code $Polar(16, K, \mathcal{F})$. In fact, in [10], the authors demonstrated that variation of the selection of frozen bits leads to a change in the polar code performance.

3.3.4 Polar Encoder Using Generator Matrix $\mathbf{G}_N$

Codewords for a $Polar(N, K)$ code are generated using $\mathbf{G}_N$ as

$$\mathbf{x} = \mathbf{u}_{\mathcal{F}^c}\mathbf{G}_{\mathcal{F}^c} + \mathbf{u}_{\mathcal{F}}\mathbf{G}_{\mathcal{F}}, \tag{3.15}$$

where $\mathcal{F}$ are the frozen bits and $\mathcal{F}^c$ corresponds to the non-frozen bits.

Or a simpler encoding expression

$$\mathbf{x} = \mathbf{u}\mathbf{G}_N \tag{3.16}$$

knowing where are the frozen bits $\mathcal{F}$ located.

3.4 Polar Decoder

In the literature, various decoders have been proposed for polar codes. The first decoders used for polar codes are the density evolution based designed polar codes, and successive cancellation decoders (SCD) for polar codes. In [10] was introduced the permuted successive cancellation (PSCD) decoder, which is a variant of the original SCD for polar codes. And the most recent decoding approach is the so-called list decoding for polar codes [**?**].

A polar decoder receives a noisy sequence $\mathbf{y}$ with probability $W_N(\mathbf{y}|u^N)$ from a noisy channel. The successive cancellation decoder observes $(\mathbf{y}^N, \mathcal{F})$ and generates an estimate $\hat{u}^N$ by computing LLRs given by:

$$\hat{u}_i = \begin{cases} 0 & \text{if } L_N^{(i)}(\mathbf{y}_1^N, \hat{u}^{i-1}) \geq 1 \\ 1 & \text{if } otherwise \end{cases}$$

. LLRs are initialized with

$$L_N^{(i)}(\mathbf{y}, \hat{u}^{i-1}) = -4\frac{\sqrt{E_c}}{N_0}y \tag{3.17}$$

where $E_c = \frac{K}{N}N_0 10^{E_b N_0/10}$.

3.5 Polar construction

One of the important steps for polar coding is the construction by selecting the best K bit-channels among N. The selection of K bit-channels is done by the Bhattacharyya parameter or the symmetric capacity, explain in the previous section. The corresponds to selection of best K bit-channels in terms of the bit error rate at a given value of RE_b/N_0 is defined as the design-SNR.

Algorithm 1 Frozen bit selection using Bhattacharyya parameter

procedure FROZENBITINDEX(N, K, design$-SNR$ EdB=(RE_b/N_0 in dB))
$S = 10^{EdB/10}$ and $n = \log_2 N$
$z^{(0)} \in R^N$, initialize $z^{(0)}[0] = exp(-S)$
for $j = 1 : n$ **do**
$u = 2^j$
for $t = 0 : \frac{u}{2} - 1$ **do**
$T = \mathbf{z}^{(0)}[t]$
$\mathbf{z}^{(0)}[t] = 2T - T^2$
$\mathbf{z}^{(0)}[u/2 + t] = T^2$
$\mathcal{F}$= indices_of_greatest_elements($z^{(0)}, N - K$) \\ Find indices of the greatest $N - K$ elements.
Return $\mathcal{F}$

Chapter 4

Lattices

In this chapter, a general description and some definitions of lattices are provided. First we introduce the concept of lattice and its Voronoi region. Then we describe another important concept to measure the density of a lattice, which is the volume-to-noise ratio. We introduce the concept of nested lattices which are used to form lattices layer by layer. Finally, the construction of lattices is provided in detail.

We formed lattices with two different constructions. Construction A uses a single binary code, and construction D uses nested binary codes. It is known that construction A produces lattices which are effective in lower dimensions. On the other hand, construction D is effective in higher dimensions. The effectiveness usually increases with the dimension; good lattices tend to be "perfect" as the dimension goes to infinity.

4.1 Definition of Lattices

Lattices are effective structures which solves various geometric, coding and quantization problems. A lattice Λ is a linear additive subgroup of the N-dimensional Euclidean space $\mathbb{R}^N$ [2]. A lattice Λ is formed by all integer combinations of basis vectors.

The generator matrix $\mathbf{G}^\Lambda$ for a lattice Λ is created by N-dimensional basic vectors. In this work, row basis vectors are used.

$$\mathbf{G}^\Lambda = \begin{bmatrix} \text{——} & \mathbf{g}_1 & \text{——} \\ \text{——} & \mathbf{g}_2 & \text{——} \\ & \vdots & \\ \text{——} & \mathbf{g}_N & \text{——} \end{bmatrix} \tag{4.1}$$

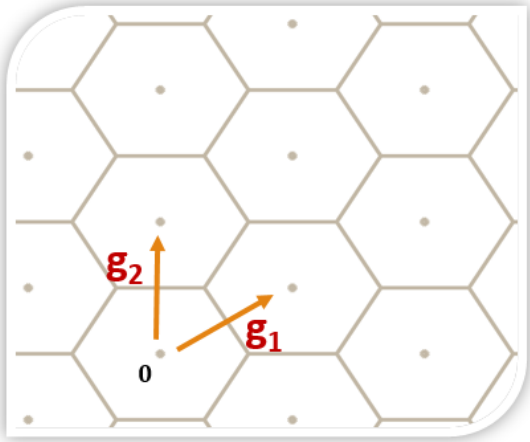

Figure 4.1. Representation of a 2-dimensional periodic linear arrangement of points. This hexagonal lattice is formed by $\mathbf{g}_1$ and $\mathbf{g}_2$ basis vectors.

A lattice point $\mathbf{x} \in \Lambda$ in a N-dimensional $\mathbb{R}^N$ space is defined by

$$\mathbf{x} = \mathbf{b} \cdot \mathbf{G}^{\Lambda}, \tag{4.2}$$

where $\mathbf{b} \in \mathbb{Z}^n$ is a row vector. The lattice point $\mathbf{x}$ has the form $\mathbf{x} = \{x_1, x_2, \ldots, x_N\}$.

The elements of the vector $\mathbf{x}$ can be computed by

$$x_j = \sum_{i=1}^{n} g_{i,j} b_i. \tag{4.3}$$

A 2-dimensional hexagonal lattice Λ is formed with the generator matrix

$$G^{\Lambda} = \begin{bmatrix} \mathbf{g}_1 \\ \mathbf{g}_2 \end{bmatrix}, \tag{4.4}$$

where $\mathbf{g}_1 = [\,1 \; 0.5\,]$ and $\mathbf{g}_1 = [\,0 \; 1\,]$. The hexagonal lattice Λ is represented graphically in Figure 4.1. Lattice points of this lattice Λ are created by equation (4.2). Once the set of lattice points are created, it is possible to perform coding by using the generator matrix. However, before beginning with encoding of lattices, let us introduce the Voronoi region.

Definition 4.1.1. The Voronoi region of a lattice Λ is defined as the set of all points in $\mathbb{R}^n$ that are closer to a lattice point than to any other lattice point. Mathematically, the volume of the Voronoi region $V(\Lambda)$ is given by

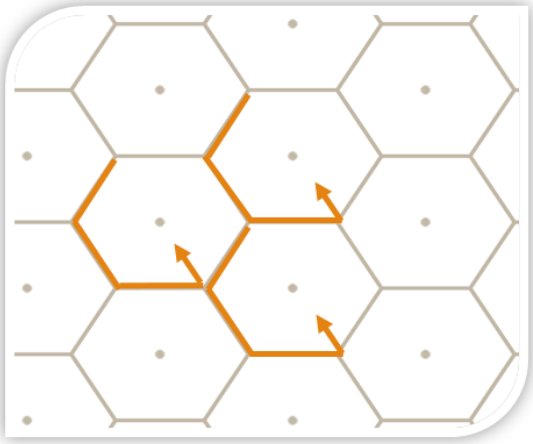

Figure 4.2. Each lattice point of the 2-dimensional lattice has a Voronoi region of hexagonal shape. Each lattice point shares 6 neighbors lattice points. Each Voronoi region, 3 neighbors edges are chosen arbitrary to be part of a particular lattice point.

$$V(\Lambda) = |det(G)|. \tag{4.5}$$

Figure 4.2 shows the Voronoi region of the hexagonal lattice Λ forming an hexagonal shape as well. In any lattice, points that are located on borders of the Voronoi region have the same Euclidean distance to two or more lattice points. In such case, lattice points are arbitrarily chosen to be part of which Voronoi region.

For example, in the 2-dimensional hexagonal lattice, there exists 6 neighbor edges. Arbitrary 3 edges are chosen to form a single Voronoi region for each lattice point.

4.2 Volume-to-Noise-Ratio

Usually the notation of signal-to-noise ratio is used for power-constrained channels where only a finite number of codewords or signals are transmitted. However, for the lattices case, the code has an infinite number of lattice points, leading to use of a different metric, called the Volume to Noise Ratio (VNR).

Definition 4.2.1. The VNR of an N-dimensional lattice in the presence of AWGN with variance σ^2, is defined as

$$VNR = \frac{V^{\frac{2}{N}}(\Lambda)}{\sigma^2} \tag{4.6}$$

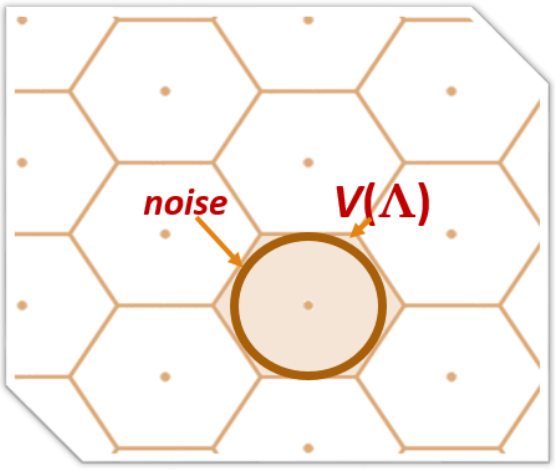

Figure 4.3. VNR is the relation between the volume of the lattice and the noise sphere. This figure shows a $2-$dimensional lattice and the representation of a $N-$dimensional AWGN channel.

The VNR is the measure of the density of the lattice. Figure 4.3 shows the relation between the Voronoi region and the noise sphere. The normalized VNR (NVNR) is calculated by

$$NVNR = \frac{|V(\Lambda)|^{2/N}}{2\pi e\sigma^2} \tag{4.7}$$

The VNR and NVNR are dimensionless numbers which are invariant to scaling or rotation of the lattice.

When a $VNR = 1$ corresponds to the capacity, and reliable decoding is only possible if

$$\sigma^2 \leqslant \frac{V(\Lambda^{\frac{2}{N}})}{2\pi e} \tag{4.8}$$

4.3 Nested Lattices

Consider the example of the ruler which has two scales. A large scale (in "cm") and another fine scale (in "mm"). Now, if we consider two marks of the large scale (cm), and take all the points in between using the fine scale. This new set can be thought to be a finite codebook. The union of all such finite codebooks gives the whole ruler.

Nested lattices follow a similar same idea. Consider two lattices Λ and Λ_{fine}, where Λ is a lattice with higher scale than the lattice Λ_{fine} with a fine scale. This pair of

N-dimensional lattices are nested if

$$\Lambda \subset \Lambda_{fine}. \tag{4.9}$$

Λ is a sublattice of Λ_{fine} denoted by:

$$\Lambda/\Lambda_{fine}. \tag{4.10}$$

It follows that each basis vector $g_{1,i}$ of Λ is an integer combination of the basis vector $\mathbf{g}_{fine,1}, \mathbf{g}_{fine,2}, \ldots, \mathbf{g}_{fine,n}$ of Λ_{fine}

$$\mathbf{g}_{1,i} = \sum_{k=1}^{n} j_{k,i}\mathbf{g}_{fine,k}. \tag{4.11}$$

4.4 Lattice Construction

In this work, lattices are formed based on linear binary codes. To produce lattices, two different types of construction are used. To construct efficient lattices by: crossing-sections of an error correcting code, construction A is used; or by building-up packing by layers construction D is used.

