Pontifícia Universidade Católica
do Rio de Janeiro

Robert Mota Oliveira

Polarization-Driven Puncturing for Polar Codes in 5G Systems

Dissertação de Mestrado

Dissertation presented to the Programa de Pós–Graduação em Engenharia Elétrica of PUC-Rio in partial fulfillment of the requirements for the degree of Mestre em Engenharia Elétrica.

Advisor: Prof. Rodrigo Caiado de Lamare

Rio de Janeiro
March 2018

Pontifícia Universidade Católica
do Rio de Janeiro

Robert Mota Oliveira

Polarization-Driven Puncturing for Polar Codes in 5G Systems

Dissertation presented to the Programa de Pós–Graduação em Engenharia Elétrica of PUC-Rio in partial fulfillment of the requirements for the degree of Mestre em Engenharia Elétrica. Approved by the undersigned Examination Committee.

Prof. Rodrigo Caiado de Lamare
Advisor
CETUC – PUC-Rio

Prof. Marco Antonio Grivet Mattoso Maia
CETUC – PUC-Rio

Prof. Lukas Tobias Nepomuk Landau
CETUC – PUC-Rio

Prof. João Terêncio Dias
CEFET/RJ

Prof. Márcio da Silveira Carvalho
Vice Dean of Graduate Studies
Centro Técnico Científico – PUC-Rio

Rio de Janeiro, March the 28th, 2018

Robert Mota Oliveira

The author graduated in Eletrical Engineering from the Universidade Federal do Espírito Santo, in Vitória, Espírito Santo, Brasil, 1999.

Bibliographic data

Oliveira, Robert Mota

Polarization-Driven Puncturing for Polar Codes in 5G Systems / Robert Mota Oliveira; advisor: Rodrigo Caiado de Lamare. – Rio de Janeiro: PUC-Rio, Departamento de Engenharia Elétrica, 2018.

v., 60 f: il. color. ; 30 cm

Dissertação (mestrado) - Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Engenharia Elétrica.

Inclui bibliografia

1. Engenharia Elétrica – Teses. 2. Engenharia Elétrica – Teses. 3. Códigos Polares;. 4. 5G;. 5. Puncionamento;. 6. Polarização de Canal;. I. Lamare, Rodrigo Caiado de. II. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Engenharia Elétrica. III. Título.

CDD: 621.3

To Eduarda and Fabrini.

Acknowledgments

I would like to express my sincere gratitude to my supervisor Dr. Rodrigo de Lamare for his supervision, support, guidance and patience throughout my research studies, without which much of this work would not have been possible.

Further thanks to all the professors that have contributed to my academic and professional career, especially to the professors of PUC-Rio.

My sincere thanks to the CAPES and PUC-Rio, for the financial support.

I also want to say thanks to my colleagues and friends of CETUC for their support and help.

My family has supported and encouraged me throughout my long years of study and for that, and so much more, they have my gratitude and love.

My daughter Eduarda, who patiently waited for my long periods of absence.

Finally, I would like to thank my wife Fabrini, who inspired me and cared for me, provided companionship and encouragement, fun and help, all when they were needed most.

Abstract

Oliveira, Robert Mota; Lamare, Rodrigo Caiado de (Advisor). **Polarization-Driven Puncturing for Polar Codes in 5G Systems**. Rio de Janeiro, 2018. 60p. Dissertação de Mestrado – Departamento de Engenharia Elétrica, Pontifícia Universidade Católica do Rio de Janeiro.

This thesis presents a polarization-driven puncturing technique for the design of punctured polar codes. The proposed puncturing strategy consists of reducing the generator matrix by relating its row index based on the channel polarization principle. The punctured codes constructed based on channel polarization are then considered with successive cancellation (SC) decoding and punctured bits known to both the encoder and the decoder. The Spectrum Distance (SD) and the Joint Spectrum Distance (JSD) are then used to performance analysis. Simulation results show that the proposed punctured polar codes outperform existing punctured polar codes.

Keywords

Polar Codes; 5G; Puncturing; Channel Polarization;

Resumo

Oliveira, Robert Mota; Lamare, Rodrigo Caiado de. **Puncionamento orientado por polarização para Códigos Polares em sistemas 5G**. Rio de Janeiro, 2018. 60p. Dissertação de Mestrado – Departamento de Engenharia Elétrica, Pontifícia Universidade Católica do Rio de Janeiro.

Esta dissertação apresenta uma técnica de puncionamento orientada pela polarização para o projeto de códigos polares puncionados. A estratégia de puncionamento proposta consiste em reduzir a matriz geradora relacionando seu índice de linha com o princípio da polarização do canal. Os códigos puncionados construídos com base na polarização do canal são então considerados para a decodificação por cancelamento sucessivos (SC) com os bits perfurados conhecidos tanto no codificador como no decodificador. A Distância de Espectro (SD) e a Distância de Espectro Conjunta (JSD) são então utilizadas para análise de desempenho. Os resultados das simulações mostram que os códigos polares puncionados propostos superam os códigos polares puncionados existentes.

Palavras-chave

Códigos Polares; 5G; Puncionamento; Polarização de Canal;

Table of contents

1 Introduction **13**
1.1 Motivation and Problems 14
1.2 Contributions 15
1.3 Dissertation Outline 15
1.4 List of Publications 16

2 Polar Codes **17**
2.1 Introduction 17
2.2 System Model 17
2.3 Channel Polarization 19
2.3.1 Channel Combining 19
2.3.2 Channel Splitting 21
2.3.3 Code Construction 22
2.4 Encoding of Polar Codes 23
2.4.1 Systematic Encoding 26
2.5 Successive Cancellation Decoding 28
2.6 Successive Cancellation List Decoding 32
2.7 CRC-Aided Successive Cancellation List Decoding 36
2.8 Puncturing and extension 37
2.9 5G scenarios 39

3 Proposed Polarization-Driven Techniques **43**
3.1 Introduction 43
3.2 Mathematical description 43
3.3 Pseudo-code 46
3.4 Analysis 47
3.5 Simulation 50

4 Conclusions **55**

Bibliography **56**

List of figures

Figure 2.1 System Model. 17
Figure 2.2 Construction of W_2 from W. 20
Figure 2.3 Construction of W_4 from W_2. 20
Figure 2.4 Channel Polarization 23
Figure 2.5 Code construction for BEC with n=8, k=4 and ϵ=0.3. 24
Figure 2.6 Encoding procedure for an (8,4) polar code, BEC with $\epsilon = 0.3$ 25
Figure 2.7 Systematic encoding of a (8,4) polar code. 27
Figure 2.8 Comparison of systematic and non-systematic polar codes. 28
Figure 2.9 Structure of the polar decoding graph for $n = 8$ 32
Figure 2.10 Successive cancellation (SC) decoding for the polar code with $n = 4$ 33
Figure 2.11 Evolution of decoding paths for $\mathcal{L} = 4$. 35
Figure 2.12 Comparison of list decoder for different list sizes. 36
Figure 2.13 Comparison of list decoder with different CRC. 38
Figure 2.14 Puncture criterions for Matrix $\mathbf{G}$. 39
Figure 2.15 Graph represents for encoder and decoder 40
Figure 2.16 Punctured Graph for the encoder and the decoder. 42

Figure 3.1 The relation between the polarized channels W_i and the rows of the generator matrix $\mathbf{G}$. 48
Figure 3.2 Polar Spectra SDC for various puncturing techniques. 49
Figure 3.3 Percentage of coinciding punctured entries of puncturing techniques. 51
Figure 3.4 Comparative BER performace of puncturing polar code n'=100 k=64. 52
Figure 3.5 Comparative BER performace of puncturing polar code n'=480 k=256. 53
Figure 3.6 Comparative BER performace of puncturing polar code n'=1920 k=1600. 54

List of tables

Table 2.1 Operations performed by a SC decoder with $n = 8$. 31
Table 2.2 5G requirements 41

Table 3.1 Polarization vector $\mathbf{b}$ for $n = 8$ 46
Table 3.2 Polar Spectra for Figures 3.4, 3.5 and 3.6 53

List of Abreviations

5G	5th Generation
AWGN	Additive White Gaussian Noise
B-DMC	Binary Discrete Memoryless Channel
BEC	Binary Erasure Channel
BER	Bit Error Rate
BP	Belief Propagation
BPSK	Binary Phase Shift Keying
BSC	Binary Symmetric Channel
CA-SCL	CRC-Aided Successive Cancellation List
CHW	Complement Hamming Weight
CRC	Cyclic Redundancy Check
eMBB	Extended Mobile Broadband
FEC	Forward Error Correction
FER	Frame Error Rate
GF(2)	Binary Galois Field
HW	Hamming Weight
JSD	Joint Spectrum Distance
LDPC	Low Density Parity Check
LLR	Log-Likelihood Ratio
ML	Maximum Likelihood
PD	Polarization-Driven
PS	Polar Spectra
RQUP	Reversal Quasi-Uniform Puncturing
SC	Successive Cancellation
SCL	Successive Cancellation List
SD	Spectrum Distance
SDC	Spectrum Distance for Complementary Path Weight
SDP	Spectrum Distance for Path Weight
SNR	Signal To Noise Ratio

Divide the difficulties you encounter into as many parts as possible for your best solution.

René Descartes, *Discours de la méthode pour bien conduire sa raison, et chercher la verité dans les sciences.*

1
Introduction

Polar codes are a type of linear block codes proposed by Arikan [1] that are considered to be one of the most important breakthroughs in coding theory in the past decade. Polar codes asymptotically achieve capacity on a wide array of channels using an encoder and a decoder of complexity $\mathcal{O}(n \log n)$. They also have the desirable property that the same decoder can be used to decode all polar codes of different rates and designs. Because of the low complexity, there are many studies on decoder enhancement and implementation in hardware.

Polar codes achieve capacity for asymptotically large block lengths. For finite block lengths, however, the performance of polar codes is worse than the state-of-the-art Low Density Parity Check (LDPC) codes [2] [39] [40] [41], and the originally proposed successive cancellation decoder is not optimal. In the past few years, construction as well as decoding algorithms of polar codes have improved significantly. Tal and Vardy [2] showed that at moderate block lengths, polar codes outperform state-of-the-art LDPC codes in terms of error rate on the Additive White Gaussian Noise (AWGN) channel when concatenated even with the simplest cyclic redundancy check (CRC) codes using a successive-cancellation list decoder [2]. The work of Tal and Vardy produced a powerful code with near-optimal performance and a relatively low-complexity decoder.

Polar codes are very different from other modern coding schemes. The code construction is rather unique in that designed codes are dependent on the underlying channel, where the code construction can change with the Signal-to-Noise-Ratio (SNR) of AWGN channel. The puncturing or extension of polar codes is also non-trivial compared to other coding schemes because the codes are generated with lengths that are powers of two and this results in more challenging puncturing and extension requirements. Since the polar codes are dependent on the underlying channel, the fading channel performance is thus important to consider.

1.1 Motivation and Problems

The development of the 5th generation (5G) wireless communications systems and other wireless standards is currently at an accelerated pace and is motivated by the exponential increase in data transmission in today's networks [3] [4] [47] [49]. The different uses of future wireless services will be increasingly demanding, whether it is vehicle communication or low-power sensor networks.

In addition, 5G systems have challenging requirements regarding required [48] transmission rates, power consumption, coverage, and latency in data transmission. Among the central elements of 5G systems are the waveform used in the transmission and the channel coding technique adopted to ensure the required performance on the link. Recently, polar code have been adopted for the control channels of the extended Mobile Broadband (eMBB) scenario of 5G systems [50].

Polar coding has the potential to meet all these requirements if the coding scheme applied is well designed. The main obstacle to implementation is the need for an adequate puncturing and/or extension technique to meet the length of the proposed frame structure in 5G systems.

In their original construction, polar codes only allowed code lengths that are powers of two, while the message length (the number of information bits) is arbitrary. This suggests that codeword length of standard polar codes will be 64, 128, 256, 512, 1024, etc. This restriction on the code lengths is a major drawback of polar codes, which needs to be overcome for practical applications. Recent works have proposed ways to modify polar codes in order to achieve any code length, using the techniques of puncturing from coding theory [5]. For 5G systems, the proposed frame structure for several use cases will likely consider codewords with lengths varying from around 100 bits for machine-type communications (MTC), 480 bits for Internet-of-Things (IoT) and 1920 for eMBB.

Puncturing is the technique of reducing the original length of a codeword. In our case we will present a technique that reduces the generator matrix of the code. In this dissertation we propose a puncturing technique for polar codes with better performance as compared to those reported in the literature. Based on channel polarization, the proposed technique is more suitable for decoding by successive cancellation and has promising results when compared to the techniques proposed by Niu [5] and Wang [6], which are among the most effective puncturing techniques in the literature.

1.2 Contributions

The contributions presented in this work are the development of a puncturing technique based on channel polarization, namely the polarization-driven (PD) puncturing technique and the study of polar codes and their decoding algorithms for 5G wireless communication systems. This technique is suitable for decoding by successive cancellation or other approaches, and is applied in the reduction of the generator matrix. The coding and decoding procedures are the same as those applied in the original polar codes. In particular, we describe the design of rate-compatible polar codes using the proposed PD method and their application to 5G scenarios. We also carry out an analysis of the proposed method using the Spectrum Distance (SD) and the Joint Spectrum Distance (JSD) [5] as a performance metric used to compare different puncturing techniques. In addition, we present the technique of cutting the coding/decoding graph as a suitable solution for implementation of encoding and decoding algorithms [7].

For this, was constructed the polar code for the AGWN channel, and was implemented a non-recursive version of the list decoder. Which allows the implantation of puncturing techniques in list decoder, since the decoding algorithm proposed by Tal [15] is recursive and higher complexity.

1.3 Dissertation Outline

The rest of this dissertation is organized as follows:

• Chapter 2, a review of the literature is provided, covering the fundamentals of polarization, polar codes, puncturing methods and a number of key concepts which will be used throughout this dissertation. The general coding system including channel models considered are introduced and the specific features of polar codes, including the properties of the encoder and the decoder. We also introduce the basic decoder called Successive Cancellation decoder (SC), the Successive Cancellation List decoder (SCL) and the CRC-Aided Successive Cancellation List (CA-SCL) decoder. Furthermore, the 5G scenarios are described with a focus on block size and information rates for each application.

• Chapter 3 presents the polarization-driven (PD) puncturing technique and relevant material description, pseudo-code, analysis, complexity and simulations. We describe in detail the Spectrum Distance (SD) and the Joint Spectrum Distance (JSD) methods as well as methods of performance analysis for codes. Moreover, a description of the technique for cutting the cod-

ing/decoding graph and its relation with the generator matrix and polarized channels is provided.

- Chapter 4 summarizes the contributions of the dissertation, provides conclusions for each of the topics considered and discusses related open problems with potential for future work.

1.4 List of Publications

Some of the results in this dissertation have been published, or will be submitted to publications.

Conference Papers:

– R.M. Oliveira e R.C. de Lamare, "Códigos Polares e Puncionamento Baseado em Polarização para Sistemas 5G", XXXV Simpósio Brasileiro de Telecomunicações e Processamento de Sinais (SBrT2017), São Pedro-SP, 2017.