4.4.1 Construction A

The packing constructed by cross-sections is called construction A, this construction is effective in up to 15 dimensions. Conway and Sloane [2] defined construction A as the construction which specifies a set of centers for a sphere packing in $\mathbb{R}^N$. A point $\mathbf{x} = (x_1, ..., x_N)$ is a center if and only if $\mathbf{x}$ is congruent modulo 2 to a codeword from the binary code $\mathcal{C}$ of (N, M, d_{min}), N-dimensional packing with M codewords (or M spheres) and minimum distance d_{min} apart.

Definition 4.4.1. Construction A is a method for generating a lattice by lifting a linear binary code $\mathcal{C}$ to the Euclidean space. We use $\mathbf{x} \mod 2 = (x_1 \mod 2, ..., x_n \mod 2)$ to denote a modulo-2 reduction of each of the components of $\mathbf{x} \in \mathbb{R}^N$.

The mathematical expression for a sphere packing $\Lambda(\mathcal{C})$ in $\mathbb{R}^N$ is given by

$$\Lambda_{\mathcal{C}} = \sqrt{2}\mathbf{x}(mod\ 2) \in \mathcal{C} \tag{4.12}$$

With this expression, the origin is always a center. The centers consist of all vectors which can be obtained from the codewords of C by adding arbitrary even numbers to the components and then dividing by $\sqrt{2}$

However, if we are interested in constructing lattices which always produces integral lattices. The $\sqrt{2}$ needs to be omitted from equation (4.12). That is,

$$\Lambda_{\mathcal{C}} = \mathbf{x}(mod\ 2) \in \mathcal{C} \tag{4.13}$$

$\Lambda_{\mathcal{C}}$ is spanned by an extended $N \times (N + K)$ generator matrix

$$\mathbf{G}^{\Lambda_{\mathcal{C}}} = [\mathbf{G}|qI_N], \tag{4.14}$$

where **G** is the generator matrix of $\mathcal{C}$ and I_N is the $N \times N$ identity matrix, q is the q-ary linear code $\mathcal{C}$, which induces a modulo-q lattice given by

$$\Lambda_{\mathcal{C}} = \mathbf{x} \in \mathbb{R}^N : (mod\ q) \in \mathcal{C} \tag{4.15}$$

The generator matrix can be reduced to a standard $N \times N$ matrix. Where the $\mathcal{C}$ has a systematic representation:

$$\mathbf{G}^{\Lambda_{\mathcal{C}}} = \begin{bmatrix} I_K & | & 0 \\ P & | & qI_{N-K} \end{bmatrix}, \tag{4.16}$$

and P is an $(N - K) \times K$ matrix. That is, each codeword $\mathbf{c} = G\mathbf{w}$ consist of the information vector **w** itself (the systematic part), concatenated with $P\mathbf{w}$ (the parity symbols).

4.4.2 Construction D

Construction D was introduced in [11] by Barnes and Sloane. Construction D works with a set of generator matrices, and more precisely, it requires a set of nested binary linear codes $\mathcal{C}_0 \supseteq \mathcal{C}_1 \supseteq ... \supseteq \mathcal{C}_a$ with parameters $[N, K_i, d_i]$ for each binary linear code $\mathcal{C}_i$. $\mathcal{C}_0$ is the universe code $[N, N, 1]$. There exists a restriction given by the minimum distance $d_i \geq \frac{4^i}{\gamma}$, where $\gamma = 1$ or 2, for $i = 1, ..., a$.

To form lattices with construction D, first a basis $\mathcal{C}_1, \ldots, \mathcal{C}_N$ is chosen from $\mathcal{C}_i$ for $i = 1, \ldots, a$ such that a generator matrix $\mathbf{G}^{\Lambda}_{N,N}$ is obtained for the universal code $\mathcal{C}_0$.

$$\mathcal{C}_1, \ldots, \mathcal{C}_N \text{ span}\mathcal{C}_i \rightarrow \mathbf{G}^{\Lambda}_{N,N} \tag{4.17}$$

for $i = 1, \ldots, a$.

Secondly, permute rows $\mathcal{C}_1, \ldots, \mathcal{C}_N$. If the permuted matrix is an upper triangular matrix, a map should be defined by

$$\pi_i : \mathbf{G}_2 \rightarrow \mathbf{Q} \tag{4.18}$$

by $\pi_i(x) = \frac{x}{2^{i-1}}$ when $x = \{0, 1\}$ and for $i = 1, \ldots, a$. And letting the same symbol π_i denotes the map

$$\pi_i : \mathbf{G}_2^N \rightarrow \mathbf{Q}^N \tag{4.19}$$

by $\pi_i(x_1, \ldots, x_N) = (\pi_i(x_1), \ldots, \pi_i(x_N))$ and $K_{a+1} = 0$.

The obtained lattice Λ is

$$\Lambda = l + \sum_{i=1}^{a} \sum_{j=1}^{k_i} \alpha_j^{(i)} \pi_i(\mathcal{C}_j) \tag{4.20}$$

where $l \in (2\mathbf{Z})^N$ and $\alpha_j^{(i)} = 0, 1$.

The set of sub-lattices $\Lambda_1 \subset \ldots \subset \Lambda_a$ induces a partition of Λ into equivalence classes module definition Λ' denoted by Λ/Λ'. The order is denoted by $|\Lambda/\Lambda|$, an example is the binary partition denoted by $\Lambda/\Lambda' = 2$. A n-dimensional lattice partition chain is denoted by $\Lambda/\Lambda_1/.../\Lambda_{r-1}/\Lambda'$ for $r \geq 1$. For each partition Λ_{l-1}/Λ_l, a code C_l selects a sequence of representative coset of Λ_l.

Chapter 5

Polar Lattices

The optimal input distribution for a lattice Λ is uniform distribution, and to achieve the channel capacity over an AWGN channel, AWGN-good lattices are needed. In [12], multilevel construction of AWGN-good lattices is shown. In this work, we will follow the idea to construct lattices by single levels (construction A) and multilevel (construction D) for polar lattices.

5.1 Polar Lattices

A polar lattice Λ_P is specified by $[N, K]$, where N is the code length in bits and K is the number of information bits. Frozen bits $\mathcal{F}$ are used to select subcodes of the polar code $Polar(N, K, \mathcal{F})$. $\mathcal{F}$ is a subset of $N - K$. The transition probabilities $W(y|x)$ is given by the polar code.

An information vector $\mathbf{b} = (b_1, \ldots, b_N)$ is the original information. A polar lattice enocode the sequence $\mathbf{b}$ to obtain a lattice point $\mathbf{x} = (x_1, \ldots, x_N)$, which is the input of the channel when N-channels are used. A received sequence $\mathbf{y} = (y_1, \ldots, y_N)$ is the input of the polar lattice decoder. $\hat{\mathbf{b}} = (\hat{b}_1, \ldots, \hat{b}_N)$ is the estimated $\mathbf{b}$ sequence which is pass to the final destination.

A sequence $\hat{\mathbf{x}} = (\hat{x}_1, \ldots, \hat{x}_N)$ is the estimate vector of the lattice point, which is the output of the decoder for construction A or the decoder for multilevel lattices..

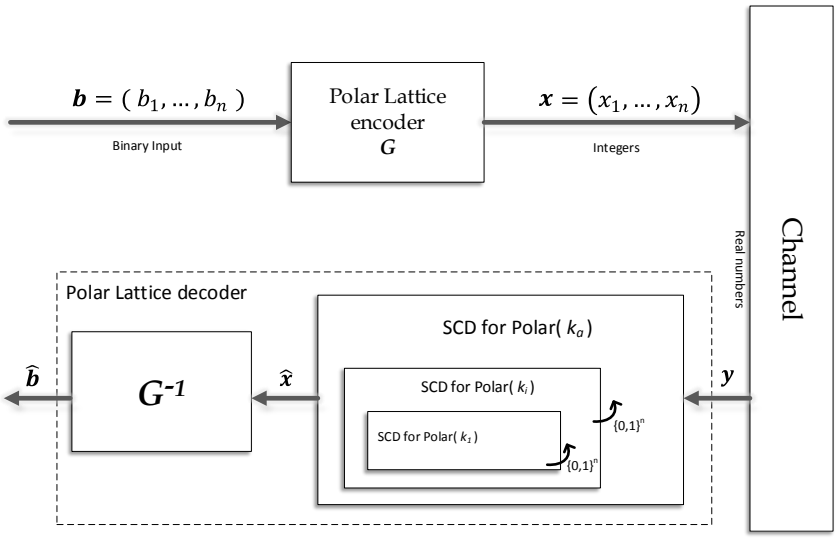

Figure 5.1. Block diagram of Polar Lattices.

5.2 Proposed Communication System Model

The proposed model for this work is depicted on Figure 5.1. A sequence **b** is encoded by a polar lattice encoder, the output of the encoder is a lattice point **x** which is transmitted over a noisy channel. A received sequence **y** is decoded by a Polar lattice decoder to obtain an estimated sequence $\hat{\mathbf{b}}$.

The polar lattice decoder comprehend a binary decoder and an integer decoder. The binary decoder uses a code $\mathcal{C}$ to obtain and sequence $\hat{\mathbf{x}}$ that is passed to the integer decoder to obtain a final $\hat{\mathbf{b}}$ sequence. Any binary decoder that can decode polar codes is suitable. In this work, a successive cancellation decoder (SCD) is used. The integer decoder is simple the inverse of the polar lattice generator matrix.

To build lattices by construction A, a single polar code is required. However, to build a lattice by construction D, a set of nested lattices is required.

5.3 Polar Lattices Formed by Construction A

In this thesis, polar lattices formed by construction A are represented by $\Lambda_P[N, \mathbf{K}]$. Where **K** is a vector of just one element $\mathbf{K} = K$ Construction A requires a binary polar code $Polar(N, K, \mathcal{F})$. For polar lattices, any polar code is suitable. However, in order to easily form a lattice by construction A, equation (3.6) is used. A generator matrix

for polar codes G^P is obtained, in this matrix the Plotkin construction operation is performed first, and then the permutation matrix $\mathbf{R}_N$ is apply.

A polar lattice Λ_P is then obtained by

$$\Lambda_P = \mathcal{C} + 2(\mathbb{Z})^N, \tag{5.1}$$

and its generator matrix $\mathbf{G}^{\Lambda_P}$ is obtained as

$$G^{\Lambda_P} = \textbf{replace } \{F^c\} \text{from } \left[2I_N\right] \text{ to } \left[\mathbf{G}^P\right] \tag{5.2}$$

where $\mathbf{G}^{\Lambda_P}$ is a square matrix. It is obtained by concatenating the generator matrix $\mathbf{G}^P$ of the polar code with an identity matrix of length N. The index of the frozen bits are replaced by the identity matrix.

5.3.1 Encoder

The polar lattice Λ_P has as encoder a generator matrix G^{Λ_P} constructed with equation (5.1). The generator matrix G^{Λ_P} produces lattice points $\mathbf{x}$, which are the input of the channel.

$$\mathbf{x} = \mathbf{b} \cdot \mathbf{G}^{\Lambda_P}, \tag{5.3}$$

where $\mathbf{b}$ is a binary vector and $\mathbf{x}$ is a integer vector.

5.3.2 Construction A Polar Lattice Decoder

To find the closest lattice point from the noisy received sequence $\mathbf{y} \in \mathbb{R}^N$. The following algorithm is used for construction A.

1. Reduce the receive sequence $\mathbf{y}$ to a range of $\mod 2$ and denote it as $\mathbf{yp}$.
2. Subtract all y_i with the vector yp_i and denote it as a vector z_i.
3. Denote $\mathbf{S}$ as the set of i for which $yp_i > 1$.
4. Replace y_i with $2 - y_i$ for $i \in \mathbf{S}$.
5. Because $\mathbf{y}$ is now in the Voronoi region, apply the decoder for the polar code $Polar(N, K, \mathcal{F})$ to $\mathbf{y}$. The obtained vector denote it as $\mathbf{t}$.
6. Replace t_i with $2 - t_i$ for $i \in S$.

7. The closest lattice point to $\mathbf{y}$ is denoted by $\hat{\mathbf{x}} = t_i + 2z_i$.

5.3.3 Example of the Construction A

Let construct a 15-dimensional polar lattice $\Lambda_P(K)$ with a binary code $\mathcal{C}_1$ which is a polar code $Polar(N = 16, K = 5)$. The generator vectors for $\mathcal{C}_1$ are the rows of the following matrix

$$\mathbf{G}_{16,5}^{P} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{matrix} z_8 \\ z_{12} \\ z_{14} \\ z_{15} \\ z_{16} \end{matrix} \tag{5.4}$$

We have a total of $K = 5$ independent rows obtained from the full rank generator matrix that was shown in previous chapter in the matrix (3.13).

The indexes of the vector z_{16} obtained by the Bhattacharyya parameter and arranged in ascending order are $\pi_{16} = \{z_{16}, z_8, z_{12}, z_{14}, z_{15}, \sout{z_4, z_6, z_{10}, z_7, z_{11}, z_{13}, z_2, z_3, z_5, z_9, z_1}\}$. Where the frozen bits $\mathcal{F}$ are dashed by a horizontal line. The polar code used is denoted as $Polar(16, 5, \{z_4, z_6, z_{10}, z_7, z_{11}, z_{13}, z_2, z_3, z_5, z_9, z_1\})$ with generator matrix $G_{16,5}^{P}$ formed with the information bits $\mathcal{F}^c = \{z_{16}, z_8, z_{12}, z_{14}, z_{15}\}$. The rate is $R = \frac{5}{16}$.

To form a polar lattice with construction A, let us use the equation (5.1). A generator matrix G^{Λ_P} is obtained by (5.2). The rows (frozen bits $\mathcal{F}$) of the generator matrix $G_{16,16}^{P}$ are replace by two times the same rows of the identity matrix I_N. Obtaining the following shape

$$G_{16}^{\Lambda_P} = \begin{bmatrix} 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \tag{5.5}$$

5.4 Polar Lattices Formed by Construction D

Polar lattices formed by construction D are represented by $\Lambda_P(K^1, K^2, \ldots, K^i)$, which are lattices formed by a group of nested binary codes, $\mathcal{C}_a \subseteq \mathcal{C}_{a-1} \subseteq \cdots \subseteq \mathcal{C}_1$. In this work, polar lattices Λ_P are constructed using construction D and selecting different nested polar codes $\mathcal{C}_i$, where $1 \leq i \leq a$.

Construction D is one of the various lattice constructions which produces lattices from a group of nested binary codes. Each subcode of the polar code is obtained by choosing rows from the full rank generator matrix.

Rows from the subcode are the information bits of a polar code, meaning that the frozen bit index where removed from the full rank generator matrix. Once subpolar codes are identified, a construction D is used with the following equation

$$\Lambda_P = \mathcal{C}_a + 2\mathcal{C}_{a-1} + ... + 2^{a-1}\mathcal{C}_1 + 2^a(\mathbb{Z})^N, \tag{5.6}$$

where $1 \leq i \leq a$, and it is a N-dimensional real lattice.

Polar lattices constructed by construction D are represented by $\Lambda_{P_i}(K_i^1, K_i^2)$, where

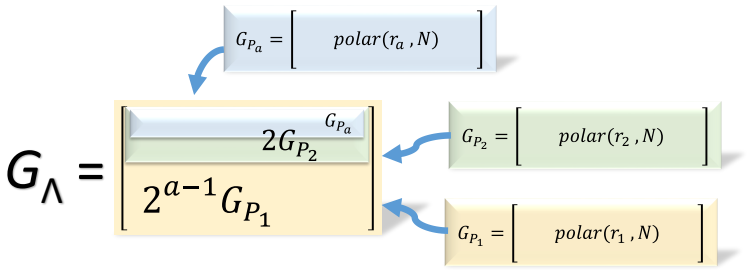

Figure 5.2. Polar lattices generator matrix $\mathbf{G}^{\Lambda_P}$. This generator matrix form a three-level polar lattice.

i is the polar lattice with polar code $\mathcal{C}_i$. K^1 and K^2 are the information bits for the first and second polar sub-code respectively.

For construction D, the restriction is on the minimum distance of the code $d_{min(a)} \geq \frac{4^a}{j}$, where j is either 1 or 2, and a is the level of the lattice. All levels contribute equality to the Lattice d^2_{min}.

5.4.1 Minimum Distance d_{min} of the Polar Code

A very important and computationally demanding problem is to find the minimum distance of a code. An approach to find the minimum distance for polar lattices is the use of the minimum distance for the Plotkin construction. In [13], authors obtained the minimum distance of the additive Reed-Muller (ARM) code constructed using Plotkin construction.

They showed that the minimum distance for any $ARM(r, m)$ code obtained by applying the Plotkin construction with length $N = 2^m$; size 2^K codewords given by

$$K = \sum_{i=0}^{r} \binom{m}{i}, \tag{5.7}$$

the minimum distance is 2^{m-r}.

In [3] it was shown that Reed-Muller codes are constructing by using Plotkin construction and then remove from the generator matrix the row vectors with Hamming weights $\geq 2^{n-r}$. The difference with polar codes constructed with Plotkin construction is the selection of row vectors, in polar codes row vectors are removed according to the Bhattacharyya parameter or the symmetric capacity.

To construct polar lattices we set the minimum distance to be $d^\Lambda_{min} = \frac{4^{(a)}}{2}$, where a is the level of the polar lattice. Assuming the minimum distance, then we can set the number of information bits K given by

$$K = \sum_{i=0}^{r \leq m - \log(\frac{4^{(a)}}{2})} \binom{m}{i}, \tag{5.8}$$

where the inequality is obtained by

$$2^{m-r} = d_{min}^{\Lambda} \geq \frac{4^{(a)}}{2}. \tag{5.9}$$

5.4.2 Multilevel Encoder

Multilevel refers to the number of nested codes selected to construct lattices. When only one code is selected is called single level lattice. When two nested codes are used, the lattice is called two level lattice, and continuing in the same way.

The generator matrix for polar lattice codes $\mathbf{G}^{\Lambda_P}$ is constructed based on equation (5.6) and it is depicted graphically on the figure 5.2. The generator matrix on figure 5.2 forms a three-level polar lattice, because three different polar codes are used.

The generator matrix G^{Λ_P} produces lattice points $\mathbf{x}$, which are the input to the channel.

$$\mathbf{x} = \mathbf{b} \cdot \mathbf{G}^{\Lambda_P}, \tag{5.10}$$

where $\mathbf{b}$ are integers and $\mathbf{x}$ are binary vectors.

5.4.3 Multilevel Decoder

In [14], a multilevel decoding is used. The decoding is done on each independent subpolar code and passes the decoded sequence to the next level. Each level is treated as construction A, once the decoder for the code C obtained the vector v, it is passed to the next level.

Figure 5.3 shows the block diagram of a polar lattice constructed with two levels.

A multistage soft decision decoder is used to find the closest lattice point from the noisy received sequence $\mathbf{y} \in \mathbb{R}^N$. The algorithm of the multistage lattices with construction D is shown as follows.

Lattices are constructed by $\Lambda_P = \mathcal{C}_a + 2\mathcal{C}_{a-1} + ... + 2^{a-1}\mathcal{C}_1 + 2^a(\mathbb{Z})^N$, with a-binary codes.

1. Checking if it is already the a-time that the algorithm is performed. If so, $round(\mathbf{y})$

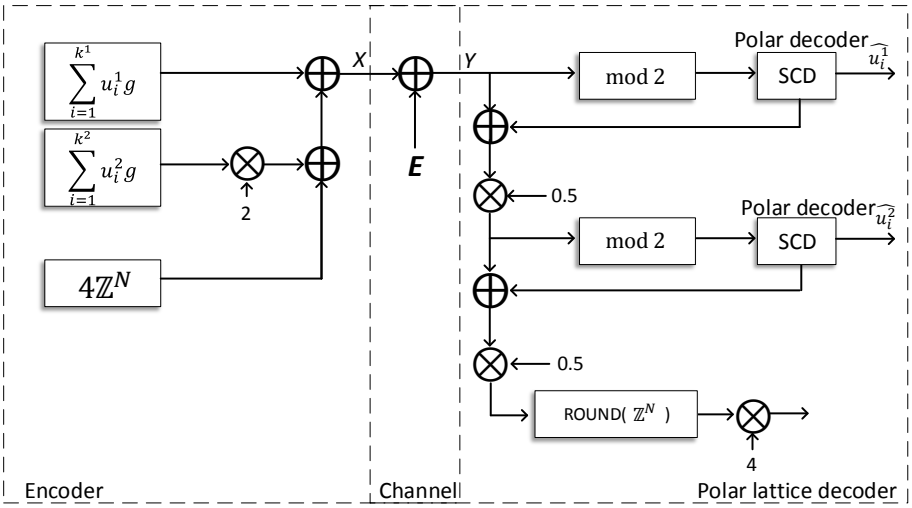

Figure 5.3. A polar lattice with two levels.

and denote it as **v**, return to the previous step to continue with the algorithm. If it is not the a-time, proceed with the algorithm to step 2.

2. Reduce the receive sequence **y** to a range of $(y \mod 2) - 1$ and denote it as **yp**.

3. For the closest even (**e**) and odd (**o**) integers to yp_i, denote **b** as

$$b_i = 2(\frac{e_i + o_i}{2} - yp_i) \quad : \quad e_i < o_i \tag{5.11}$$

$$b_i = 2(-\frac{e_i + o_i}{2} + yp_i) \quad : \quad \text{otherwise} \tag{5.12}$$

4. Because **b** is now in the Voronoi region, apply the decoder for the polar code $Polar(N, K, \mathcal{F})$ to **b**. The obtained vector denote it as **t**.

5. Replace y_i with $\frac{y_i - t_i}{2}$. Repeat the algorithm with this new value of **y**. The value of the variance σ is now consider as half for the $Polar(N, K, \mathcal{F})$ decoder. Continue the algorithm with new value of **v**.

6. Because it is a recursive algorithm, return $t + 2v$ and denote it as **v**.

7. The closest lattice point to **y** is denoted by $\hat{\mathbf{x}} = round(\mathbf{y})$.

5.4.4 Two Level Polar Lattice Example

For the coding polar lattice of two levels, the nested binary codes $\mathcal{C}_1$ and $\mathcal{C}_2$ should satisfy $\mathcal{C}_2 \subseteq \mathcal{C}_1$. Let us construct a polar lattice $\Lambda(N = 16, K_1 = 15, K_2 = 5)$ code with nested polar codes. $\mathcal{C}_1$ is the polar code $polar(16, 15, \mathcal{F}_1)$ and $\mathcal{C}_2$ is the polar code $polar(16, 5, \mathcal{F}_2)$. Where the frozen bits are $\mathcal{F}_1 = \{z_1\}$ and $\mathcal{F}_2 = \{z_4, z_6, z_{10}, z_7, z_{11}, z_{13}, z_2, z_3, z_5, z_9, z_1\}$.

The generator matrix for the $polar(16, 15)$ code is:

$$G^P_{16,15} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \tag{5.13}$$

and the generator matrix for the $polar(16, 5)$ code is:

$$G^P_{16,5} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \tag{5.14}$$

Basically, Construction D is used to embed binary numbers in the real space. By follow the formula (5.6), that can be reduced to

$$\Lambda_P = \mathcal{C}_2 + 2\mathcal{C}_1 + 4(\mathbb{Z})^N, \tag{5.15}$$

a generator matrix for this polar lattice is:

$$G^{\Lambda_P}_{N=16,K_1=15,K_2=5} = \begin{bmatrix} 4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 2 & 2 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 2 & 0 & 0 & 2 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 2 & 0 & 2 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 2 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 2 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \tag{5.16}$$

Chapter 6

Numerical Results

In this thesis we show the performance simulation of various polar lattices formed by construction A and construction D. To find which lattice produces the lowest bit error rate on certain VNR, polar lattices are designed by variation of the designing rate. In other words, the block length of the polar lattice is fixed and the information bits K are variable.

Results on this thesis are aim to be used as a guideline to construct efficient polar lattices, not only for small dimension (construction A) but also higher dimension using construction D.

Simulation results are showed in four different plots:

- SER vs VNR. Comparison of the integer lattice point $\mathbf{x}$ and the estimated real lattice point $\hat{\mathbf{x}}$.
- WER vs VNR. Comparison of the integer lattice point $\mathbf{x}$ and the estimated real lattice point $\hat{\mathbf{x}}$.
- BER vs VNR. Comparison of the binary information vector $\mathbf{b}$ and the binary estimated information vector $\hat{\mathbf{b}}$.
- WER vs VNR. Comparison of the binary information vector $\mathbf{b}$ and the binary estimated information vector $\hat{\mathbf{b}}$.