Journal Papers:

– C. Healy, Z. Shao, R.M. Oliveira, R.C. de Lamare, L. Mendes, "Knowledge-Aided Informed Dynamic Scheduling for LDPC Decoding of Short Blocks", IET Communications, 2018.

– R.M. Oliveira e R.C. de Lamare, "Rate-Compartible Polar Codes Based on Polarization-Driven Puncturing" (Under Preparation).

2
Polar Codes

2.1
Introduction

This chapter provides a review of polar codes and their encoding and decoding methods. It begins with a description of the channel polarization phenomenon followed by a review of polar codes and their construction. It then describes the basic encoding method for non-systematic and systematic codes. The low-complexity successive cancellation decoding method is reviewed with a simplified algorithm description. We continue with the description of the successive cancellation list decoding and also the CRC-aided successive cancellation list decoding, which are widely used in the literature. Finally, a review on puncturing techniques and 5G scenarios is carried out.

2.2
System Model

Figure 2.1 shows a block diagram of the polar coding system considered in this dissertation.

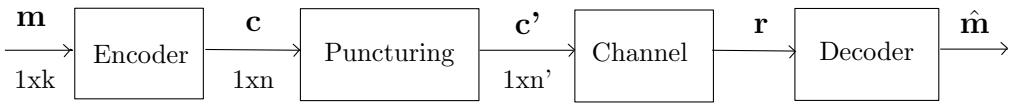


Figure 2.1: System Model.

In this system, $\mathbf{m}$ is the binary *message* vector that is transmitted, with k bits, where $\mathbf{m} \in \{0,1\}^{1\mathrm{X}k}$. It is through the generator matrix $\mathbf{G}$ that the message $\mathbf{m}$ is encoded, producing the *codeword* with n bits, that is,

$$\mathbf{c} = \mathbf{m} \cdot \mathbf{G} \tag{2-1}$$

with $\mathbf{c} \in \{0,1\}^{1\mathrm{X}n}$. With an appropriate puncturing technique, the message $\mathbf{c}$ has its length reduced to n' resulting in the punctured codeword **c'**, where $2^{l-1} < n' < 2^l$, $l = \log_2 n$ is an integer that defines the levels in the polarization tree. The *punctured codeword* **c'** is then transmitted over a channel with additive white Gaussian noise (AWGN), resulting in the received vector

$$\mathbf{r} = \mathbf{c'} + \mathbf{w}, \tag{2-2}$$

where $\mathbf{w}$ is the vector corresponding to the noise. In the decoding step, the decoding algorithm observes $\mathbf{r}$ in order to estimate $\mathbf{m}$. We call it an estimated message $\hat{\mathbf{m}}$, and if $\mathbf{m} = \hat{\mathbf{m}}$ we say that the message has been fully recovered. The problem we are interested in solving is how to design punctured codes with the best performance.

Let $W : \mathcal{X} \to \mathcal{Y}$ denote a binary discrete memoryless channel (B-DMC), with input alphabet $\mathcal{X}$, output alphabet $\mathcal{Y}$, and the channel transition probability $W(y|x)$, $x \in \mathcal{X}$, $y \in \mathcal{Y}$. The input alphabet can take {0,1} binary values, however the output alphabet and the transition probabilities are continuous such that they can take any arbitrary values in [0,1] interval. We write $W^n : \mathcal{X}^n \to \mathcal{Y}^n$ to denote the vector channel that corresponds to n independent uses of W with input alphabet $\mathcal{X}^n$, output alphabet $\mathcal{Y}^n$ and transition probabilities $W^n(y_1^n|x_1^n) = \prod_{i=1}^n W(y_i|x_i)$ [1]. Consider the input vector $\mathbf{m} = \mathbf{u}_1^k$ and the output vector $\mathbf{c} = \mathbf{x}_1^n$ and $\mathbf{r} = \mathbf{y}_1^n$. In general, we use the notation an $\mathbf{a}_1^n$ to denotes a vector $(a_1, a_2, \ldots, a_n)$ and $|\mathbf{a}_1^n|$ is the cardinality. Specifically $k = |\mathbf{m}| = |\mathbf{u}_1^k|$.

A channel W is defined as memoryless if its transition probability is

$$W_{\mathcal{Y}|\mathcal{X}}(y|x) = \prod_i^n W_{\mathcal{Y}_i|\mathcal{X}_i}(y_i|x_i). \tag{2-3}$$

The channel mutual information with equiprobable inputs, or symmetric capacity, is defined by

$$I(W) \triangleq \sum_{y\in\mathcal{Y}} \sum_{x\in\mathcal{X}} \frac{1}{2} W(y|x) log \frac{W(y|x)}{\frac{1}{2}W(y|0) + \frac{1}{2}W(y|1)} \tag{2-4}$$

and the corresponding reliability metric, the Bhattacharyya parameter is defined by

$$Z(W) \triangleq \sum_{y \in \mathcal{Y}} \sqrt{W(y|0)W(y|1)}. \tag{2-5}$$

Note that the Bhattacharyya parameter is the measure of reliability and it is an upper bound on the probability of maximum likelihood (ML) decision.

All logarithms used throughout this dissertation are base-2 logarithms; any different case will be stated explicitly.

2.3 Channel Polarization

The channel polarization operation creates n synthetic channels $\{W_n^{(i)} : 1 \leq i \leq n\}$ from n independent copies of the B-DMC W [1]. The polarization phenomenon decomposes the symmetric capacity, $I(W_n^{(i)})$ of these synthetic channels towards 0 or 1 such that $I(W_n^{(i)}) \simeq 0$ implies that the ith channel is completely noisy and $I(W_n^{(i)}) \simeq 1$ implies that the ith channel is perfectly noiseless. The capacity separation enables to send information (free) bits through the noiseless channels and redundancy (frozen) bits through the noisy channels.

Let $\mathcal{A}$ be the information set and $\mathcal{A}^c$ be the frozen set. The input vector, $\mathbf{u}_1^n$ consists of both information bits $u_\mathcal{A}$ and frozen bits $u_{\mathcal{A}^c}$ such that $u_\mathcal{A} \in \mathcal{X}^k$ and $u_{\mathcal{A}^c} \in \mathcal{X}^{n-k}$, $k = |\mathcal{A}|$.

In the following sections, we present the channel polarization as channel combining, channel splitting and channel construction.

2.3.1 Channel Combining

A B-DMC W_n is generated by combining two independent copies of $W_{n/2}$. At the 0th level of the recursion, we have only one copy of W and we set $W_1 \triangleq W$. The first level of the recursion combines two copies of W_1 for obtaining the channel $W_2 : \mathcal{X}^2 \to \mathcal{Y}^2$, as shown in Figure 2.2 [1] with transition probabilities, with $\oplus$ being the binary operator XOR,

$$W_2(y_1, y_2|u_1, u_2) = W(y_1|u_1 \oplus u_2)W(y_2|u_2). \tag{2-6}$$

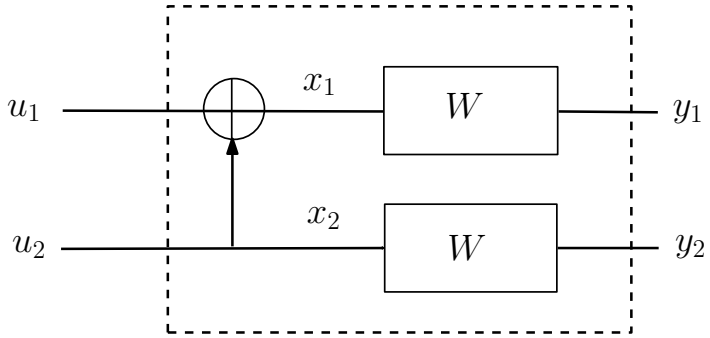


Figure 2.2: Construction of W_2 from W.

In the second level we obtain $W_4 : \mathcal{X}^4 \to \mathcal{Y}^4$ from two copies of W_2 as shown in Figure 2.3. In this case the transition probabilities are given by

$$W_4(y_1^4|u_1^4) = W_2(y_1^2|u_1 \oplus u_2, u_3 \oplus u_4)W_2(y_3^4|u_2, u_4), \tag{2-7}$$

where W denotes the smallest B-DMC, $\{0, 1\} \to Y$.

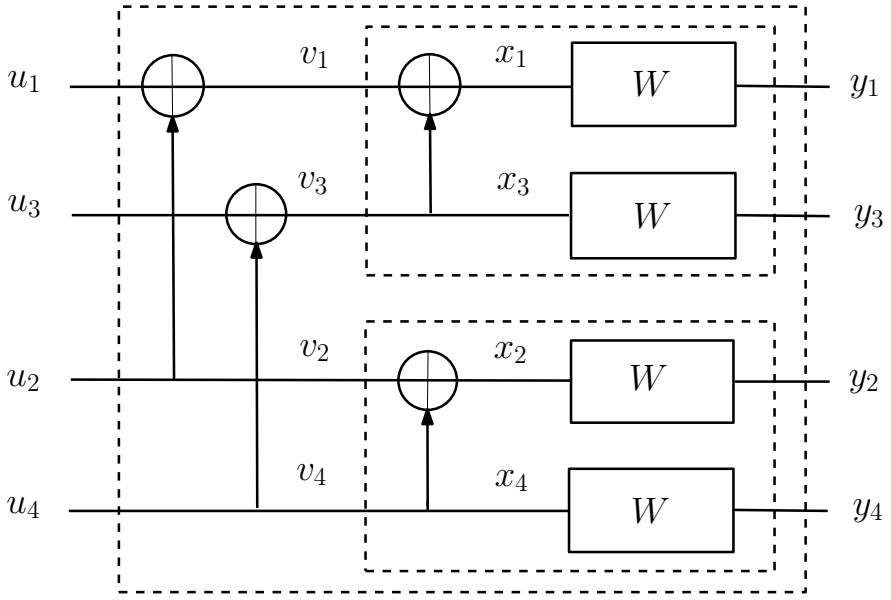


Figure 2.3: Construction of W_4 from W_2.

In a similar procedure, W_n is constructed recursively from $W_{n/2}$, $W_{n/4}$, ..., W_2, W at l steps, where $n = 2^l$ [1].

2.3.2 Channel Splitting

Having synthesized the vector channel W_n, the next step is channel splitting. This involves splitting the vector channel W_n into n B-DMCs $W_n^{(i)} : \mathcal{X} \to \mathcal{Y}^n \times \mathcal{X}^{i-1}$, $1 \leq i \leq n$. The transition probability [1] of the bit-channel $W_n^{(i)}$ is defined as

$$W_n^{(i)}(y_1^n, u_1^{i-1}|u_i) = \sum_{u_{i+1}^n \in \mathcal{X}^{n-i}} \frac{1}{2^{n-1}} W_n(y_1^n|u_1^n). \tag{2-8}$$

Each pair of the new B-DMC channels is obtained after applying the single-step polar transform to one of the channels of the previous step:

$$(W_{n/2}^{(i)}, W_{n/2}^{(i)}) \to (W_n^{(2i-1)}, W_n^{(2i)}). \tag{2-9}$$

We have seen how the local single-step polar transform affects the rate and reliability parameters. We next state the same results for the general case. For the recursive channel transformation [1] we have $(W_{n/2}^{(i)}, W_{n/2}^{(i)}) \to (W_n^{(2i-1)}, W_n^{(2i)})$, $n = 2^l$, $n \leq 0$, $0 \leq i \leq n$ and for the special case that W is a binary erasure channel (BEC) with erasure probability ϵ, we have that

$$I(W_n^{(2i-1)}) = I(W_{n/2}^{(i)})^2 \tag{2-10}$$

$$I(W_n^{(2i)}) = 2I(W_{n/2}^{(i)}) - I(W_{n/2}^{(i)})^2 \tag{2-11}$$

where $I(W_1^{(1)}) = 1 - \epsilon$, and

$$Z(W_n^{(2i-1)}) = 2Z(W_{n/2}^{(i)}) - Z(W_{n/2}^{(i)})^2 \tag{2-12}$$

$$Z(W_n^{(2i)}) = Z(W_{n/2}^{(i)})^2 \tag{2-13}$$

where $Z(W_1^{(1)}) = \epsilon$.

The bit-channel $W_n^{(i)}$ is the channel that a successive cancellation decoder sees when decoding the ith bit u_i with perfect knowledge of channel output $\mathbf{y}_1^n$.

Having combined two independent copies of W to create the vector channel $W_2 : \mathcal{X}^2 \to \mathcal{Y}^2$, we continue the channel polarization procedure by splitting back this super channel into two B-DMCs, $W_1 : \mathcal{X} \to \mathcal{Y}^2$ and $W_2 : \mathcal{X} \to \mathcal{Y}^2 \times \mathcal{X}$, defined as

$$W_2^{(1)}(y_1|u_1) = \frac{1}{2} \sum_{u_2 \in \mathcal{X}} W_2(y_1, y_2|u_1, u_2) W(y_1|u_2) \tag{2-14}$$

$$W_2^{(2)}(y_1, u_1|u_2) = \frac{1}{2} W_2(y_1, y_2|u_1, u_2) W(y_2|u_2). \tag{2-15}$$

Figure 2.5 shows $W_2^{(1)}$ and $W_2^{(2)}$. The transition probabilities are calculated in consecutive order from the top splitting operation to the bottom splitting operation, because the decision bit u_1 must be known before the bottom splitting operation.

Using Equations (2.10) and (2.11), we can show the polarization effect of the recursive method we are using. In Figure 2.4 we plot the polarization effect for the case of a BEC with erasure probability $\epsilon = 0.5$ and with the increase of n. Note that the channel polarization converges to $I(W) = 1$ and $I(W) = 0$.

2.3.3 Code Construction

The aim of the polar code construction is to determine $\mathcal{A}$ and $\mathcal{A}^c$ sets according to the capacity of individual channels [1] [2]. Since polar codes are channel specific codes, the code construction may differ from channel to channel. Channel parameters, such as σ for AWGN and ϵ for BEC are an input to a code construction method. For BEC W, code construction for $(n = 8,\ k = 4)$ polar code with an erasure probability $\epsilon = 0.3$ is shown in Figure 2.5.

Initially, the reliability of the smallest channel, $W_1^{(1)}$ sets as $\epsilon = 0.3$. After that the reliability of the first channel, $W_2^{(1)}$ is calculated as $Z(W_2^{(1)}) = 2Z(W_1^{(1)}) - Z(W_1^{(1)})^2$, where $Z(W_n^{(i)})$ is the erasure probability of the ith length n channel starting from top. At the same time, the second channel can be calculated as $Z(W_2^{(2)}) = Z(W_1^{(1)})^2$. In general, the recursive formula for calculating top and bottom channels is (2.12) and (2.13).

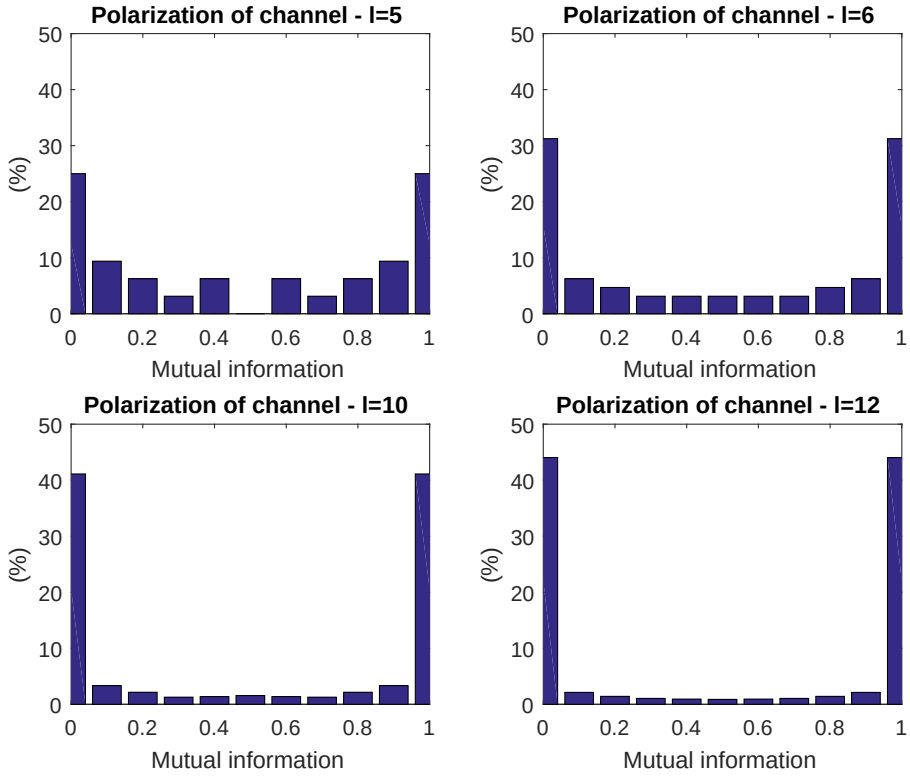


Figure 2.4: Channel Polarization

The polar code is designed by selecting the $|\mathcal{A}|$ channel to transmit information bits such that $Z(W_l^{(i)}) \leq Z(W_l^{(j)})$, with $i \in \mathcal{A}$ and $j \in \mathcal{A}^c$.

At the end of stage l (in this case $l = 3$), the erasure probability of all length n channels appears. At this point, the channels which have the lowest k erasure probabilities are set as free and the others are set as frozen. The algorithm has log n stages and performs the calculations. Polar code construction for a symmetric B-DMC is performed by using Monte-Carlo simulation [1].

2.4 Encoding of Polar Codes

Polar codes can be encoded by using a simple linear mapping. For the code block length n the generator matrix, $\mathbf{G}_n$ is defined [1] as

$$\mathbf{G}_n = \mathbf{B}_n \mathbf{F}^{\otimes l} \tag{2-16}$$

for any $n = 2^l$ as $l \geq 1$, where $\mathbf{B}_n$ is a bit-reversal matrix and $\mathbf{F}^{\otimes l}$ is

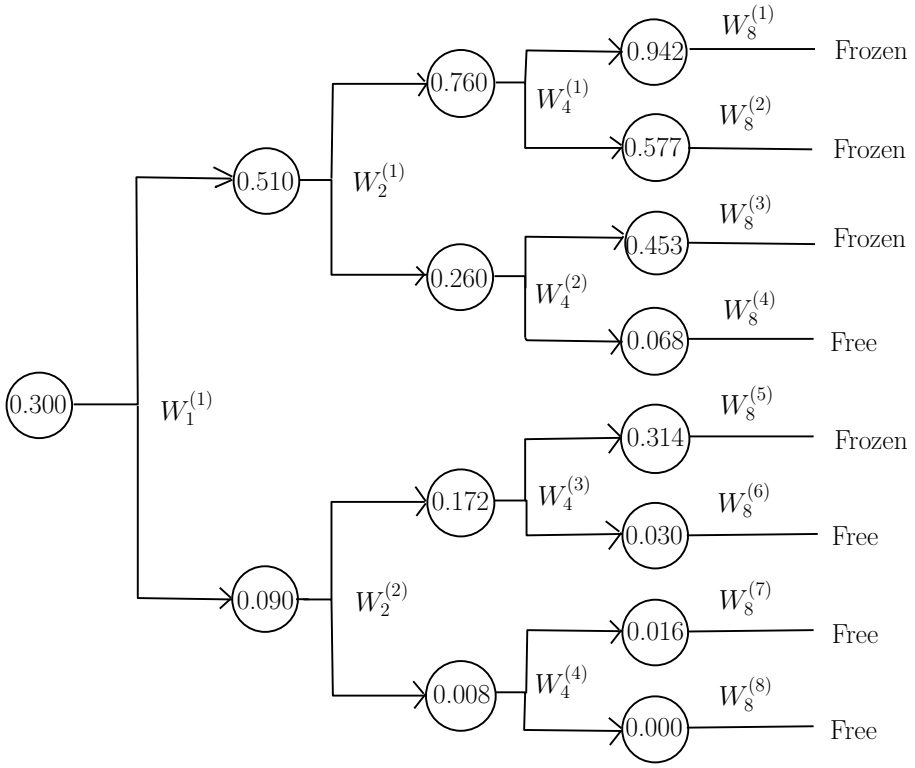


Figure 2.5: Code construction for BEC with n=8, k=4 and ϵ=0.3.

the nth Kronecker power of the matrix $\mathbf{F} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$. Below we show the construction of an array $\mathbf{F}^{\otimes 3}$:

$$\mathbf{F}^{\otimes 3} = \begin{bmatrix} \mathbf{F}^{\otimes 2} & 0 \\ \mathbf{F}^{\otimes 2} & \mathbf{F}^{\otimes 2} \end{bmatrix} = \begin{bmatrix} \mathbf{F} & 0 & 0 & 0 \\ \mathbf{F} & \mathbf{F} & 0 & 0 \\ \mathbf{F} & 0 & \mathbf{F} & 0 \\ \mathbf{F} & \mathbf{F} & \mathbf{F} & \mathbf{F} \end{bmatrix}$$

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

In [1], the input sequence $\mathbf{u}_1^n$ is permuted by a permutation matrix $\mathbf{B}_n$, i.e.,

$$\mathbf{x}_1^n = \mathbf{u}_1^n \mathbf{B}_n \mathbf{G}_2^{\otimes l} \tag{2-17}$$

Since the permutation only serves as a reordering of the indices of $(x_1, \ldots, x_n)$ and it does not affect the properties of polar codes, we skip this permutation for simplicity of presentation throughout the dissertation.

A length k polar encoder has an input vector $\mathbf{u}_1^n$ and an output vector $\mathbf{x}_1^n$. The mapping of $\mathbf{u} \rightarrow \mathbf{x}$ is linear over the binary field, GF(2) such that $\mathbf{x}_1^n = \mathbf{u}_1^n \mathbf{G}_n$. The rows of $\mathbf{G}_n$ are linearly independent and they form basis for the code space.

The factor graph representation of 8-bit encoder is shown in Figure 2.6, where $\oplus$ symbol represents binary XOR operation. Note that channel quality determines which channels will be data and which ones will be frozen.

As a result, the following equations are obtained at the output of 8-bit encoder.

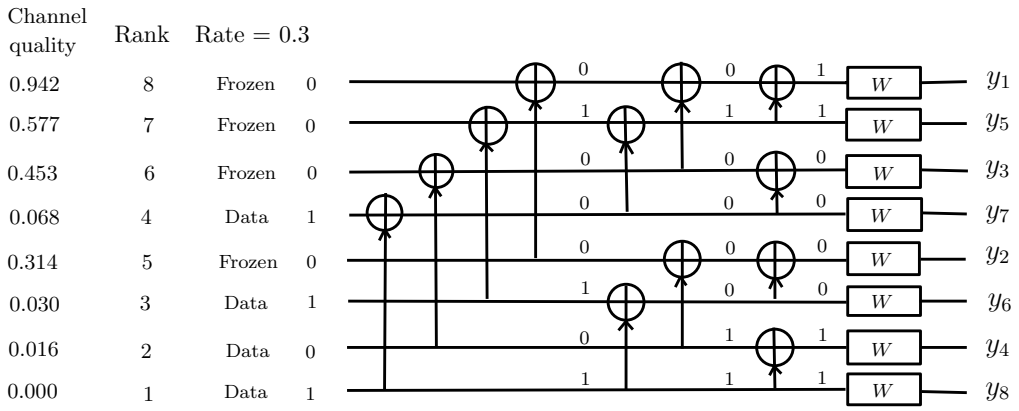


Figure 2.6: Encoding procedure for an (8,4) polar code, BEC with $\epsilon = 0.3$

$$
\begin{aligned}
x_1 &= u_1 \oplus u_2 \oplus u_3 \oplus u_4 \oplus u_5 \oplus u_6 \oplus u_7 \oplus u_8 \\
x_2 &= u_5 \oplus u_6 \oplus u_7 \oplus u_8 \\
x_3 &= u_3 \oplus u_4 \oplus u_7 \oplus u_8 \\
x_4 &= u_7 \oplus u_8 \\
x_5 &= u_2 \oplus u_4 \oplus u_6 \oplus u_8 \\
x_6 &= u_6 \oplus u_8 \\
x_7 &= u_4 \oplus u_8 \\
x_8 &= u_8
\end{aligned} \tag{2-18}
$$

In general, an n-bit encoder includes $(\frac{n}{2}\log_2 n)$ XOR operations. The free bits can be observed at the output of a polar encoder using non-systematic encoding of polar codes.

2.4.1 Systematic Encoding

Encoding schemes for polar codes can be either non-systematic, as shown in Figure 2.6, or systematic as discussed in [12]. Systematic polar codes offer better BER than their non-systematic counterparts, that is, there is an increase in the effective mean Hamming distance in the generated code word [9]; while maintaining the same Frame Error Rate (FER). Furthermore, they allow the use of low-complexity rate-adaptation techniques such as the code shortening method proposed in [25]. Flexible low-complexity systematic encoding of polar codes is discussed at length in [26].

The encoding scheme presented in the previous section, where the input vector $\mathbf{u}$ is multiplied by the generator matrix $\mathbf{G}$, results in a non-systematic polar codeword $\mathbf{x}$ in which information bits do not appear explicitly. A systematic codeword on the other hand can be information bits do not appear explicitly separated into information and parity bits, i.e. it includes the information bits unaltered. There are multiple advantages to systematic encoding: easier information extraction, better BER, and suitability for use in a turbo coding schemes are some examples.

A systematic encoding scheme for polar codes was introduced in [12] and was shown to improve the BER. This systematic encoding algorithm starts with the information bits, denoted $x_{\mathcal{A}}$ here, and calculates the parity bits according to

$$x_{\mathcal{A}^c} = x_{\mathcal{A}}(\mathbf{G}_{\mathcal{A}\mathcal{A}})^{-1}\mathbf{G}_{\mathcal{A}\mathcal{A}^c} \tag{2-19}$$

where $\mathbf{G}_{\mathcal{A}\mathcal{A}}$ is a submatrix of $\mathbf{G}$ whose rows and columns correspond to the information bit indices, and $\mathbf{G}_{\mathcal{A}\mathcal{A}^c}$'s rows and columns corresponds to the information and frozen bits, respectively. The parity and information bits are then combined into a systematic codeword where they are placed in the frozen and information bit locations, respectively.

When the recursive nature of the generator matrix $\mathbf{G}$ is exploited, a serial implementation of this algorithm has a time complexity of $\mathcal{O}(n \log n)$; where n is the code length.

Figure 2.7 shows an example of the low-complexity systematic encoding scheme proposed in [26]. It comprises two non-systematic encoding passes and a bit masking operation in between. For a (8,4) polar code, a n bit vector $\mathbf{u} = (0, 0, 0, a_0, 0, a_1, a_2, a_3)$, where $(a_0, \ldots, a_3)$ are the $k = 4$ information bits, enters the first non-systematic encoder from the left. Then, using bit masking, the locations corresponding to frozen bits are reset to 0 before propagating the updated vector through the second non-systematic encoder.

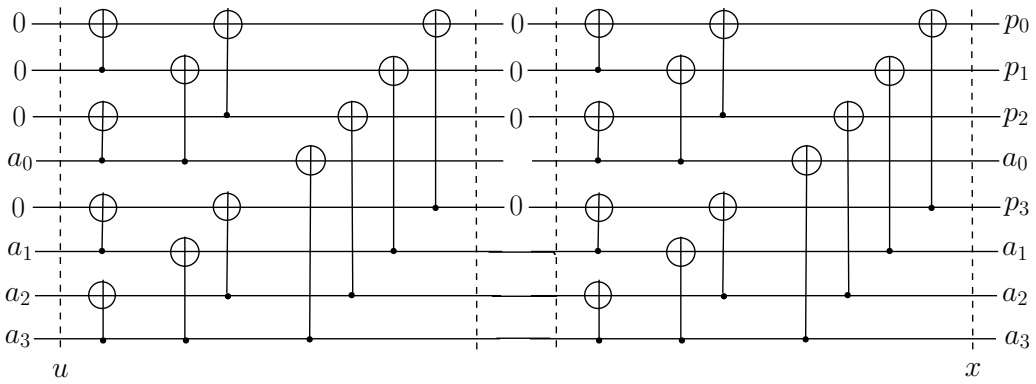


Figure 2.7: Systematic encoding of a (8,4) polar code.

The systematic coding can be obtained with the following expression, obtained prior to puncturing,

$$\mathbf{x} = \mathbf{u}\mathbf{G}_n\mathbf{p}\mathbf{G}_n, \tag{2-20}$$

where $\mathbf{p} = [u_i(i \in \mathcal{A}^c) = 0, u_i(i \notin \mathcal{A}^c) = 1]$. In decoding, we have

$$\hat{\mathbf{u}}' = \hat{\mathbf{u}}\mathbf{G}_n, \tag{2-21}$$

where $\hat{\mathbf{u}}$ is obtained by SC.

Figure 2.8 shows the performance of polar codes with $n = 1024$ and rate r = 0.5 using systematic and non-systematic codes in AWGN channel. We measure the BER against the signal-to-noise ratio, defined as the ratio of the bit energy, E_b, and the power spectral density, N_0, in dB. The results show that systematic codes obtain significant gains over non-systematic codes as compared to a non-systematic encoding scheme even though it has not been established theoretically in the literature why this happens [12].