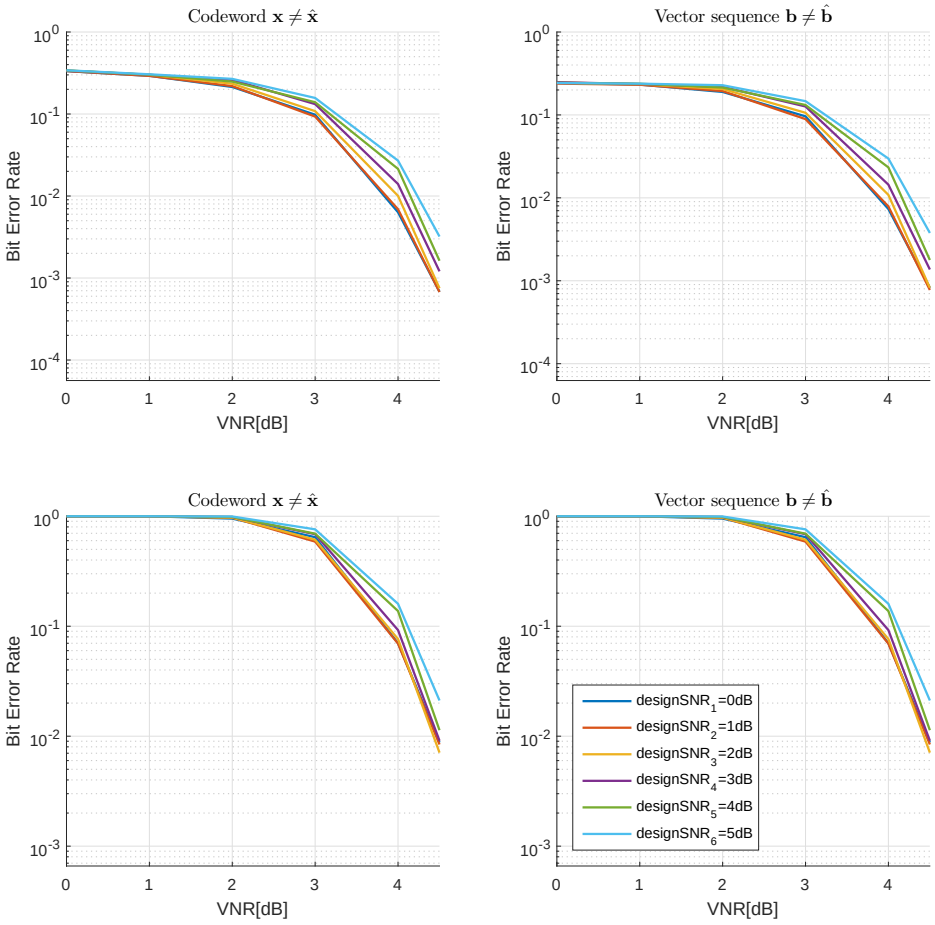

Figure 6.1. Various half rate polar codes $Polar(128, 64)$ of block length $N = 128$ are generated. Each polar code is constructed with different Bhattacharyya parameter according to a different SNR-design. SNR-design chosen arbitrary are 0dB, 1dB, 2dB, 3dB, 4dB and 5dB. The horizontal axes of the plot are E_b/N_0 in [dB]. N_0 is assume to be 2σ. The vertical axes are BER.

6.1 Polar Codes

The first simulation result shows the polar code constructed with a block code of $N = 128$. Figure B.1 shows the BER performance of a half rate polar code $Polar(128, 64)$. Each line on the plot is design with different SNR-design of 0dB, 1dB, 2dB, 3dB, 4dB and 5dB for a polar code. We observe in Figure B.1 that the best BER performance is obtained by an SNR-design of 0dB since it is the lowest error rate target (SNR−design=target−error).

6.1.1 Polar Codes under VNR

Figure B.2 also shows the plot of the BER performance of the $Polar(128, 64)$ polar code with half rate. However, this time the horizontal axes are in terms of VNR [dB]. This is calculated by assuming that the polar code generator matrix is a lattice generator matrix. The variance used for the nose is obtained by

$$\sigma^2 = V(\Lambda)10^{-(C_{ch}+VNR[db])/10} \tag{6.1}$$

where the capacity of the channel is $C_{ch} = 10\log_2(2\pi e)$. And the volume $V(\Lambda)$ is the volume of the polar code generator matrix obtained by equation (4.5).

We can observe that this transformation does not modify the performance of the polar itself. Now the objective is to improve the performance of a polar code by using a polar lattices.

In the next section polar lattices are constructed by a SNR-design of 0dB and we compare the results of the polar lattices with results of this section (polar codes constructed by SNR-design of 0dB).

6.2 Polar Lattice Construction A

In this sections, polar lattices simulation are done with an SNR-design of 0dB, N=128 and N=1024. Several polar lattices are constructed by changing the rate but the block length N is fixed with 128 or 1024 according to the simulation.

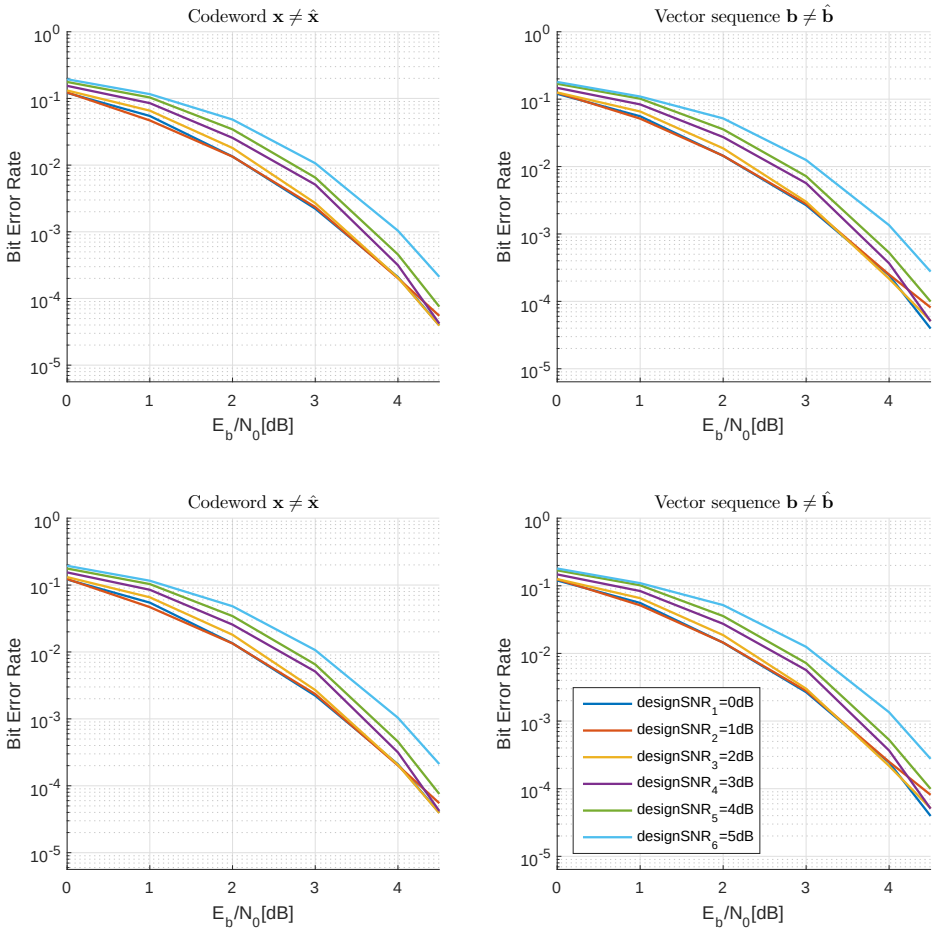

Figure 6.2. Bit error rate performance of polar codes under VNR. The horizontal axes of the plot are the VNR in [dB]. The vertical axes are BER. Various polar codes of block length is $N = 128$, and rate=$\frac{1}{2}$ $Polar(128, 64)$ are generated with different Bhattacharyya parameter according with the SNR-design. The SNR-designs are 0dB, 1dB, 2dB, 3dB, 4dB and 5db.

K	10	16	20	30	40	64	80	96	112	127
rate	0.08	0.13	0.16	0.23	0.31	0.5	0.63	0.75	0.88	0.99
best performance	8	9	7	6	5	3	2	best	4	10

Table 6.1. 128-dimensional lattices. Performance comparison of polar lattices constructed by selecting various polar subcode rates. $Kset = \{10, 16, 20, 30, 32, 40, 45, 64, 80, 96, 100, 112, 127\}$

6.2.1 128-Dimensional Polar Lattice $\Lambda_{P^{(1)}}$

By using construction A, we design several polar lattices $\Lambda_{P^{(1)}_{K_i}}$, with dimension $N = 128$. For each lattice several rates $(\frac{K}{N})$ are chosen to compare its SER performance. K_i is chosen arbitrary from the set of K information bits to be tested. $Kset = \{10, 16, 20, 30, 32, 40, 45, 64, 80, 96, 100, 112, 127\}$, i represent distinct polar lattices $\Lambda_{P^{(1)}_{K_i}}$ formed with the K_i information bit.

Figure B.3 shows the performance of the polar lattice $\Lambda_{P^{(1)}}[N, K_i]$. The plot on the left side shows the symbol error rate (SER) obtained by the comparison between the lattice point transmitted **x** and the estimated lattice point received $\hat{\mathbf{x}}$. While on the right side of Figure B.3, a plot of the BER is showed. This plot shows the comparison between the information sequence **b** and the estimated received sequence $\hat{\mathbf{b}}$.

We can observed that the best performance at VNR of 4dB is obtained with a rate of $\frac{96}{128}$. The four highest best performance are reached when $Kset = \{96, 80, 64, 100\}$. When performing encoding, it is expected that when the dimension K_i of the subcode is increased, the SER performance is shift to the higher VNR region, as it happens on any correcting code.

We observe that higher design-rate has better BER performance and SER performance.

6.2.2 Variation of the Polar Lattice $\Lambda_{P^{(1)}}$

A slightly different construction was also tested. Instead of simple doing a replacement of the linear independent vector of the code $\mathcal{C}$ into the identity matrix, see equation (5.2). We constructed a generator matrix G^{Λ_P} not only replacing the vectors, but also adding them.

$$G^{\Lambda_P} = \textbf{sum}\ \{\left[2I_N\right], \left[G^P\right]\} \text{from} \mathcal{F}^c \iff \mathcal{F}^c \in \mathcal{F}^c \tag{6.2}$$

We observe that some integers on the generator matrix G^{Λ_P} have value of 3 instead

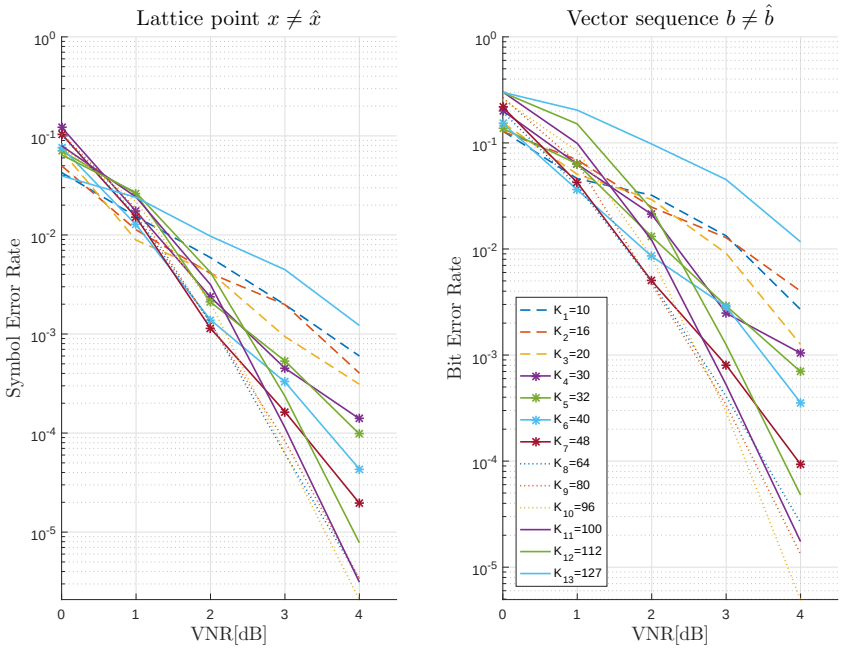

Figure 6.3. Symbol error rate performance (left plot) of polar codes under VNR, and Bit error rate performance (right plot). Polar Lattice are generated by construction A. 128-dimensional lattice. Various polar codes $\mathcal{C}$ are tested, each polar code is constructed by selecting K linear independent vectors. Polar lattices are constructed with $Kset = \{10, 16, 20, 30, 32, 40, 45, 64, 80, 96, 100, 112, 127\}$ linear independent vectors. The first polar code $\mathcal{C}$ has a total of $K_1 = 10$ linear independent vector, the second polar code has $K_2 = 16$ linear independent vector, etc.

of 2, as it happens using equation (5.2) that output values of 2. The performance improved only for the BER when comparing the transmitted information vector **b** and the estimated information vector $\hat{\mathbf{b}}$. However, the performance for the following three test perform is the same. 1) Performance of the SER for the lattice point **x** compared with the estimated codeword $\hat{\mathbf{x}}$, 2) WER between **x** and $\hat{\mathbf{x}}$, and 3) the WER between **b** and $\hat{\mathbf{b}}$. Shown on Figure B.4. An interesting observation was found in this test. The lattice with highest rate $K = 127$, improved by 3dB, a short hypothesis is given, however, further investigation is reserved for future work.

A higher noise is presented on lower VNR rates. In this work, errors are generated randomly with certain σ_{ch}, which is obtained by the volume of the lattice and the VNR used.

$$\sigma_{ch} = V(\Lambda)^{\frac{2}{N}} \cdot 10^{-\frac{(cap[dB]+VNR[db])}{10}} \tag{6.3}$$

And the fundamental volume of the lattice is given by

$$V(\Lambda) = |det(\Lambda)| \tag{6.4}$$

$V(\Lambda)$ can also be obtained by the expression given in [14]

$$V(\Lambda) = (2^{a-1})^N \cdot 2^{N-\sum_{l=1}^{a} k_l} \tag{6.5}$$

This expression has a relation with the information bits K. Higher the K's, smaller the volume, providing smaller noise. a is the number of subcodes used, in this case we are using only 1 subcode for construction A. Let us assume K_1=64, the volume of this lattice is $V_\Lambda = 1.8447 \times 10^{19}$, which gives a high σ_{ch}. On the other hand, selecting K_1=127, provides $V_\Lambda = 2$, leading to a small error rate.

6.2.3 $\Lambda_{P(1)}$ Known Frozen Bits on the Codeword

On the previous simulation, the information sequence **b** does not know which frozen bits were selected to form the polar lattice $\Lambda_{P(1)}$. For this subsection, et assume **b** knows which are the frozen bits and do not transmit information on those bits. After that encode **b** with G^{Λ_P}. Decode the noisy sequence with the construction A algorithm and compare the estimate sequence $\hat{\mathbf{b}}$ with the information sequence **b**. The results of the simulation are shown in Figure B.5.

With this plot we can observe that neither few or many independent row codevectors

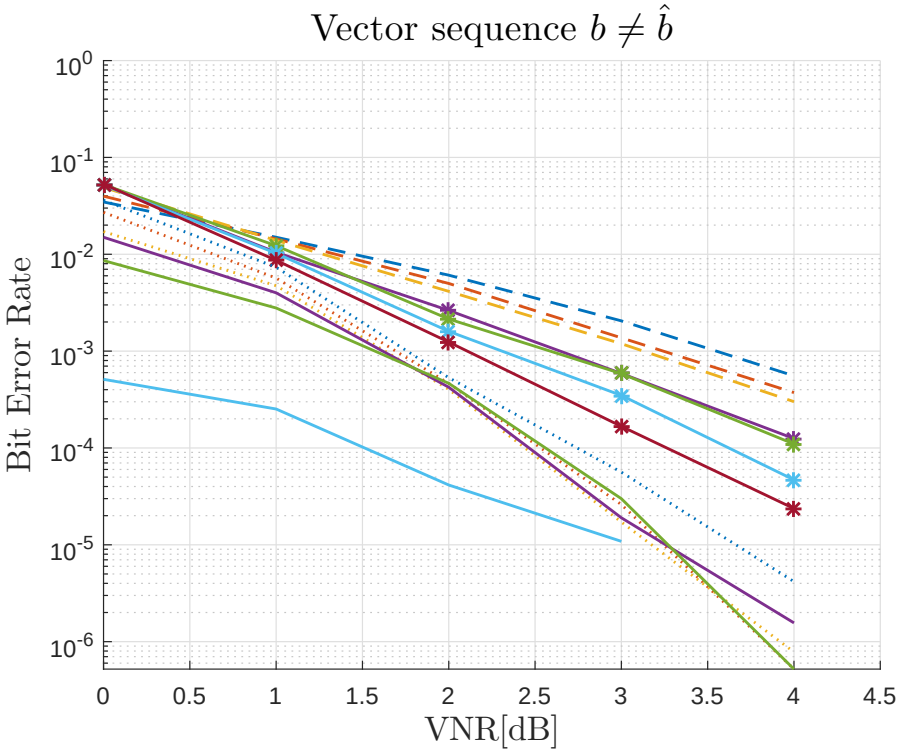

Figure 6.4. Bit error rate performance of a 128-dimensional polar lattice generated by construction A. Generator matrix of the polar lattice is obtained by adding the integer matrix with the polar generator matrix. Various polar codes $\mathcal{C}$ are tested, each polar code is constructed by selecting K linear independent vectors. Polar lattices are constructed with $Kset = \{10, 16, 20, 30, 32, 40, 45, 64, 80, 96, 100, 112, 127\}$ linear independent vectors.

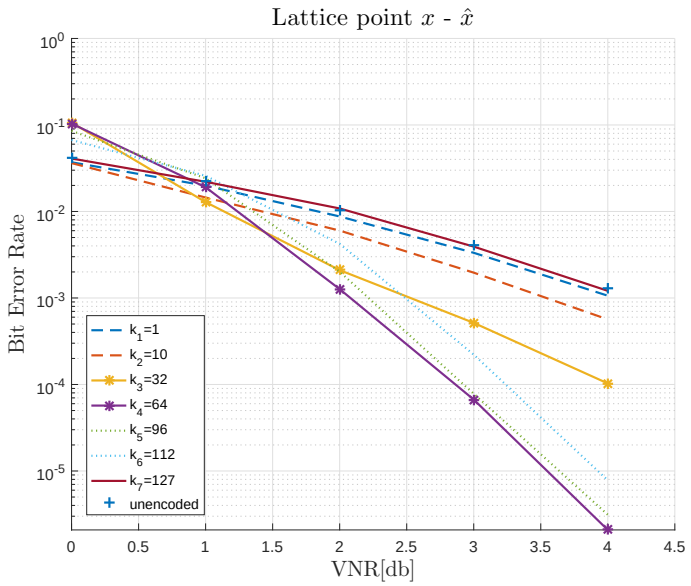

Figure 6.5. Polar Lattice generated by construction A. The information vector **b** knows the location of the frozen bits and it allocate information only in the non-frozen bits. N=128 dimensional lattice. Various polar codes C are tested, each polar code is constructed by selecting K linear independent vectors. The first polar code C has only one linear independent vector $K_1 = 1$, the second polar code has $K_2 = 10$ linear independent vector, ... the following lattices are constructed with $K = \{1, 10, 32, 64, 96, 112, 127\}$ linear independent vectors.

on the generator matrix outperform. This is expected since with few independent row codevectors means that we are encoding with only an identity matrix. On the other hand, with too many independent row vectors, we are encoding with only the polar code generator matrix, which does not provide any extra gain.

The best performance is obtained with the polar lattice $[N = 128, K = 64]$, and the second best is obtained with $[N = 128, K = 112]$.

Another similar simulation was performed. The information sequence encoder and decoder knows which are the frozen bits. We let the decoder of the polar lattice assing the frozen bits to the estimated sequence $\hat{\mathbf{b}}$. The simulation results show a similar performance as the Figure for SER vs VNR, and WER vs VNR for the comparison between **x** and $\hat{\mathbf{x}}$. And for BER vs VNR, and WER vs VNR for the comparison between **b** and $\hat{\mathbf{b}}$ the result outperform as expected.

6.2.4 Longer block code length, $N = 1024$ $\Lambda_{P(4)}$

A longer, $N = 1024$ dimensional lattices $\Lambda_{P(4)}$ are constructed with construction A. In this design we are increasing the block length. Figure B.6 shows the performance

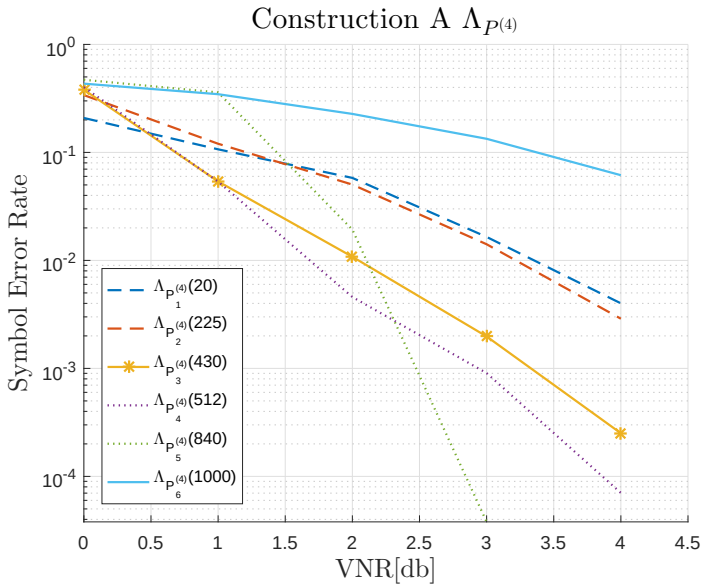

Figure 6.6. Polar lattice generated by construction A. N=1024 dimensional lattice. Polar lattices $\Lambda_{P^{(4)}}$ are generated with information bits $K = \{20, 225, 430, 512, 840, 1000\}$. Lattices are listed as follows $\Lambda_{P_1^{(4)}}(20)$, $\Lambda_{P_2^{(4)}}(225)$, $\Lambda_{P_3^{(4)}}(430)$, $\Lambda_{P_4^{(4)}}(512)$, $\Lambda_{P_5^{(4)}}(840)$, $\Lambda_{P_6^{(4)}}(1000)$.

of polar lattices of 1024-dimensional lattices. Lattices are constructed as $\Lambda_{P_1^{(4)}}(20)$, $\Lambda_{P_2^{(4)}}(225)$, $\Lambda_{P_3^{(4)}}(430)$, $\Lambda_{P_4^{(4)}}(512)$, $\Lambda_{P_5^{(4)}}(840)$, and $\Lambda_{P_6^{(4)}}(1000)$; where K was selected arbitrary. The best performance for the SER at $VNR = 3$ is given by $\Lambda_5^{(4)}$. This performance is measured between the lattice point $\mathbf{x}$ and the expected lattice point $\hat{\mathbf{x}}$.

6.3 Polar Lattice construction D

6.3.1 Single Level Polar Lattice $\Lambda_{P^{(3)}}$

Polar lattices were constructed by using construction D and only a subcode. It implies that we are forming another lattice similar as construction A. However, the decoding algorithm for construction D is employed. This design is plot on figure 6.8. No relevant interest in this design since the performance of construction A and construction D are the same as expected. However, it is interesting to mention that the highest K do not provide the best WER. The best WER is given by $\Lambda_{P_{10}^{(3)}}$ followed by $\Lambda_{P_{11}^{(3)}}$.