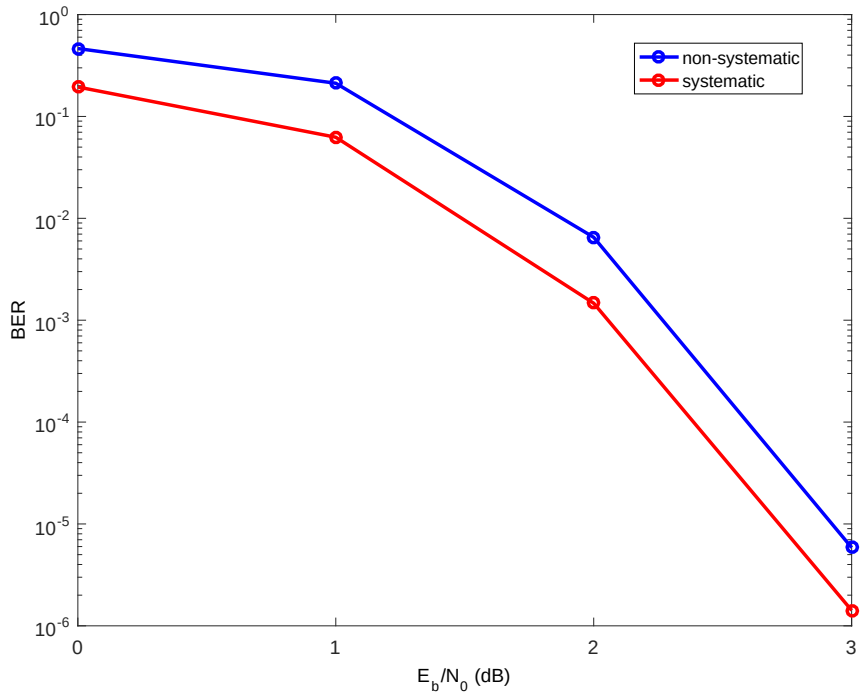


Figure 2.8: Comparison of systematic and non-systematic polar codes.

2.5 Successive Cancellation Decoding

It has been proved in [1] that polar codes with SC decoding can achieve the capacity of B-DMCs. Since then, many other decoding algorithms have been proposed [18], [21], [22], [23] and [24]; to improve the error rate performance in the finite-length regime. We will review the SC decoding algorithm in this section.

SC decoder estimates the transmitted bits $\mathbf{u}_1^n$ as $\hat{\mathbf{u}}_1^n$ by using the received codeword $y \in \mathcal{Y}$ at the B-DMC, $W_n : \mathcal{X}^n \rightarrow \mathcal{Y}^n$. The channel likelihood information gained from the received codeword is represented as log likelihood ratios (LLR). The decoder performs soft-decision decoding as computing intermediate LLR values by using channel LLR values. After a sequence of LLR computations, SC decoder computes $\hat{\mathbf{u}}_1^n$ hard decisions in a successive order from $\hat{u}_1$ to $\hat{u}_n$. In other words, $\hat{u}_i$ is decided according to $\hat{u}_1^{i-1}$ for $1 < i \leq n$.

A high level description of the SC decoding algorithm is illustrated in Algorithm 1 [10] [11]. The algorithm takes the received codeword $\mathbf{y}_1^n$, the code block length n and the information set $\mathcal{A}$ as input and calculates the estimated free bits $\hat{u}_\mathcal{A}$ as an output vector. There are n decision steps in the algorithm. If a hard decision belongs to the frozen set $\mathcal{A}^c$, the decision will be a frozen decision such that it is known by both encoder and decoder. Otherwise, the decoder sets its hard decision with respect to the soft decision information. After all n decisions are calculated, the output of the decoder is made by the hard decisions, which belong to the free set.

Algorithm 1: Successive Cancellation Decoding

Input: received codeword, $\mathbf{y}_1^n$
Input: code block length, n
Input: information set, $\mathcal{A}$
Input: frozen bit vector, $u_{\mathcal{A}^c}$
Output: estimated free bits, $\hat{u}_\mathcal{A}$

```
begin
    for i ⟵ 1 : n do
        if i ∉ A then
            û_i ⟵ u_i
        else
            if log( W_n^(i)(y_1^n, û_1^{i-1} | û_i = 0) / W_n^(i)(y_1^n, û_1^{i-1} | û_i = 1) ) ≥ 0 then
                û_i ⟵ 0
            else
                û_i ⟵ 1
    return û_A
```

Three different functions are defined to illustrate the behavior of the SC decoder. These functions are called f, g, and d. Firstly, the f function is responsible for the calculation of top channel splitting operation, defined in Section 2.3.2. The f function, with likelihood ratio (L) representation is

$$
\begin{aligned}
f(L_a, L_b) &= L_c && (2\text{-}22)\\
&= \frac{W(y_1^2|\hat{u}_c = 0)}{W(y_1^2|\hat{u}_c = 1)} && (2\text{-}23)\\
&= \frac{W(y_1|\hat{u}_a = 0)W(y_2|\hat{u}_b = 0) + W(y_1|\hat{u}_a = 1)W(y_2|\hat{u}_b = 1)}{W(y_1|\hat{u}_a = 0)W(y_2|\hat{u}_b = 1) + W(y_1|\hat{u}_a = 1)W(y_2|\hat{u}_b = 0)} && (2\text{-}24)\\
&= \frac{L_a L_b + 1}{L_a + L_b} && (2\text{-}25)
\end{aligned}
$$

where in (2-25) both numerator and denominator are divided by $W(y_1|\hat{u}_a = 1)W(y_2|\hat{u}_b = 1)$. The f function, with LLR (γ) representation is

$$
\begin{aligned}
f(\gamma_a, \gamma_b) &= \gamma_c && (2\text{-}26)\\
&= 2tanh^{-1}(tanh(\frac{\gamma_a}{2})tanh(\frac{\gamma_b}{2}) && (2\text{-}27)\\
&= sign(\gamma_a\gamma_b)min(|\gamma_a|, |\gamma_b|) && (2\text{-}28)
\end{aligned}
$$

where the min-sum approximation is defined for Belief Propagation (BP) decoding of LDPC codes [13] and this approximation was used in SC decoding of polar codes for the first time in [14].

Secondly, the g function computes the bottom channel splitting operation in the SC decoder. The g function, with likelihood ratio (L) representation of soft decisions is

$$
g(L_a, L_b, \hat{u}) = \frac{W(y_1^2|\hat{u}_c = 0)}{W(y_1^2|\hat{u}_c = 1)} \tag{2-29}
$$

$$
g(L_a, L_b, \hat{u}) = \begin{cases} \frac{W(y_1|\hat{u}_a=0)W(y_2|\hat{u}_b=0)}{W(y_1|\hat{u}_a=1)W(y_2|\hat{u}_b=1)} & \text{if } \hat{u} = 0 \\ \frac{W(y_1|\hat{u}_a=1)W(y_2|\hat{u}_b=0)}{W(y_1|\hat{u}_a=0)W(y_2|\hat{u}_b=1)} & \text{if } \hat{u} = 1 \end{cases} \tag{2-30}
$$

$$
g(L_a, L_b, \hat{u}) = L_a^{(1-2\hat{u}_c)} L_b \tag{2-31}
$$

The g function, with LLR representation is

$$
g(\gamma_a, \gamma_b, \hat{u}) = (-1)^{(\hat{u})}\gamma_a + \gamma_b \tag{2-32}
$$

Lastly, the d function is the decision function, which computes hard decisions from soft decisions such that

$$\hat{u}_i = \begin{cases} 0, & if \quad i \notin A \\ 0, & if \quad i \in A \quad \text{and} \quad \frac{W(y,\hat{u}_1^{(i-1)}|\hat{u}_i=0}{W(y,\hat{u}_1^{(i-1)}|\hat{u}_i=1} \geq 1 \\ 1, & \text{otherwise} \end{cases} \tag{2-33}$$

In (2-31), $\hat{u}_c$ represents a partial sum of previously-decoded bits. This partial sum corresponds to the propagation of decoded bits from left to right in the decoding graph.

Figure 2.9 [46] graphically shows how likelihood ratios (L) are calculated at decision nodes, starting from the channel L, through f and g functions in a recursive manner. The same applies for LLR processing.

Table 2-1 lists the operations carried out to decode each bit $\hat{u}_i$; it describes the sets of nodes activated in each stage of Figure 2.9, and serves as a starting point from which the decoding schedules explored in the following chapters are derived. This activation pattern can be determined from index i, by treating it as a binary number: bit l of that number corresponds to stage l, and f (resp. g) is used if this bit is 0 (resp. 1). Note that every node of the decoding graph must be activated to decode a received codeword.

Table 2.1: Operations performed by a SC decoder with $n = 8$.

Stage, i	0	1	2	3	4	5	6	7
$l = 2$	**f**	f	f	f	**g**	g	g	g
$l = 1$	**f**	f	**g**	g	**f**	f	**g**	g
$l = 0$	**f**	**g**	**f**	**g**	**f**	**g**	**f**	**g**

Now, we describe the successive cancellation (SC) decoding process for $n = 4$, shown in Figure 2.10.

First, $c_0 = f(L_{y_0}, L_{y_2})$ and $c_1 = f(L_{y_1}, L_{y_3})$ correspond to a $l = 1$ check node combining operation; $\hat{u}_0 = f(c_0, c_1)$ is estimated, which corresponds to a $l = 0$ check node combining operation.

Now, $\hat{u}_1$ can be estimated from c_0, c_1 and $\hat{u}_0$. Since the two check nodes have been already processed before, in this step only a single $l = 0$ variable node operation has to be performed, $\hat{u}_1 = g(c_0, c_1, \hat{u}_0)$.

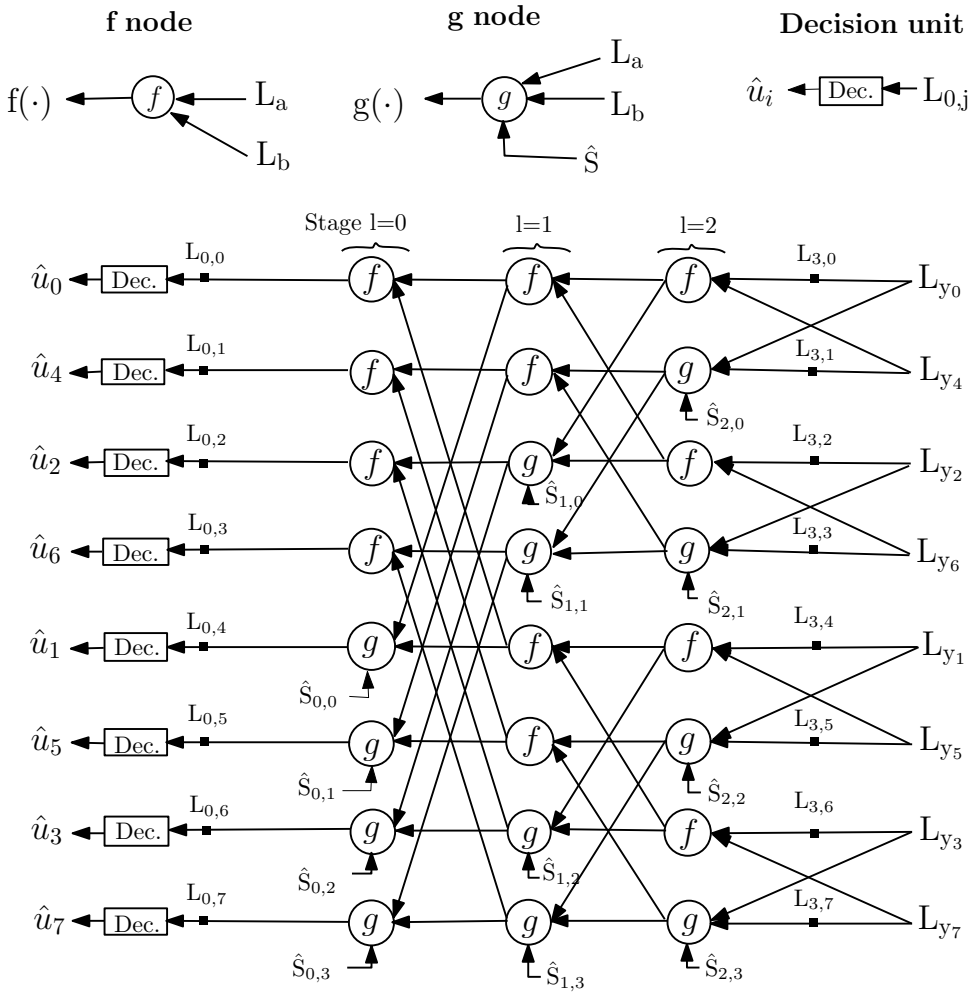


Figure 2.9: Structure of the polar decoding graph for $n = 8$

Using the relations $c_2 = g(L_{y_0}, L_{y_2}, \hat{u}_0 \oplus \hat{u}_1)$ and $c_3 = g(L_{y_1}, L_{y_3}, \hat{u}_0)$, the estimation of $\hat{u}_2$ involves a binary computation tree with two variable nodes at the bottom level and a check node at the top, $\hat{u}_2 = f(c_2, c_3)$.

Finally, $\hat{u}_3$ is obtained by a $l = 0$ variable node operation. Like in the second step, actually only the node on top of the tree has to be processed, $\hat{u}_3 = g(c_2, c_3, \hat{u}_2)$.

2.6 Successive Cancellation List Decoding

The list decoding algorithm has been introduced by Tal and Vardy [15] and generates $\mathcal{L}$ candidates for decoding or possibilities, increasing the probability that one of the results will be correct when compared to SC [19] [20]. With the use of CRC, this method has proven to be very efficient for polar codes. If $\mathcal{L}$ is large enough, the performance of ML decoding is achieved because a sufficient number of decoding paths are visited [42].

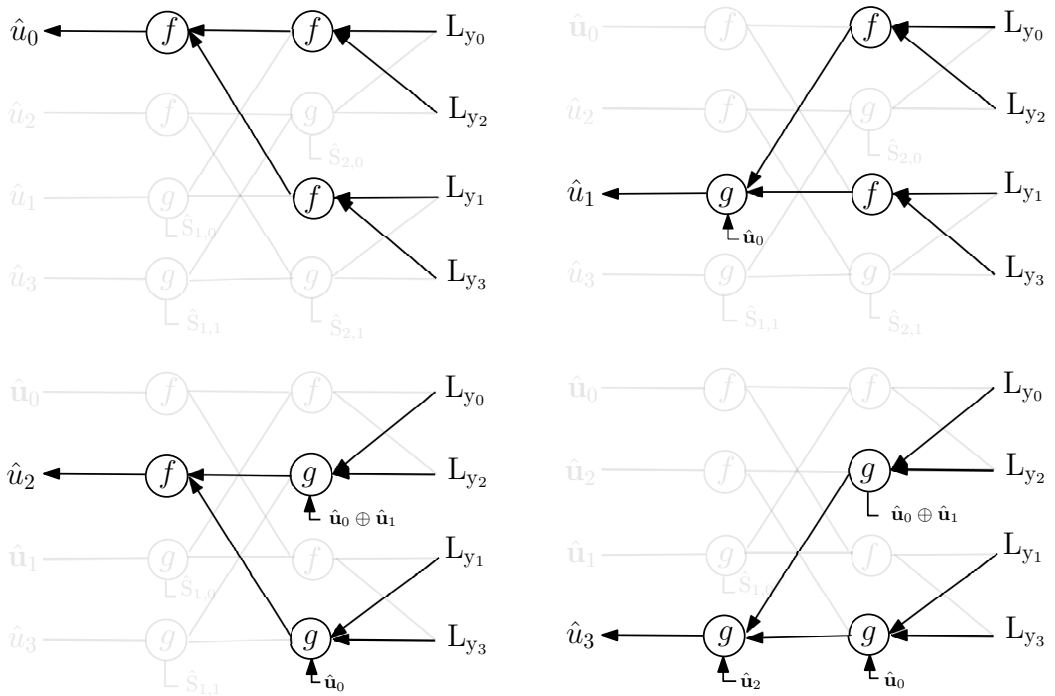


Figure 2.10: Successive cancellation (SC) decoding for the polar code with $n = 4$

Different from the selection of a particular path in each decision step in the SC decoding, the list decoding (SCL) extends two paths, $\hat{u}_i = 0$ and $\hat{u}_i = 1$, for each decision step. The complexity of SCL decoding is constrained by the size of the list. The complexity of SLC is $\mathcal{O}(\mathcal{L}n \log n)$. So $\mathcal{L}$ more reliable paths or paths are preserved at every decision step.