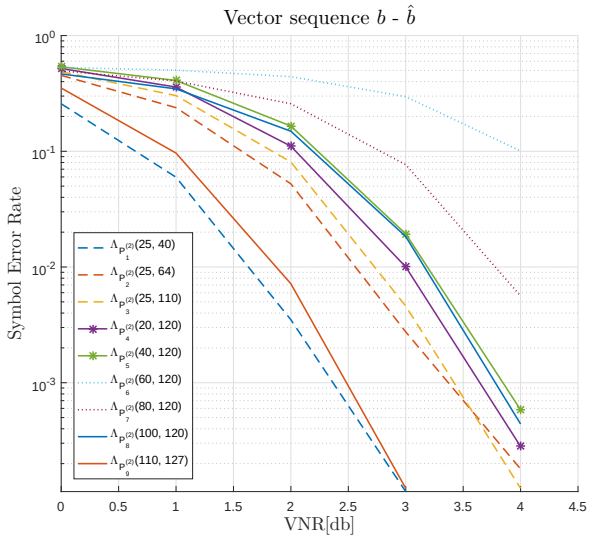

Figure 6.7. Polar Lattice generated by construction D. $N = 128$ dimensional lattice. Each polar lattice $\Lambda_{P^{(2)}}$ has two polar subcodes, each subcode is constructed by selecting k linear independent vectors. The first polar lattice $\Lambda_{P_1^{(2)}}(k_1, k_2)$ has C_1 with a total of $k_1 = 25$ linear independent vector and C_2 with $k_2 = 40$. The second polar lattice polar lattice $\Lambda_{P_2^{(2)}}(k_1, k_2)$ has $k_1 = 25$ and $k_2 = 65$ linear independent vector, the following lattices are constructed as $\Lambda_{P_3^{(2)}}(25, 110)$, $\Lambda_{P_4^{(2)}}(20, 120)$, $\Lambda_{P_5^{(2)}}(40, 120)$, $\Lambda_{P_6^{(2)}}(60, 120)$, and more than half rate for the first code are $\Lambda_{P_7^{(2)}}(80, 120)$, $\Lambda_{P_8^{(2)}}(100, 120)$, $\Lambda_{P_9^{(2)}}(110, 127)$.

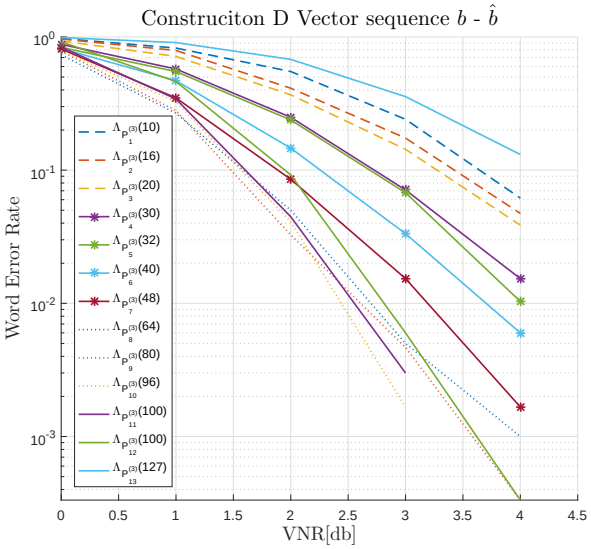

Figure 6.8. Single level Polar Lattice generated by construction D. $N = 128$ dimensional lattice. Polar lattice $\Lambda_{P^{(3)}}$ with only one polar subcode. $\Lambda_{P_1^{(3)}}(10)$,$\Lambda_{P_2^{(3)}}(16)$, $\Lambda_{P_3^{(3)}}(20)$, $\Lambda_{P_4^{(3)}}(30)$, $\Lambda_{P_5^{(3)}}(32)$, $\Lambda_{P_6^{(3)}}(40)$, $\Lambda_{P_7^{(3)}}(48)$, $\Lambda_{P_8^{(3)}}(64)$, $\Lambda_{P_9^{(3)}}(80)$, $\Lambda_{P_{10}^{(3)}}(96)$, $\Lambda_{P_{11}^{(3)}}(100)$, $\Lambda_{P_{12}^{(3)}}(100)$ and lastly $\Lambda_{P_{13}^{(3)}}(127)$.

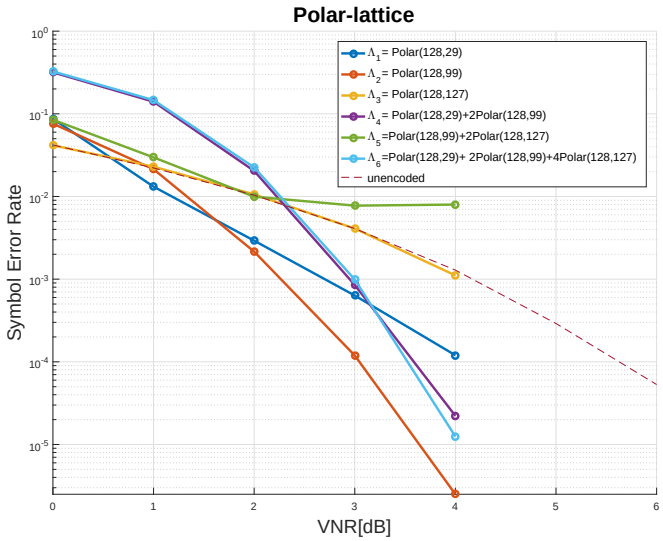

Figure 6.9. Performance of 6 different lattices with dimension 128: Three single-level lattices Λ_1, Λ_2 and Λ_3; two different two-level lattices Λ_4 and Λ_5; and finally a three-level lattice Λ_6. Λ_1 is a $PolarLattice(128, 29)$, Λ_2 is a $PolarLattice(128, 99)$, Λ_3 is a $PolarLattice(128, 127)$. Λ_4 is a two-level $PolarLattice(128, 99, 29)$ lattice, and Λ_5 is a two-level $PolarLattice(128, 127, 99)$ lattice. And Λ_6 is a three-level lattice defined by $PolarLattice(128, 127, 99, 29)$.

6.3.2 Two Level 128-Dimensional Polar lattice, $\Lambda_{P^{(2)}}$

Polar Lattices $\Lambda_{P^{(2)}}$ are constructed by using construction D. Since construction D requires nested codes, we selected polar codes by choosing linear independent codewords systematically by the equation (5.6).

As it was mention before, the indexing of frozen-bit provides the polar construction. And frozen-bits also provides the construction of polar lattice. For lattice generator matrix, frozen bits index are used to replaced rows with the same index of the identity matrix rows.

In this simulation we are interested to show the behavior of the SER performance when a pair of polar codes are chosen to form a polar lattice. In this construction, we do not take into consideration the minimum distance. Regardless it is not a formal polar lattice with construction D, we want to study the behavior of the rate pair chosen. Polar lattice codes are constructed with dimension $N = 128$, and a set of K values were taken arbitrary.

Figure 6.7 shows the performance of polar lattice $\Lambda_{P^{(2)}}$. For SER vs VNR of for the comparison between $\mathbf{b}$ and $\hat{\mathbf{b}}$. The best SER performance is obtained with $\Lambda_{P_1^{(2)}}(K_1 = 25, K_2 = 40)$ followed by $\Lambda_{P_9^{(2)}}(K_1 = 110, K_2 = 127)$.

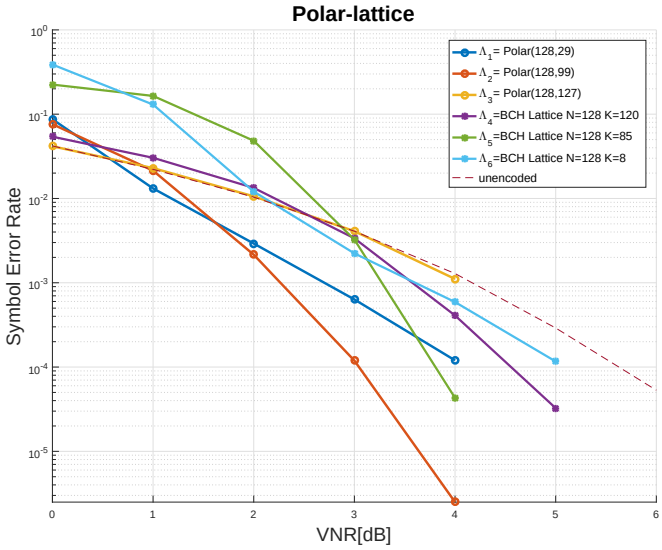

Figure 6.10. Performance of 6 different lattices with dimension 128: Three single-level lattices Λ_1, Λ_2 and Λ_3; three different single-level BCH-Lattices Λ_4, Λ_5 and Λ_6. The single-level lattice Λ_1 is a $PolarLattice(128, 29)$, Λ_2 is a $PolarLattice(128, 99)$, Λ_3 is a $PolarLattice(128, 127)$. The single-level BCH-Lattice Λ_4, Λ_5 and Λ_6 have information bits $K = \{120, 85, 8\}$ respectively.

Performance of the polar lattice $\Lambda_{P_9^{(2)}}$ was not expected to obtain such a low BER since $\Lambda_{P_{13}^{(1)}}$ got the worst SER performance with construction A.

On the other hand, $\Lambda_{P_1^{(2)}}$ and $\Lambda_{P_2^{(2)}}$ got even better performance than construction A, at VNR of 3dB, $\Lambda_{P_1^{(2)}}$ has SER of 10^{-4} and the best polar lattice $\Lambda_{P_{10}^{(2)}}$ constructed with construction A has SER of 0.29×10^{-4}.

6.4 Comparison performance for Multilevel Polar Lattice

We constructed 6 different lattices with dimension $N = 128$. Three single-level lattices Λ_1, Λ_2 and Λ_3; two different two-level lattices Λ_4 and Λ_5; and finally a three-level lattice Λ_6.

The single-level polar lattice Λ_1, Λ_2 and Λ_3 are formed by a polar codes $\mathcal{C}_3 = Polar(N = 128, K = 29)$, $\mathcal{C}_2 = Polar(N = 128, K = 99)$ and $\mathcal{C}_1 = Polar(N = 128, K = 127)$ code. The desired minimum distance for each code is ≥ 32, ≥ 8 and ≥ 2, respectively.

The two-level polar lattices Λ_4 and Λ_5 are formed as $\Lambda_4 = \mathcal{C}_3 + 2\mathcal{C}_2 + 2^2\mathbb{Z}^N$, $\Lambda_3 = \mathcal{C}_2 + 2\mathcal{C}_1 + 2^2\mathbb{Z}^N$.

The three-level polar lattice Λ_6 is formed as $\Lambda_6 = \mathcal{C}_3 + 2\mathcal{C}_2 + 4\mathcal{C}_1 + 8\mathbb{Z}^N$.

From simulation on Figure 6.9, we observed that a polar lattice with lower minimum distance outperform. These results are intuitive, because considering a fix lattice volume, the best performance happens when a higher amount of lattice points are packed on the same volume. On this simulation, we can also conclude that this specific single-level polar lattice perform better than a three-level polar lattice. Figure 6.10 also shows the comparison of single-level polar lattice codes with single-level BCH lattice codes. Simulation results show that polar lattice outperform BCH lattice for low VNR. It is interesting question to know if on higher VNR it has the same performance?. Looking at the $BCH - Lattice(128, 85)$, it seems that it tents to reduce its SER for higher VNR. This question is left for future work.

Chapter 7

Conclusions

In this work, we constructed polar lattices by two methods, construction A and construction D. Construction A is requires only a single binary code. Whereas, construction D requires nested binary codes. In this work we chose polar code as the binary code.

By simple comparing the VNR performance of the polar code and the lattice code, we can observed that the polar lattice outperform the polar code itself.

For construction A, we conclude that the best polar lattice is obtained by selecting a polar code closer to half rate. Simulation on section 6.2.1 shows that the best rate is $\frac{96}{128}$ assuming that we transmit N-bits of information. However, section 6.2.3 shows that the best rate for the polar codes is $\frac{64}{128}$, when it is assume that K-bits of information are transmitted.

For construction D, we conclude that the selection of multilevel do not improve the SER performance for the polar lattice. A single-level lattice outperform on the SER for the same N-dimensional lattice. It is also conclude that lattices with multilevel perform better in terms of SER when the minimum distance is lower, we expect this result since considering a fix lattice volume, the best performance happens when a higher amount of lattice points are packed on the same volume.

And finally we conclude that polar lattices perform better on the lower VNR for the SER than BCH-Lattices.

Appendix A

Bit Reversal Permutation Matrix $\mathbf{B}_N$

For matrices $A = \{a_{i,j}\}$ and $B = \{b_{i,j}\}$, $A \otimes B$ represent the Kronecker product defined by

$$A \otimes B \doteq \begin{bmatrix} a_{1,1}B & a_{1,2}B & \cdots \\ a_{2,1}B & a_{2,2}B & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix} \tag{A.1}$$

And the Kronecker powers are defined as $A^{\otimes n} \triangleq A \otimes A^{\otimes n-1} = A^{\otimes n-1} \otimes A$.

The reverse shuffle permutation matrix is obtained by the matrix " $\mod -p$ perfect shuffle" $S^T_{2,N/2}$ defined by

$$S^T_{2,\frac{N}{2}}(s_1, s_2, \ldots, s_N)^T = (s_1, s_3, \ldots, s_{N-1}, s_2, s_4, \ldots, s_N)^T \tag{A.2}$$

where $R_N = S^T_{2,\frac{N}{2}}$ denotes the $N \times N$ revere shuffle permutation matrix defined by

$$(s_1, s_2 x \ldots, s_N) R_N = (s_1, s_3, \ldots, s_{N-1}, s_2, s_4, \ldots, s_N)^T \tag{A.3}$$

R_N reorders the vector in the way that the index experiences a left circular shift. It leads that the least significant bit of the index now become the most significant bit of the index.

The N by N bit reversal permutation matrix B_N can be defined recursively in terms of R_N given by

$$B_N = R_N \begin{bmatrix} B_{\frac{N}{2}} & 0 \\ 0 & B_{\frac{N}{2}} \end{bmatrix} \tag{A.4}$$

$$B_N = R_N(I_2 \otimes B_{\frac{N}{2}}) \tag{A.5}$$

This holds because applying R_N reorders the vector so that the least significant bit of the index becomes the most significant bit of the index. Multiplying by $I_2 \otimes B_{\frac{N}{2}}$ separately reorders the first and last halves of the vector according to bit reversal of length $\frac{N}{2}$.

Appendix B

Plot results

B.1 Polar Codes

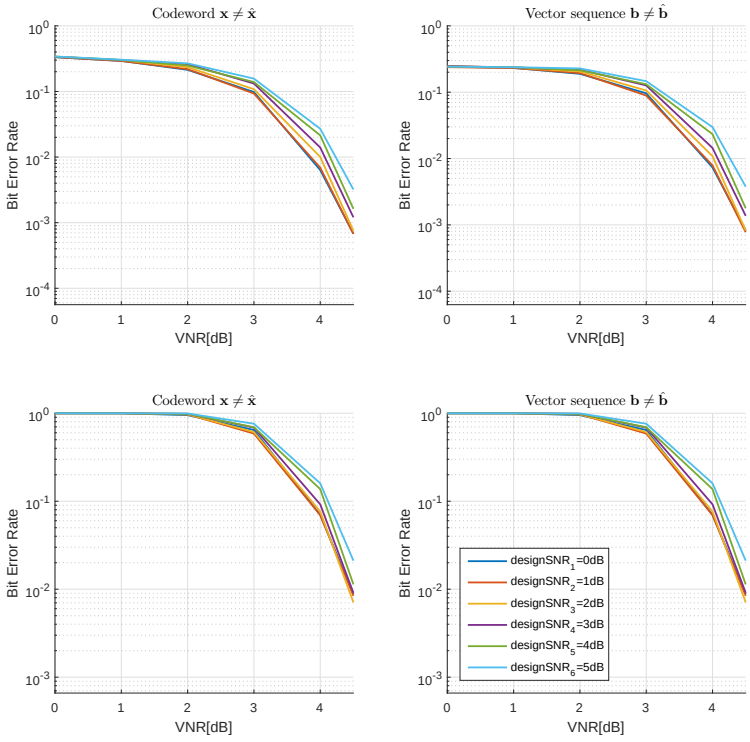

Figure B.1. Various half rate polar codes $Polar(128, 64)$ of block length $N = 128$ are generated. Each polar code is constructed with different Bhattacharyya parameter according to a different SNR-design. SNR-design chosen arbitrary are 0dB, 1dB, 2dB, 3dB, 4dB and 5dB. The horizontal axes of the plot are E_b/N_0 in [dB]. N_0 is assume to be 2σ. The vertical axes are BER.

B.1.1 Polar Codes under VNR

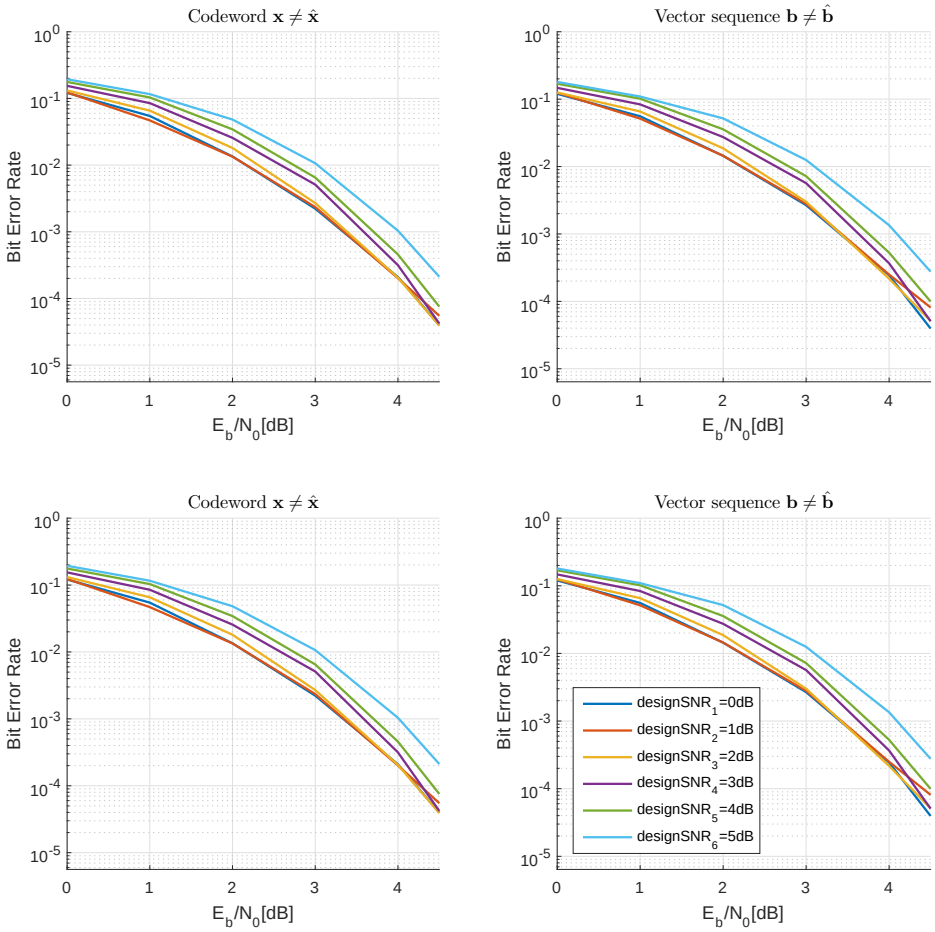

Figure B.2. Bit error rate performance of polar codes under VNR. The horizontal axes of the plot are the VNR in [dB]. The vertical axes are BER. Various polar codes of block length is $N = 128$, and rate=$\frac{1}{2}$ $Polar(128, 64)$ are generated with different Bhattacharyya parameter according with the SNR-design. The SNR-designs are 0dB, 1dB, 2dB, 3dB, 4dB and 5db.

B.1.2 128-Dimensional Polar Lattice $\Lambda_{P(1)}$

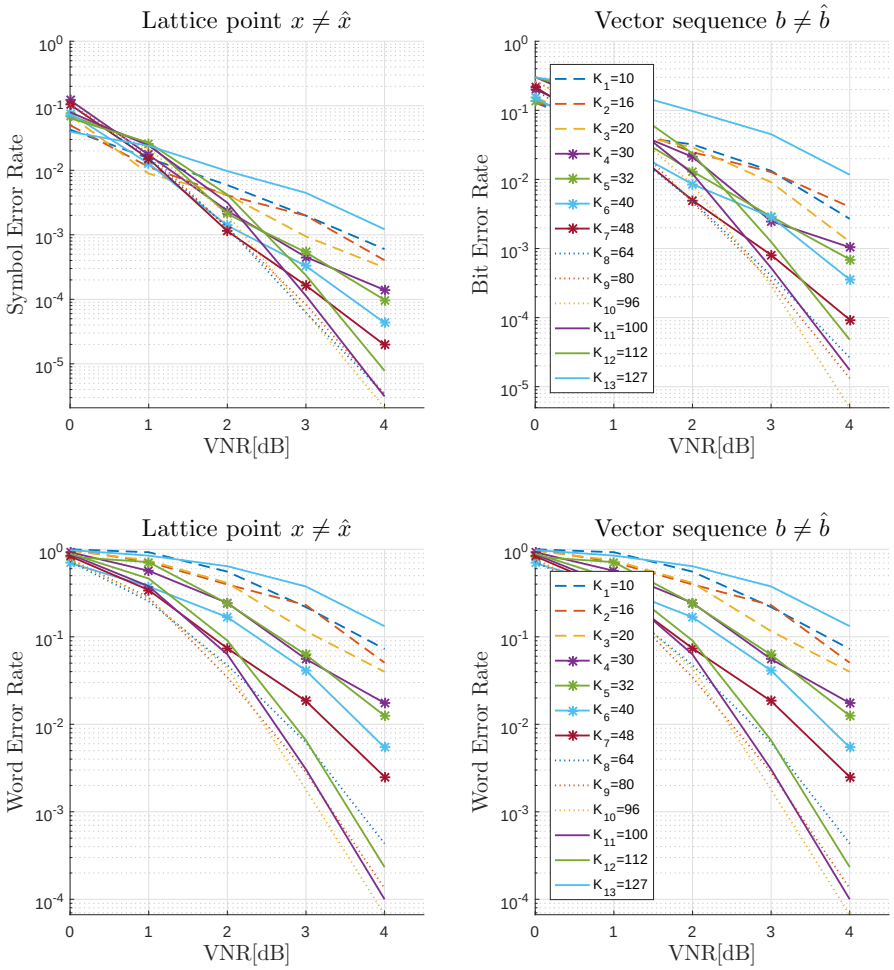

Figure B.3. Symbol error rate performance (left plot) of polar codes under VNR, and Bit error rate performance (right plot). Polar Lattice are generated by construction A. 128-dimensional lattice. Various polar codes $\mathcal{C}$ are tested, each polar code is constructed by selecting K linear independent vectors. Polar lattices are constructed with $Kset = \{10, 16, 20, 30, 32, 40, 45, 64, 80, 96, 100, 112, 127\}$ linear independent vectors. The first polar code $\mathcal{C}$ has a total of $K_1 = 10$ linear independent vector, the second polar code has $K_2 = 16$ linear independent vector, etc.