Let $\hat{u}_i[\tau]$ be the estimate of $\hat{u}_i$ in the τth path, where $\tau \in 1, 2, ..., \mathcal{L}$. The path metric (PM) is used to measure the reliability of the paths. The PM of $\hat{u}_i[\tau]$ is approximated by

$$PM_1^{(i)} = PM_1^{(i-1)} + c_i[\tau] \cdot |\mathcal{L}_n^{(i)}[\tau]|, \tag{2-34}$$

where i is the index of the decision step, $PM_\tau^{(0)} = 0$, $c_i[\tau] = 0$ when $i \in A^c$ and $c_i[\tau] = 1$ otherwise.

Since u_i has two possible values, the $\mathcal{L}$ paths or candidates with lower metrics are selected in the decision step. After the last bit has been decoded, the paths or path with lower metric is considered as the decoding result.

A high level description of the algorithm is shown in Algorithm 2. The SCL algorithm takes the received codeword $\mathbf{y}_1^n$, the code block length n, the information set $\mathcal{A}$, the frozen bit vector $u_{\mathcal{A}^c}$ and the maximum list size $\mathcal{L}$ as

input and calculates the estimated information bits $\hat{u}_{\mathcal{A}}$ as output. The current list size variable ($\mathrm{c}\mathcal{L}$) is set to 1 at the initialization of the algorithm. If the ith hard decision belongs to the frozen set $\mathcal{A}^c$, the ith hard decisions of all $\mathcal{L}$ lists are updated with the frozen decision, u_i. In case of a free decision, the decoder checks whether the current list size is equal to the maximum list size. If they are not equal, the current list size doubles and the decoder can track likelihoods of both decisions. In case all lists are occupied, the decoder sorts $2\mathcal{L}$ likelihoods to continue with the best $\mathcal{L}$ decoding paths. At the end of the last decision step, the decoder outputs the free bits from the best entry in the list as $\hat{u}_{\mathcal{A}}$.

Algorithm 2: Successive Cancellation List Decoding

Input: received codeword, $\mathbf{y}_1^n$
Input: code block length, n
Input: information set, $\mathcal{A}$
Input: frozen bit vector, $u_{\mathcal{A}^c}$
Input: maximum list size, $\mathcal{L}$
Output: estimated free bits, $\hat{u}_{\mathcal{A}}$

```
1  begin
2      cL ⟵ 1, // current list size variable
3      for i ⟵ 1 : n do
4          if i ∉ A then
5              for τ ⟵ 1 : cL do
6                  û_{τ,i} ⟵ u_i
7          else
8              if cL ≠ L then
9                  for τ ⟵ 1 : cL do
10                     û_{τ,i} ⟵ 0
11                     û_{τ+cL,i} ⟵ 1
12                 cL ⟵ 2cL
13             else
14                 s ⟵ sort(W_n^(i)(y_1^n, û_1^{i-1} | û_{1,i}^L))
15                 for τ ⟵ 1 : cL do
16                     û_{τ,i} ⟵ s
17     return û_A
```

An example for $\mathcal{L} = 4$ is considered in Figure 2.11. Assume $n = 4$ and all bits are unfrozen. In (a) the decoding algorithm starts and the first bit can be either 0 or 1.

In the second step (b) the second bit assumes either 0 or 1 thus the the possible words are {00, 01, 10, 11} but the number of paths is not greater than $\mathcal{L} = 4$.

In (c) it is shown all possible options for the first, second and third bit but now the paths are 8 so we must keep track of only the $\mathcal{L} = 4$ most likely paths.

In (d) it is shows that we keep only the words {010, 011, 100, 111}. Decoding with the list algorithm continues for the fourth bit in (e); however, the paths are 8 which is too much so it prunes for the best $\mathcal{L} = 4$ paths in (f). Finally, the decoding algorithm terminates and we obtain the codewords {0100, 0110, 0111, 1111}.

Unfortunately, the performance of polar codes is in general worse than turbo and LDPC codes of similar length. However, researchers noticed that the transmitted codeword is not the most likely codeword but it is on the list that the decoder generates. Thus, the performance could be improved using an additional CRC. Using both CRC and SCL decoder, they discovered that the performance is comparable to state of the art LDPC and turbo codes.

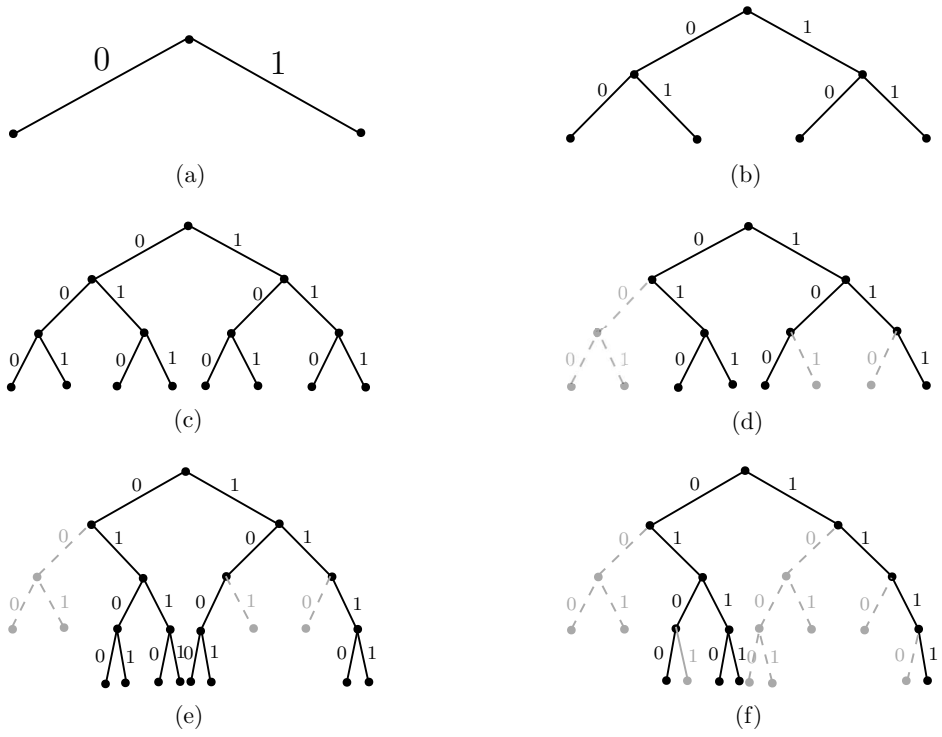


Figure 2.11: Evolution of decoding paths for $\mathcal{L} = 4$.

Figure 2.12 shows the performance of decoders per list with different $\mathcal{L}$ list sizes for systematic polar codes in the eMBB scenario with $k = 256$ and $n = 480$. The results show that the performance of the decoder per list is equivalent to the successive decoder for a list with $\mathcal{L} = 1$ and improves with the increase of candidates in the lists. In particular, the gains from the use of

lists are quite significant over the successive decoder ($\mathcal{L} = 1$) and the gains decrease as the list size $\mathcal{L}$ increases, a trend known as diminishing returns.

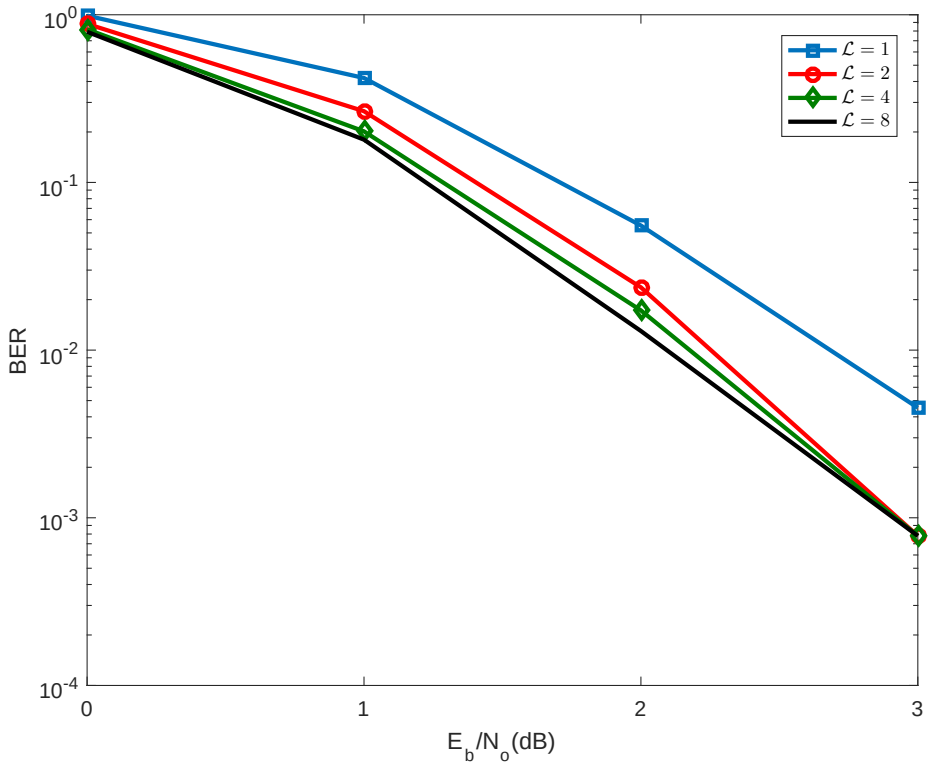


Figure 2.12: Comparison of list decoder for different list sizes.

2.7 CRC-Aided Successive Cancellation List Decoding

The CRC-Aided Successive Cancellation List Decoding (CA-SCL) algorithm consists of SC decoding, SCL decoding and CRC decoding algorithms. The aim of the algorithm is to increase the throughput of the SCL decoder in [15], [16], [52] and [53].

A high level description of the algorithm is shown in Algorithm 3. Inputs of the adaptive SCL decoding algorithm are the received codeword $\mathbf{y}_1^n$, the code block length n, the information set $\mathcal{A}$, the frozen bit vector $u_{\mathcal{A}^c}$ and the list size $\mathcal{L}$. The output of the algorithm is the free bit vector $\hat{u}_{\mathcal{A}}$.

At the beginning of the algorithm, the SC decoder calculates a free bit candidate vector. If the CRC of that vector is true, the algorithm terminates with the output of the SC decoder. In case of incorrect CRC vector, the algorithm calls a SCL algorithm with the list $\mathcal{L}$. At this time, the SCL algorithm calculates $\mathcal{L}$ hard decision candidate vectors. If one of them has

a valid CRC, the algorithm terminates with that output. If none of them has a valid CRC, the algorithm terminates with the most probable hard decision candidate vector [17].

Algorithm 3: CRC-Aided Successive Cancellation List Decoding

Input: received codeword, $\mathbf{y}_1^n$
Input: code block length, n
Input: information set, $\mathcal{A}$
Input: frozen bit vector, $u_{\mathcal{A}^c}$
Input: maximum list size, $\mathcal{L}$
Output: estimated information bits, $\hat{u}_{\mathcal{A}}$

```
1  begin
2      j ⟵ false // valid CRC vector of SC variable
3      k ⟵ false // valid CRC vector of SCL variable
4      û_A ⟵ Sucessive Cancellation Decoding (y_1^n, n, A, u_{A^c})
5      j ⟵ Cyclic Redundancy Check Decoding (û_A)
6      if j is true then
7          return û_A
8      else
9          û_{l,A} ⟵
             Sucessive Cancellation List Decoding (y_1^n, n, A, u_{A^c}, L)
10         for l ⟵ 1 : L do
11             k ⟵ Cyclic Redundancy Check Decoding (û_{l,A})
12             if k is true then
13                 û_A ⟵ û_{τ,A}
14                 return û_A
15         û_A ⟵ û_{τ,A}
16         return û_A
```

The CRC polynomial project [51] depends on the maximum total length n of the block to be designed considering data and CRC. According to [38], where d is the degree of the generator polynomial, the maximum total length of the block is given by 2^{d-1}. The CRC-8 is to be used for $n \leq 128$ and the CRC-16 for is suggested $n \leq 32768$.

Figure 2.13 shows the performance of the list decoders with different CRC for systematic polar codes in the eMBB scenario with rate r = 0.25 and $n = 1920$. The results show that the performance of the decoder per list improves with the increase of candidates in the lists and size of the CRC.

2.8 Puncturing and extension

Various puncturing methods for polar codes have been proposed in the literature. The work in [5] introduces a classification based on the type of

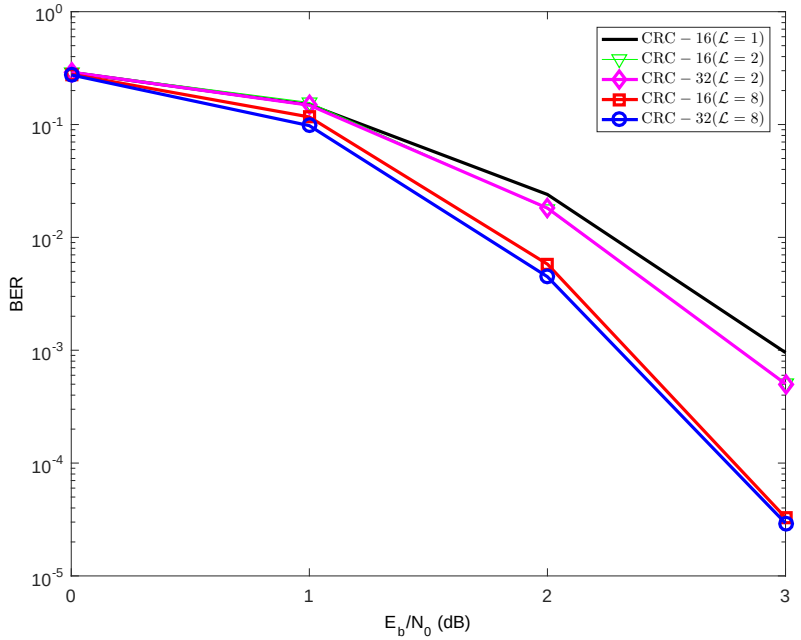


Figure 2.13: Comparison of list decoder with different CRC.

decoder, successive cancellation (SC) and belief propagation (BP).

For the BP decoder, a puncturing method is used with optimization techniques along with communication systems employing retransmission techniques, such as Hybrid Automatic Repeat reQuest (HARQ) in [27], [28], [29] and [30]; by either making use of different properties of the punctured code, in terms of minimal distance, exponent bound, stopping-tree puncturing or the reduced generator matrix method in [30], [31], [32] and [33]; or relying on the density evolution analysis [34] and [35].

On the other hand, for the puncturing schemes operating with the SC decoder, the work in [36] introduced the concept of capacity-one puncturing and devised a simple puncturing method based on the columns weight (CW), whereas the study in [37] proposed a search algorithm to jointly optimize the puncturing patterns and the values of the punctured bits.

Figure 2.14 shows the representation of the generating matrix $\mathbf{G}$ and its various indices and variables within the context of polar codes, which can be used as a puncturing criterion: the polarization rank, the partial distance and the decoding sequence.

Figure 2.15 shows the equivalent graph that represents the encoding and decoding procedure with the puncturing based on the generator matrix $\mathbf{G}$. This equivalence has the advantage that the simplification of the implementation of the puncturing algorithm and as a reference for code implementation.

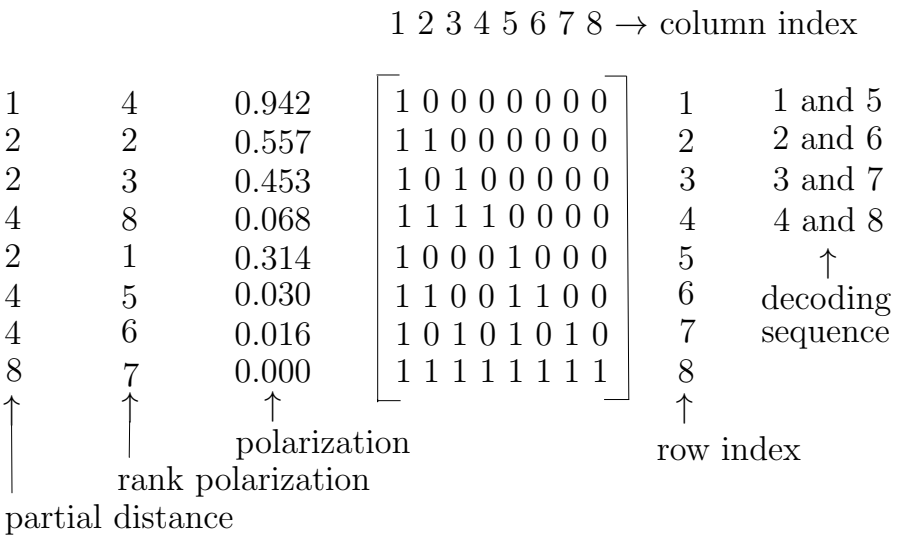


Figure 2.14: Puncture criterions for Matrix $\mathbf{G}$.

Below we show an example of puncturing the generator matrix $\mathbf{G}_8$ to get $\mathbf{G}_5$:

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ \cancel{1} & \cancel{0} & \cancel{1} & \cancel{0} & \cancel{1} & \cancel{0} & \cancel{1} & \cancel{0} \\ \cancel{1} & \cancel{1} & \cancel{1} & \cancel{1} & \cancel{1} & \cancel{1} & \cancel{1} & \cancel{1} \end{pmatrix} \rightarrow \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 \end{pmatrix}$$

and the graph equivalent of the puncturing is shown in Figure 2.16.

Extension techniques for polar codes is an unexplored subject. Arikan [1] suggests a second decoding algorithm based on the extension of an primitive SC encoder. An encoder could be extended with frozen bits and its implementation should follow the previous suggestion. Despite the lack of references to an extension technique for polar codes, in an initial analysis it is observed that small code extensions can be achieved with few modifications in the encoding and decoding algorithms.

2.9 5G scenarios

The requirements of 5G systems that affect channel coding techniques are performance in terms of bit error rate (BER), latency, computational complexity and energy efficiency [3].

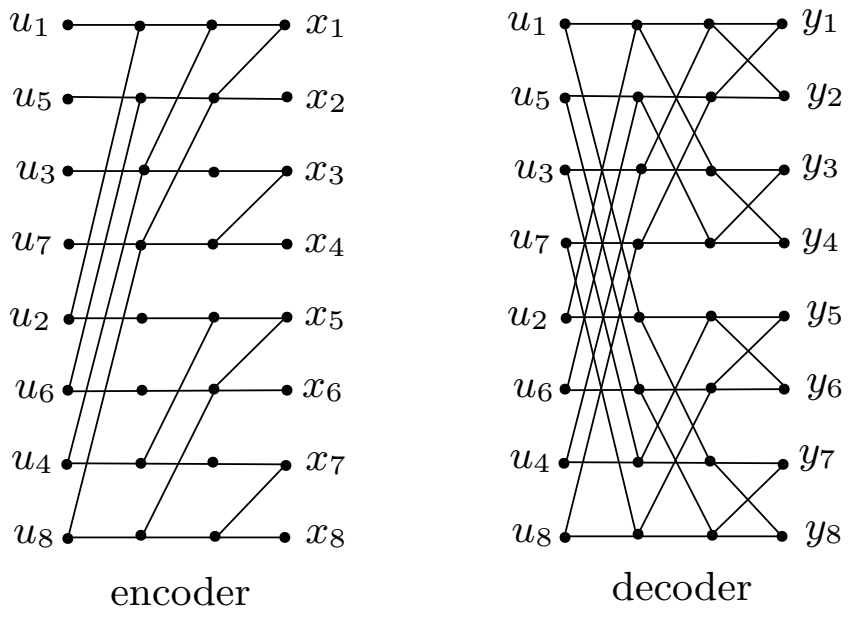


Figure 2.15: Graph represents for encoder and decoder

In addition to the requirements [3] [47], the scenarios for applications in 5G systems require different coding approaches [48] [49]. The first scenario called Bitpipe is dedicated to applications with high transmission rates of users with low mobility, where transmission rates should be higher than 10Gbps, latency should be less than 10ms and the block size corresponding to the codeword should be 64800 bits or 1920 bits and the code rate should be r = 0.83 to ensure high transmission efficiency. It is important to note that this last codeword length, $n = 1920$, is best suited for applications with multiple users that can decode the codeword independently of other users.

The second scenario called eMBB is dedicated to applications with high rates of users with different degrees of mobility and rural environments where transmission rates should be higher than 10Mbps, latency should be less than 100ms and block size corresponding to code must be 64800 bits or 1920 bits and the code rate should be r = 0.25 to ensure high performance.

The third scenario is called the Tactile Internet and consists of low latency transmissions, should be less than 1ms, where the codeword size is only 480 bits and the code rate should be r = 0.53.

The fourth scenario called IoT communication focuses on transmissions between machines and Internet of Things devices, where transmission rates should be in the range of 1Kbps to 10Mbps, the latency should be between 1 and 100ms, the size of the code word should be 480 bits and the code rate should be R = 0.53.

The last scenario, called machine-to-machine (M2M) communication,

focuses on transmissions between machines, where transmission rates should be in the range of 1Kbps to 10Mbps, the latency should be between 1 and 100ms, the size of the codeword should be 100 bits and the code rate should be r = 0.64.

Table 2.2 illustrates the requirements mentioned for the different scenarios of 5G, in which R_b is the baud rate; n is codeword length; t is the latency and r is the coding rate.

Table 2.2: 5G requirements

	Bitpipe	eMBB	Tactile Internet	IoT	M2M
R_b	R_b>10Gbps	R_b>10Mbps	R_b<10Mbps	R_b<10Mbps	R_b<10Mbps
n	1920	1920	480	480	100
t	t<10ms	t<100ms	t<1ms	t<100ms	t<100ms
r	0.83	0.25	0.53	0.53	0.64

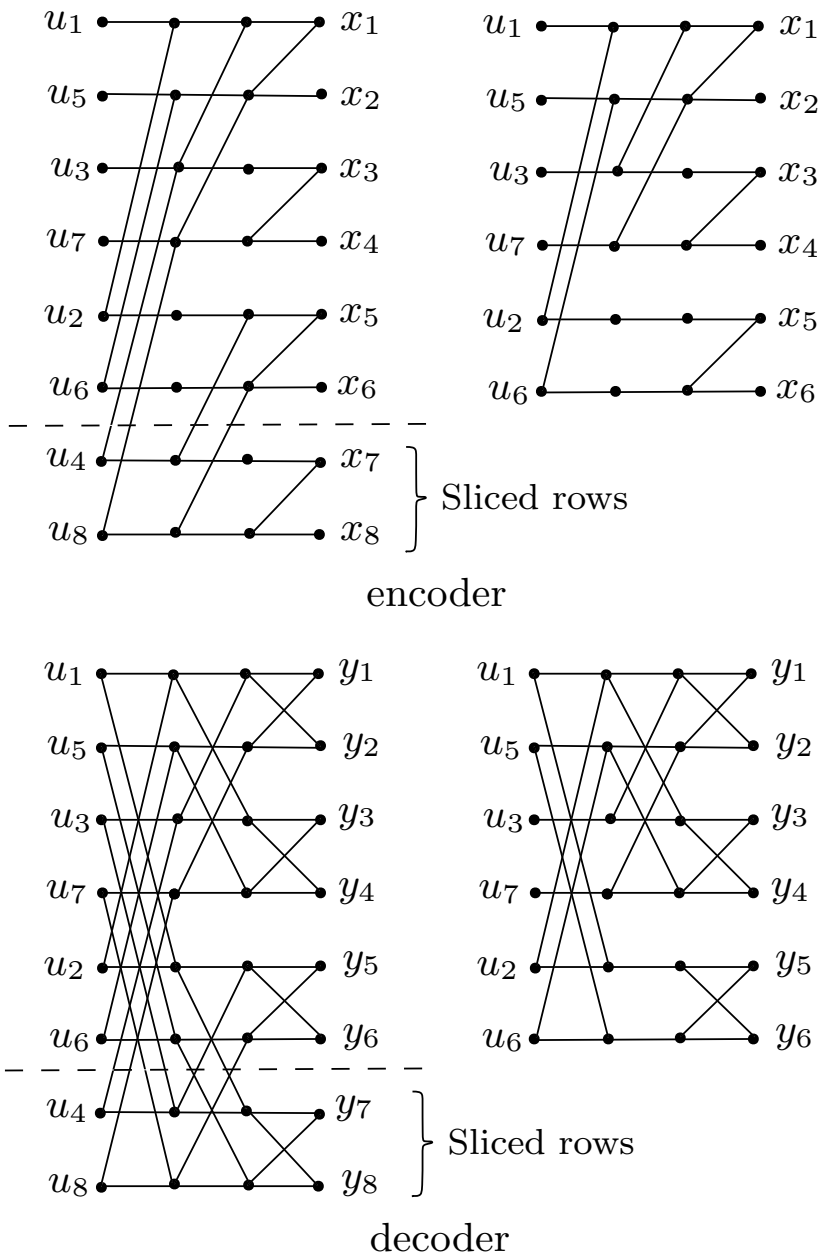


Figure 2.16: Punctured Graph for the encoder and the decoder.

3
Proposed Polarization-Driven Techniques

3.1
Introduction

This chapter presents the polarization-driven (PD) puncturing technique for the design of punctured polar codes. The proposed puncturing strategy consists of reducing the generator matrix by relating its row index based on the channel polarization principle. The punctured codes constructed based on channel polarization are then considered with successive cancellation (SC) decoding and punctured bits known to both the encoder and the decoder. The Spectrum Distance (SD) and the Joint Spectrum Distance (JSD) are then used to performance analysis. The technique is then applied in the 5G scenarios. Simulation results show that the proposed punctured polar codes outperform when compared to the techniques proposed by Niu [5] and Wang [6].

The rest of this chapter is presented as follows: in Section 3.2, we will provide the mathematical description of the method in detail, where we will present the channel polarization procedure, the choice of punctured channels, and the reduction of the generator matrix $\mathbf{G}$. Section 3.3 describes the puncturing method in algorithmic form, allowing a straightforward implementation. Section 3.4 discusses the SD and JSD analysis method, the relation between the polarized channels W_i and the rows of the generator matrix $\mathbf{G}$, and a comparison is made with the results obtained by the techniques proposed by Niu [5] and Wang [6]. In Section 3.5, we analyze the results obtained in the simulation when applied to the scenarios discussed, where the performance improvement is observed.

3.2
Mathematical description

In this section, we detail the proposed PD puncturing technique. The purpose of puncturing is to reduce the size of the generator matrix $\mathbf{G}_n \in \mathbb{R}^{k\times n}$ from $k \times n$ to $k \times n'$, such that $n' < n$. In particular, the size reduction is obtained by eliminating row and columns of the matrix $\mathbf{G}_n$. Consider the punctured vector $\mathbf{p}$ which contains the indices of the rows of the matrix $\mathbf{G}_n$ to

be punctured, where $|\mathbf{p}| = n - n'$ designates its number of elements. Consider the set $\mathbf{G}_n^r$ of all punctured matrices, $r = (1, \ldots, \binom{n}{n'})$. We define

$$\mathbf{G}_{n'}^* = \arg\min_{\mathbf{G}_{n'}^r} \text{BER}, \tag{3-1}$$

where the BER is adopted. Many works resort to exhaustive searches with an optimization method to determine $\mathbf{G}_{n'}^*$. We can consider equation 3.1 represents the general formula of the punctured $\mathbf{G}_{n'}$ matrix.

In contrast to prior work, we propose the PD puncturing technique for computing $\mathbf{p}$, where the channels with the lowest polarization indices are eliminated.

As discussed in section 2.2, we will briefly review the polarization channel calculation. The polarization channel $W_n^{(i)}$ is calculated with the Bhattacharya parameter satisfying the following recursion:

$$\begin{cases} Z(W_n^{(2i-1)}) = 2Z(W_{n/2}^{(i)}) - Z(W_{n/2}^{(i)})^2 \\ Z(W_n^{(2i)}) = Z(W_{n/2}^{(i)})^2. \end{cases} \tag{3-2}$$

Using the notation in [1], for $n = 8$ we have l stages of polarization and the $Z_0 = 0.5$, the polarization stages are

- stage 1:
  $Z(W^+) = 2(Z_0) - (Z_0)^2 = 2(0.5) - (0.5)^2 = 0.75$ and
  $Z(W^-) = (Z_0)^2 = (0.5)^2 = 0.25$
- stage 2:
  $Z(W^{++}) = 2Z(W^+) - Z(W^+)^2 = 0.9375$,
  $Z(W^{-+}) = 2Z(W^-) - Z(W^-)^2 = 0.4375$,
  $Z(W^{+-}) = Z(W^+)^2 = 0.5625$ and
  $Z(W^{--}) = Z(W^-)^2 = 0.0625$
- stage 3, by induction:
  $Z(W^{+++}) = 0.99609$,
  $Z(W^{-++}) = 0.68359$,
  $Z(W^{+-+}) = 0.80859$,
  $Z(W^{--+}) = 0.12109$,
  $Z(W^{++-}) = 0.87891$,
  $Z(W^{-+-}) = 0.19141$,

$Z(W^{+--}) = 0.31641$ and
$\mathrm{Z(W^{---}) = 0.00391.}$

The channels $(W^{+++}, W^{-++}, W^{+-+}, W^{--+}, W^{++-}, W^{-+-}, W^{+--}, W^{---})$ can be written with $(W_0, W_1, W_2, W_3, W_4, W_5, W_6, W_7)$. We define the polarization vector as:

$$\mathbf{b} \triangleq \left[\; Z(W_0); Z(W_1); \ldots; Z(W_{n-1}) \; \right]^T, \tag{3-3}$$

where the polarization vector $\mathbf{b}$ is computed through (3-2). An example for stage 3 we have $\mathbf{b} = [0.99609, 0.68359, 0.80859, 0.12109, 0.87891, 0.19141, 0.31641, 0.00391]^T$.