B.1.3 Variation of the Polar Lattice $\Lambda_{P^{(1)}}$

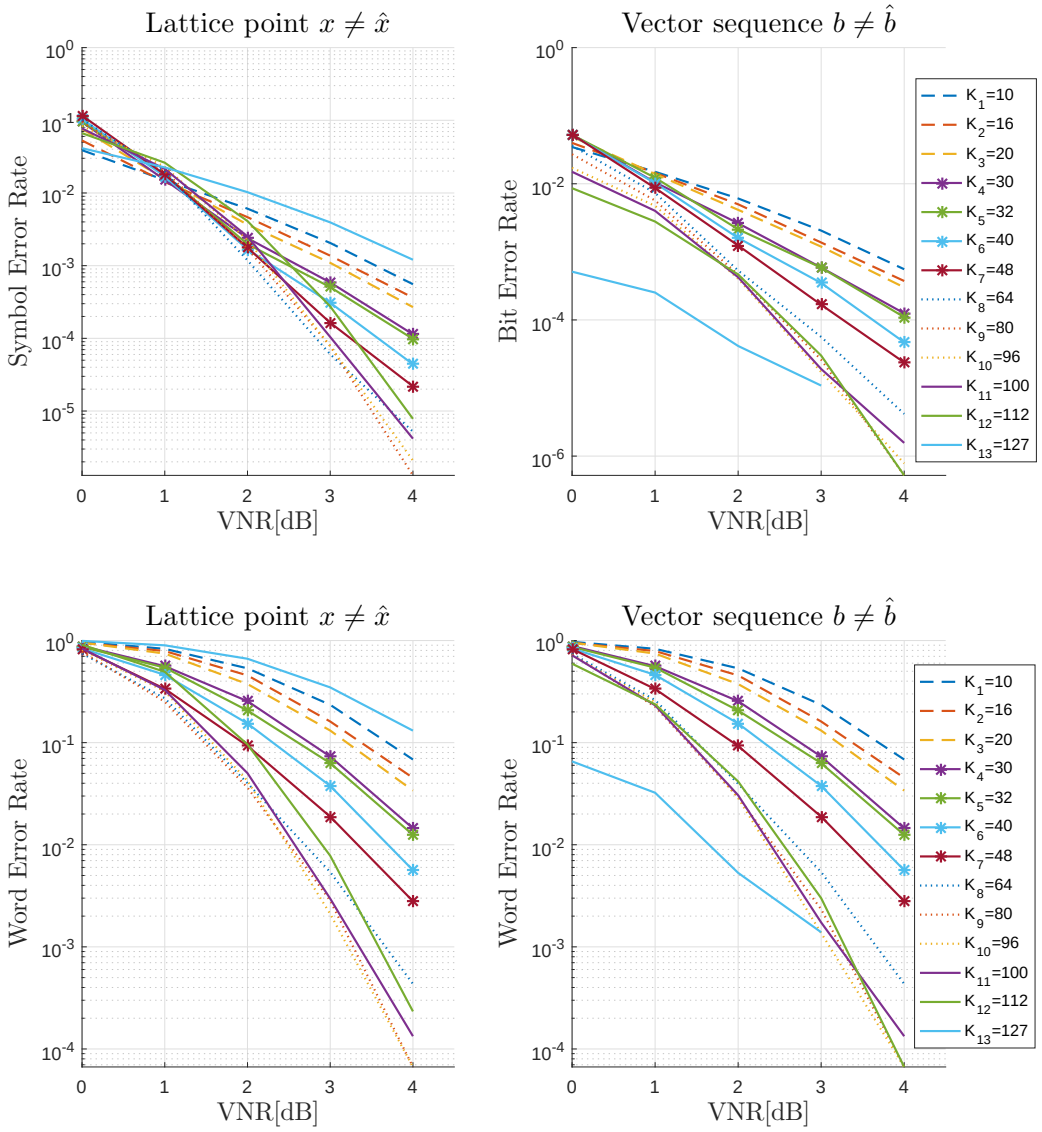

Figure B.4. Bit error rate performance of a 128-dimensional polar lattice generated by construction A. Generator matrix of the polar lattice is obtained by adding the integer matrix with the polar generator matrix. Various polar codes $\mathcal{C}$ are tested, each polar code is constructed by selecting K linear independent vectors. Polar lattices are constructed with $Kset = \{10, 16, 20, 30, 32, 40, 45, 64, 80, 96, 100, 112, 127\}$ linear independent vectors.

B.1.4 $\Lambda_{P(1)}$ Known Frozen Bits on the Codeword

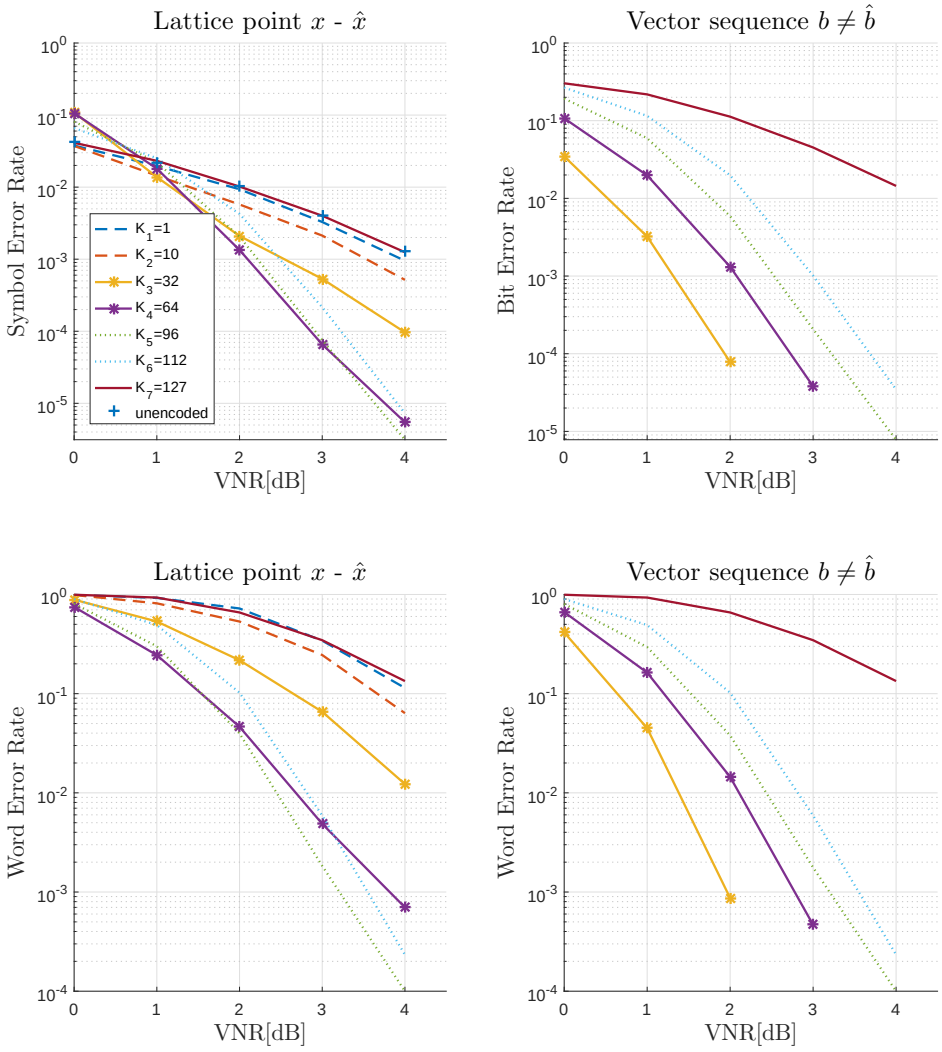

Figure B.5. Polar Lattice generated by construction A. The information vector **b** knows the location of the frozen bits and it allocate information only in the non-frozen bits. N=128 dimensional lattice. Various polar codes C are tested, each polar code is constructed by selecting K linear independent vectors. The first polar code C has only one linear independent vector $K_1 = 1$, the second polar code has $K_2 = 10$ linear independent vector, ... the following lattices are constructed with $K = \{1, 10, 32, 64, 96, 112, 127\}$ linear independent vectors.

B.1.5 Longer block code length, $N = 1024$ $\Lambda_{P^{(4)}}$

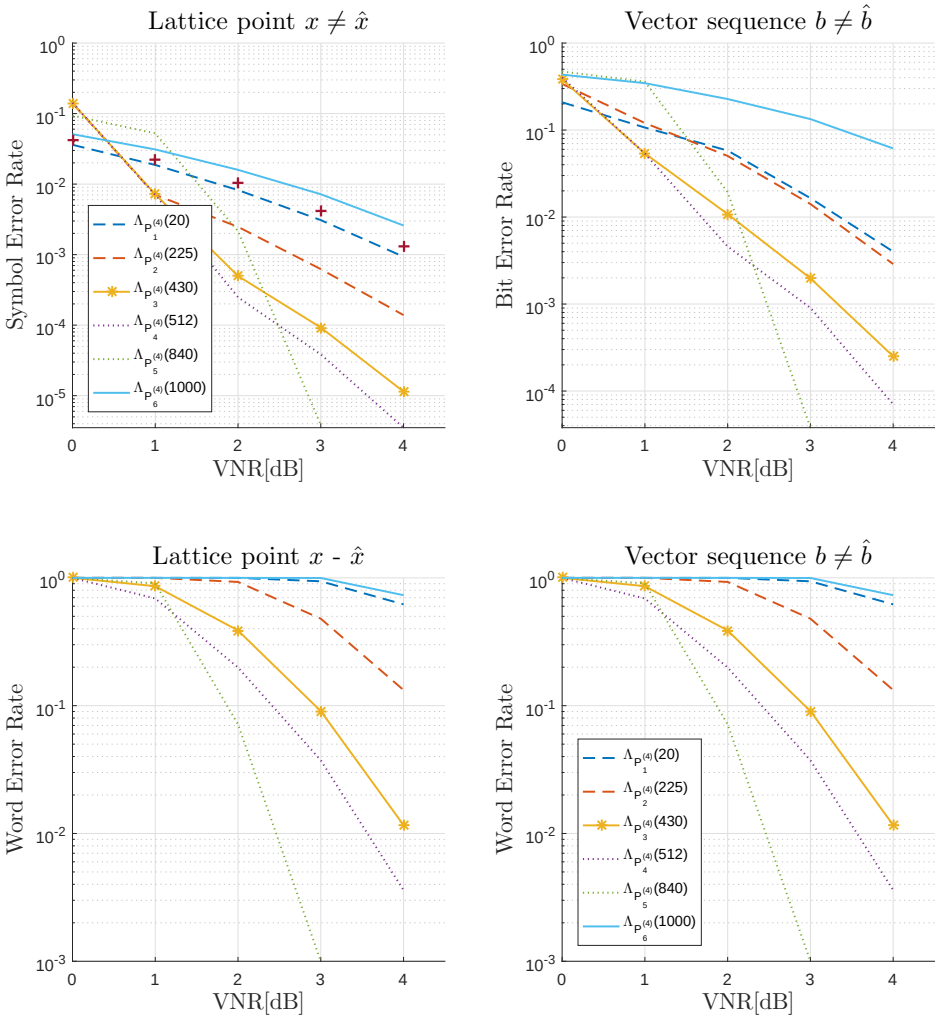

Figure B.6. Polar lattice generated by construction A. N=1024 dimensional lattice. Polar lattices $\Lambda_{P^{(4)}}$ are generated with information bits $K = \{20, 225, 430, 512, 840, 1000\}$. Lattices are listed as follows $\Lambda_{P_1^{(4)}}(20)$, $\Lambda_{P_2^{(4)}}(225)$, $\Lambda_{P_3^{(4)}}(430)$, $\Lambda_{P_4^{(4)}}(512)$, $\Lambda_{P_5^{(4)}}(840)$, $\Lambda_{P_6^{(4)}}(1000)$.

Bibliography

[1] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near shannon limit error – correcting coding and decoding: Turbo-codes," *Technical Program, Conference Record, IEEE International Conference on Communications, ICC '93 Geneva*, pp. 1064–1070, May 1993.

[2] J. H. Conway and N. J. A. Sloane, *Sphere Packings, Lattices and Groups*. New York, NY, USA: Springer-Verlag, 2nd ed., 1993. ISBN 0-387-97912-3.

[3] E. Arikan, "Performance comparison of polar codes and reed-muller codes," *IEEE Comm. Lett.*, vol. 12, pp. 447 – 449, June 2008.

[4] G. Sarkis, P. Giard, A. Vardy, C. Thibeault, and W. J. Gross, "Fast polar decoders: Algorithm and implementation," *IEEE Journal on Selected Areas in Communications*, vol. 32, pp. 946–957, May 2014.

[5] C. Leroux, , A. J. Raymond, G. Sarkis, I. Tal, A. Vardy, and W. J. Gross, "Hardware implementation of successive-cancellation decoders for polar codes," *Journal of Signal Processing Systems*, vol. 69, no. 3, pp. 305–315, 2012.

[6] I. Tal and A. Vardy, "How to construct polar codes," *IEEE Trans. on Info. Theory*, vol. 59, pp. 6562–6582, Oct 2013.

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

[8] H. Li and J. Yuan, "A practical construction method for polar codes in awgn channels," *IEEE 2013 Tencon – Spring*, pp. 223–226, 2013.

[9] M. Andersson, V. Rathi, R. Thobaben, J. Kliewer, and M. Skoglund, "Nested polar codes for wiretap and relay channels," *IEEE Communications Letters*, vol. 14, pp. 752–754, August 2010.

[10] H. Vangala, E. Viterbo, and Y. Hong, "Permuted successive cancellation decoder for polar codes," *International Symposium on Information Theory and its Applications*, pp. 438–442, October 2014.

[11] E. S. Barnes and N. J. A. Sloane, "New lattice packings of spheres," *Canad. J. Math*, vol. 35, no. 1, pp. 117–130, 1983.

[12] G. D. Forney, M. D. Trott, and S.-Y. Chung, "Sphere-bound-achieving coset codes and multilevel coset codes," *IEEE Transactions on Information Theory*, vol. 46, pp. 820–850, May 2000.

[13] J. Pujol, J. Rif, and L. Ronquillo, "Construction of additive reed-muller codes ," 2011.

[14] Y. Yan and C. Ling, "A construction of lattices from polar codes," *IEEE Information Theory Workshop*, pp. 124–128, 2012.