The key idea of the proposed PD method is to puncture in the generator matrix $\mathbf{G}$ the indices of the rows that correspond to the channels with smallest values of polarization.

Different from the method of obtaining the polarized channels originally proposed by Arikan [1], section 2.3.3, we adopt here the logarithm domain. As discussed in section 2.5, the functions g and f in the logarithm domain simplify the synthesis of polar codes in hardware [43], [44] and [45].

The punctured channels can be obtained by sorting the polarization vector $\mathbf{b}$. The goal of sorting is to determine a permutation $k(1)k(2)\ldots k(n)$ of the indices $\{1, 2, ..., n\}$ that will organize the entries of the polarization vector $\mathbf{b}$ in increasing order [8]:

$$Z(W_{k(1)}) \leq Z(W_{k(2)}) \leq \ldots \leq Z(W_{k(n)}). \tag{3-4}$$

Consider the sort function $[\mathbf{a}, \mathbf{k}] = \mathrm{sort}(\mathbf{b})$ which implements (3-4), where $\mathbf{a}$ lists the sorted $\mathbf{b}$ and $\mathbf{k}$ contains the corresponding indices of $\mathbf{a}$.

Table 3.1 show an example of the polarization vector $\mathbf{b}$ for $n = 8$, sorting vector $\mathbf{a}$ and the new index $\mathbf{k}$.

The vector $\mathbf{k} = [8, 5, 6, 2, 7, 3, 4, 1]$ contains the indices of the polarization values of the channels in increasing order, which is used to obtain the puncturing vector $\mathbf{p}$ of the proposed PD method:

$$\mathbf{p} = [k(1), \ldots, k(n - n')]. \tag{3-5}$$

Table 3.1: Polarization vector $\mathbf{b}$ for $n = 8$

$\boldsymbol{b}$	0.996	0.683	0.808	0.121	0.878	0.191	0.316	0.003
index	1	2	3	4	5	6	7	8

After sorting								
$\boldsymbol{a}$	0.003	0.878	0.191	0.683	0.316	0.808	0.121	0.996
$\boldsymbol{k}$	8	5	6	2	7	3	4	1

3.3 Pseudo-code

In Algorithm 4 we have included a pseudo-code of the proposed PD method, including details about the size reduction of the generator matrix $\mathbf{G}_n$ and the puncturing of the channels with the lowest polarization values.

Algorithm 4: Proposed PD algorithm

Data: Index each column by $\{1, 2, ..., n\}$
Data: Index each row by $\{1, 2, ..., n\}$
Input: punctured codeword with length n'
Input: $\mathbf{G}_n$
Output: $\mathbf{G}_{n'}$

```
begin
    Calculate the polarization channel vector b for n in log domain
    Calculate [a, k] = sort(b)
    Calculate the puncturing vector p = [k(1), ..., k(n − n')]
    for y ⟵ 1 : |p| do
        r_min ⟵ p(y)
        Delete row from G_n with index r_min
        Delete column from G_n with index r_min
        G_n' ⟵ G_n
    return G_n'
```

We consider now an example with punctured polar codes with length $n' = 5$. We choose the generator matrix $\mathbf{G}_8$. For the puncturing of $\mathbf{G}_8$ to $\mathbf{G}_5$, the channels with the lowest polarization rank values are W_7, W_5 and W_3, the puncturing vector is $\mathbf{p} = (8, 6, 4)$ and $|\mathbf{p}| = 3$. The 1st element of $\mathbf{p}$ is 8, which results in the deletion of the 8th column and the 8th row of $\mathbf{G}_8$. The

2nd element of $\mathbf{p}$ is 6, which requires the elimination of the 6th column and the 6th row. At last, the 3rd element of the $\mathbf{p}$ is 4, which requires the deletion of the 4th column and the 4th row. The matrix $\mathbf{G}_8$ with the indication of the deletions and the resulting generator matrix $\mathbf{G}_5$ are respectively given by

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ \cancel{1} & \cancel{1} & \cancel{1} & \cancel{1} & \cancel{0} & \cancel{0} & \cancel{0} & \cancel{0} \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ \cancel{1} & \cancel{1} & \cancel{0} & \cancel{0} & \cancel{1} & \cancel{1} & \cancel{0} & \cancel{0} \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ \cancel{1} & \cancel{1} & \cancel{1} & \cancel{1} & \cancel{1} & \cancel{1} & \cancel{1} & \cancel{1} \end{pmatrix} \rightarrow \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 \end{pmatrix}$$

The punctured codeword **c'** generated with $\mathbf{G}'_n$, which contains the bits of the binary message $\mathbf{m} = \mathbf{u}_A$ such that $Z(W_{n'}^{(i)}) \leq Z(W_{n'}^{(j)})$ for all $i \in A$, $j \in A^c$ and $\mathbf{u}_{A^c} = (u_i : i \in A^c | u_i = 0)$, is then transmitted over a channel. At the receiver, the SC decoder or another decoder can be used to obtain an estimate $\hat{\mathbf{m}}$ of the message. The encoding and decoding algorithm are the same as an original polar code.

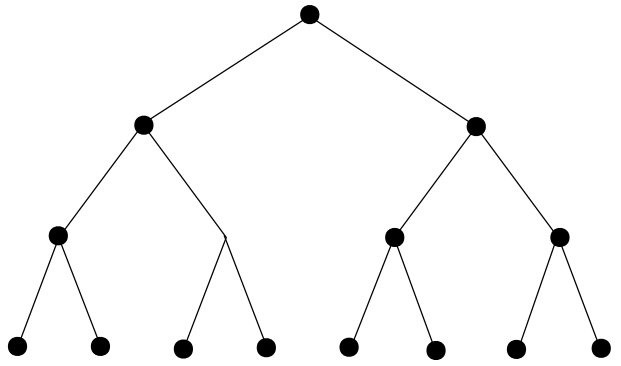
3.4 Analysis

An analysis of the Spectrum Distance (SD) for polar codes has been presented in [9] and demonstrates that systematic coding yields better BER performance than non-systematic coding with the same FER performance for both encoding schemes. However, the method used is quite costly in terms of processing time, since it generates all possible coded message combinations for a given polar code and calculates the Hamming distance between them.

An alternative metric called Polar Spectra (PS) has been studied in [5], which has a lower computational cost compared to the method in [9], and is sufficient to indicate the performance of the adopted puncturing technique. This metric is based on the channel polarization tree, the Hamming weight (HW), the complement Hamming weight (CHW) of each branch and the relation between the polarized channels W_i and the rows of the generator matrix $\mathbf{G}$. In Figure 3.1 show that the relation between the polarized channels W_i and the rows of the generator matrix $\mathbf{G}$.

An SD and a Joint Spectrum Distance (JSD) are considered as criteria

to compare puncturing techniques. The SDP and SDC are defined by the expectation of spectrum distance for path weight and spectrum distance for complementary weight, respectively.



$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{matrix} W_0 \\ W_1 \\ W_2 \\ W_3 \\ W_4 \\ W_5 \\ W_6 \\ W_7 \end{matrix}$$

8x8

W_0	W_4	W_2	W_6	W_1	W_5	W_3	W_7	
0	1	1	2	1	2	2	3	Hamming
3	2	2	1	2	1	1	0	Complementar Hamming

Figure 3.1: The relation between the polarized channels W_i and the rows of the generator matrix $\mathbf{G}$.

The SDP is given by

$$d = \sum_{(k=0)}^{l} P_1(l,k,Q)k = \sum_{(k=0)}^{l} \frac{H_n^{(k)}}{n}k, \tag{3-6}$$

where $P_1(l,k,Q) = \frac{H_n^{(k)}}{n}$ is the probability of path weight k with $|\mathbf{p}|$ bits puncturing.

The SDC is given by

$$\lambda = \sum_{(r=0)}^{l} P_0(l,r,Q)r = \sum_{(r=0)}^{l} \frac{C_n^{(r)}}{n}r, \tag{3-7}$$

where $P_0(l,k,Q) = \frac{C_n^{(k)}}{n}$. We use SDC as the main metric to evaluate the performance of the proposed and existing puncturing techniques.

The SD of the complementary Hamming weight λ, where $C_l^{(r)} = \binom{l}{r}$, l is the quantity are the levels in the polarization tree. The term $C(X) = \sum_{(r=0)}^{l} C_l^{(r)} X^r$ describes the total number of branches with a given number of zeros, or alternatively $C(X) = \sum_{(i=1:n)} X^{\mathrm{Pb_i}}$, Pb is the number of zeros of each branch.

As an example, for a $\mathbf{G}_{16}$, we have $C(X) = X^0 + 4X^1 + 6X^2 + 4X^3 + X^4$, one branch with no zero, four branches with 1 zero, six branches with 2 zeros, 4 branches with 3 zeros and one branch with 4 zeros, the $\lambda = \frac{1 \cdot 0 + 4 \cdot 1 + 6 \cdot 2 + 4 \cdot 3 + 1 \cdot 4}{16} = 2$.

Given a puncture pattern, the final metric of the SD is given by λ where the $C(X)$ is updated by removing the branches cut by puncturing, each branch corresponds to a channel, which in turn corresponds to a row (and column) in the generator matrix $\mathbf{G}$.

To obtain $\mathbf{G}_{12}$ from $\mathbf{G}_{16}$ with $\mathbf{p} = (14, 15, 16)$, updating $C(X) = 2X^1 + 5X^2 + 4X^3 + 1X^4$ and new $\lambda = \frac{2 \cdot 1 + 5 \cdot 2 + 4 \cdot 3 + 1 \cdot 4}{16} = 1.75$, always less than the previous value λ.

In Figure 3.2 show the performance of SDC for $n = 512$, comparing the values for each puncturing techniques. Note that for some values of n', in the case of the figure in question, $470 < n' < 512$, the PS indicates that the proposed PD method has lower performance than the method proposed by Niu [5]. For all other values of n', PS indicates a better performance of the DP method.

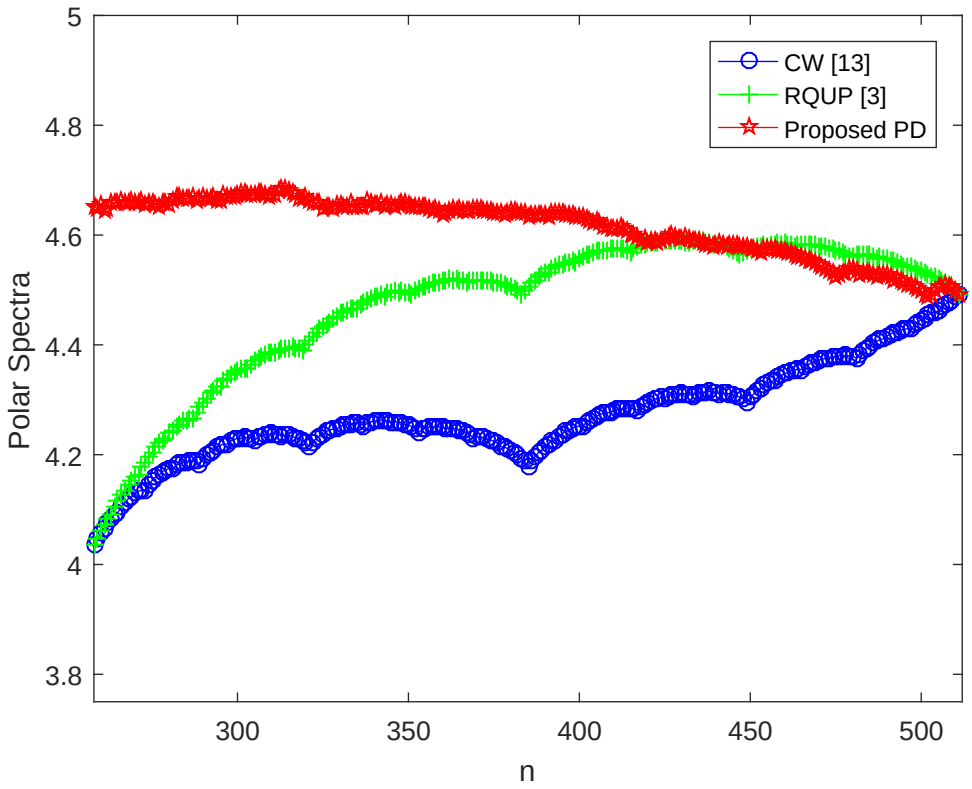


Figure 3.2: Polar Spectra SDC for various puncturing techniques.

The set of punctured branches have different weights for each puncturing

method, existing puncturing ($\mathbf{p}_{\text{ep}}$) such that $\mathbf{p}_{\text{PD}} \neq \mathbf{p}_{\text{ep}}$, $|\mathbf{p}_{\text{PD}}| = |\mathbf{p}_{\text{ep}}| = \text{y} > 0$, $\exists$ y we have

$$\sum_{i=1}^{\text{y}} C_{\text{PD(i)}}(X) < \sum_{i=1}^{\text{y}} C_{\text{CW(i)}}(X). \tag{3-8}$$

This inequality is not valid for small values of y, where the channels punctured by either method will be the same. However, there is a value of y from which the punctured channels will be different. Expanding the above equation, if we assume one chosen branch of the PD set different from the CW set, we have

$$\sum_{i=1}^{\text{y}-1} C_{\text{PD(i)}}(X) + \alpha X^{\text{y}} < \sum_{i=1}^{\text{y}-1} C_{\text{CW(i)}}(X) + \beta X^{\text{y}}, \tag{3-9}$$

where α and β are integer numbers. We then exploit the fact that $\alpha < \beta$, which yields

$$\alpha X^{\text{y}} < \beta X^{\text{y}}, \tag{3-10}$$

proving (3-8). Figure 3.3 shows the percentage of coincidence of punctured channels of different compared methods, this difference of punctured channels directly influences the performance of the proposed PD technique.

We observed that Proposed PD and RQUP have a larger coincidence curve when compared to the coincidence curve between proposed PD and CW. This fact reflects on the performance of each method, as will be seen in the next section.

3.5 Simulation

In this section, we present simulation results of rate-compatible polar codes with puncturing and a system equipped with the SC decoder, as described for the IoT and the eMBB 5G scenarios [3] and [4], which require the use of short to moderate block lengths. We measure the BER against the

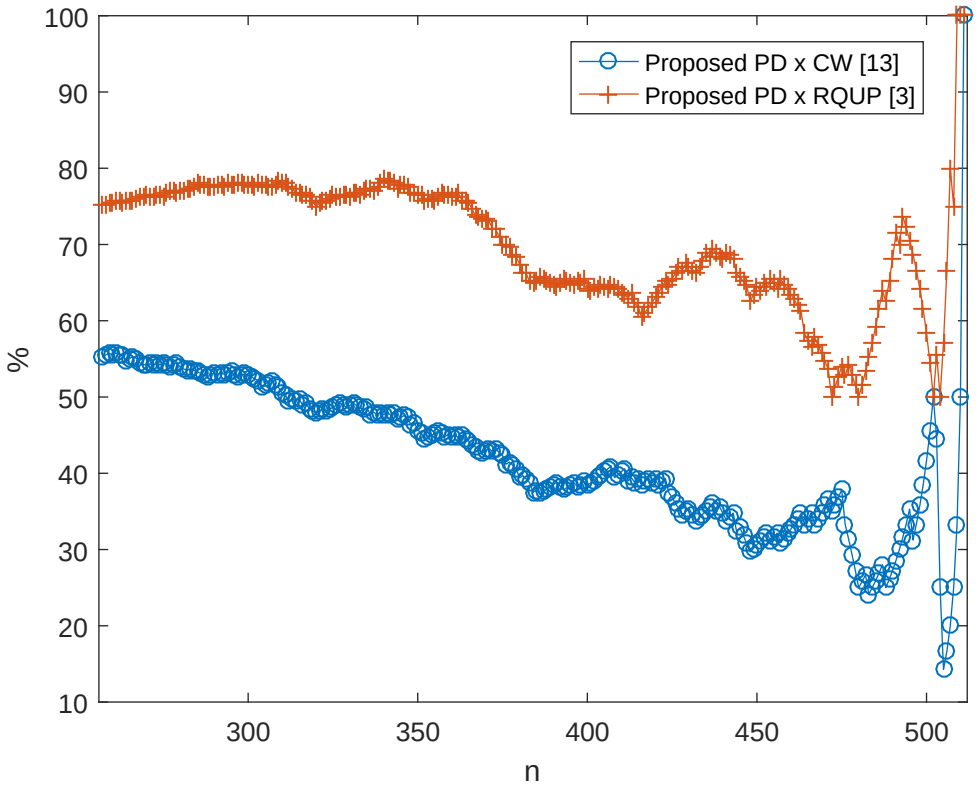


Figure 3.3: Percentage of coinciding punctured entries of puncturing techniques.

signal-to-noise ratio, defined as the ratio of the bit energy, E_b, and the power spectral density, N_0, in dB. The algorithm for polar codes uses the Tal-Vardy method [2] and Hof's technique [7]. We use 500,000 interactions for each graph.

In the first example, for M2M scenario with $n = 128$, $n' = 100$ and $k = 64$. In particular, we have reduced the $\mathbf{G}_{128}$ matrix to the $\mathbf{G}_{100}$ matrix using the proposed PD puncturing technique. The results in Figure 3.4 show that the proposed PD technique outperforms the RQUP and the CW techniques by up to 0.15 dB for the same BER performance, and approaches the performance of the mother code (MC) with $n = 128$ and $k = 64$.

In the second example, we consider a IoT scenario with $n = 512$, $n' = 480$ and $k = 256$, we have reduced the $\mathbf{G}_{512}$ matrix to the $\mathbf{G}_{480}$ matrix using the proposed PD puncturing technique. For comparison purposes, we have also included the curves associated with the existing methods CW and RQUP, as can be observed in Figure 3.5. The results in figure 3.5 show that the proposed PD technique outperforms the RQUP and the CW techniques by up to 0.25 dB for the same BER performance, and approaches the performance of the mother code (MC) with $n = 512$ and $k = 256$.

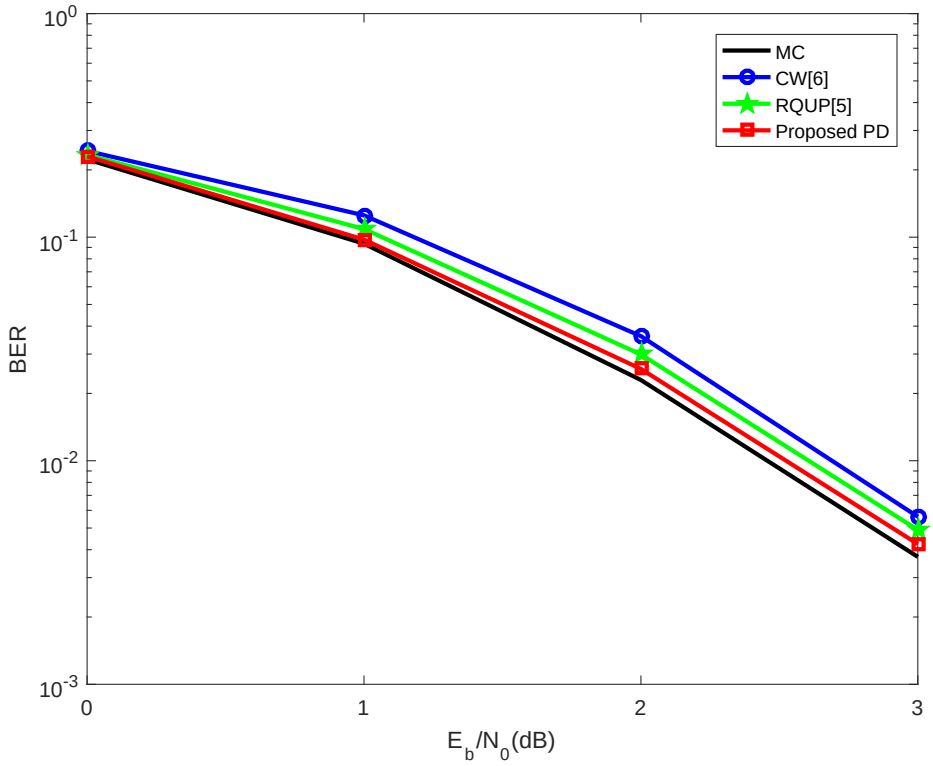


Figure 3.4: Comparative BER performace of puncturing polar code n'=100 k=64.

For eMBB scenario with $n = 2048$, $n' = 1920$ and $k = 1600$. The results in Figure 3.6 show that the proposed PD technique outperforms the RQUP and the CW techniques by up to 0.25 dB for the same BER performance, and approaches the performance of the mother code (MC) with $n = 2048$ and $k = 1600$.

We observed that in all the simulated scenarios the performance obtained by the proposed technique PD is better than those proposed by CW and RQUP. In the Table 3.2 we compare for each simulation the PS values obtained for each curve.

Note that the SDC index of the proposed model PD has a higher value than the CW and RPCT techniques, and that of the RPCT method is higher than the CW model. This corresponds to the values observed in the simulation. The SDC index gives us an indication of performance of the chosen puncturing method.

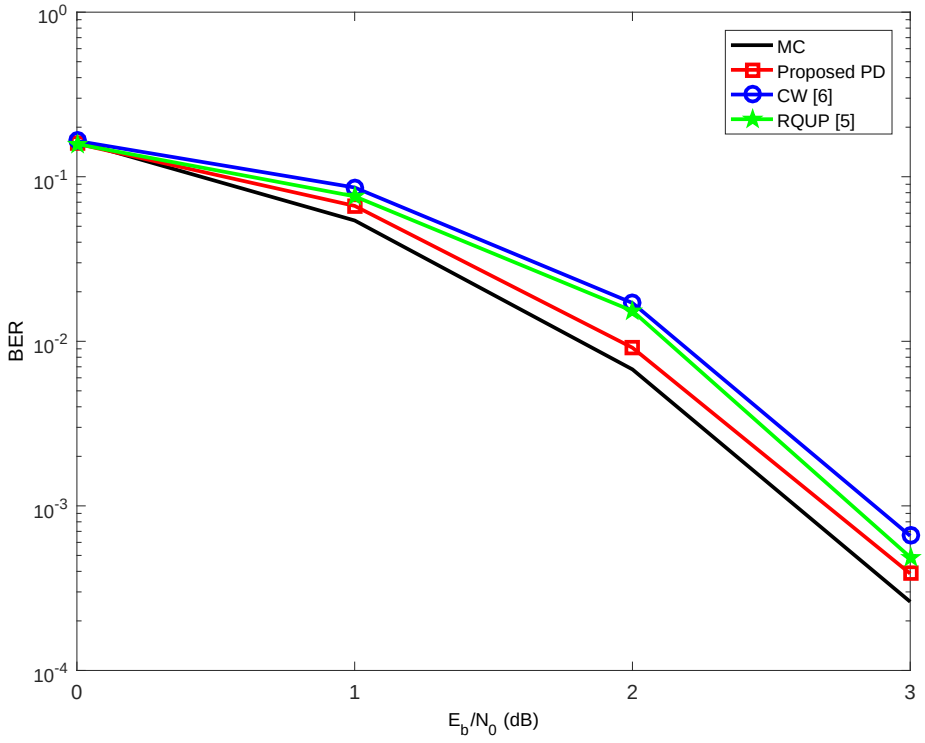


Figure 3.5: Comparative BER performace of puncturing polar code n'=480 k=256.

Table 3.2: Polar Spectra for Figures 3.4, 3.5 and 3.6

SDC	Proposed PD	RQUP [5]	CW [6]
Fig.3.4	3.73	3.46	3.45
Fig.3.5	4.62	4.46	4.43
Fig.3.6	5.62	5.44	5.43

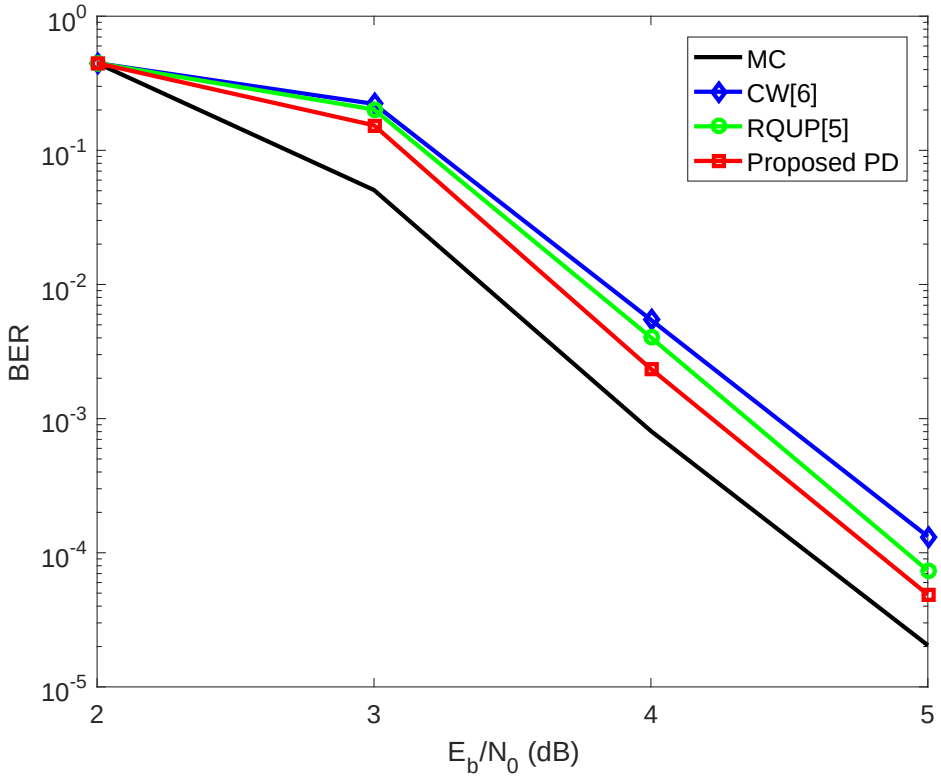


Figure 3.6: Comparative BER performace of puncturing polar code n'=1920 k=1600.

4
Conclusions

In this dissertation, we have proposed a puncturing method for polar codes based on the channel polarization. We have compared the performance of the proposed method with different puncturing techniques for polar codes with successive cancellation decoding. We have also reviewed the fundamentals of polar codes, encoding algorithms, SC, SCL and CA-SCL decoding techniques. We have then described how to construct and puncture polar codes and analysed their performance under the AWGN channel. Our motivation is that currently polar codes are being considered for 5G systems.

After analyzing the results of our simulations, we can see that the proposed method performs better than the other analyzed puncturing methods for the 5G scenarios investigated. For the punctured polar codes, with $n' = 1920$ and r = 1600, with $n' = 480$ and k = 256, and with $n' = 100$ and k = 64; the proposed PD method has better performance than the CW and RQUP methods. We show that Polar Spectra can be used as a performance evaluation metric for puncturing technique.

Future research work may focus on the study of extension techniques for polar code, on the search for new decoder architectures and strategies, such as sequential decoders, sphere decoders and adaptive list decoders, some of which promise better performance and lower complexity; and latency reduction. Another very interesting topic is the expansion of Polar Spectra to include the effect of the rate on the performance of the puncturing technique.

Bibliography

[1] ARIKAN, E.. **Channel Polarization: A Method for Constructing Capacity Achieving Codes for Symmetric Binary-Input Memoryless Channels**. IEEE Transactions on Information Theory, 55:3051–3073, 2009.

[2] TAL, I.; VARDY, A.. **How to construct polar codes**. IEEE Transactions on Information Theory, 59:6562–6582, 2013.

[3] OSSEIRAN, A.; BOCCARDI, F.; BRAUN, V.; KUSUME, K.; MARSCH, P.; MATERNUA, M.; QUESETH, O.; SCHELLMANN, M.; SCHOTTEN, H.; TAOKA, H.; TULLBERG, H.; UUSITALO, M. A.; TIMUS, B.; FALLGREN, M.. **Scenarios for 5G Mobile and Wireless Communications: The Vision of the METIS Project**. IEEE Communications Magazine, 52:26–35, 2014.

[4] FERREIRA, J. S.; RODRIGUES, H. D.; GONZALEZ, A. A.; NIMR, A.; NATTHÉ, M.; ZHANG, D.; MENDES, L. L.; FETTWEIS, G.. **GFDM Frame Design for 5G Application Scenarios**. Journal of Communication and Information Systems, 32:54–61, 2017.

[5] NIU, K.; DAI, J.; CHEN, K.; LIN, J. ZHANG, Q. T.; VASILAKOS, A. V.. **Rate-Compatible Punctured Polar Codes: Optimal Construction Based on Polar Spectra**. CORNELL UNIVERSITY LIBRARY. CORNELL UNIVERSITY, 2017. https://arxiv.org/pdf/1612.01352.pdf, acesso em: dezember of 2017.

[6] WANG, R.; LIU, R.. **A novel puncturing scheme for polar codes**. IEEE Communications Letters, 18:2081–2084, 2014.

[7] HOF, E.. **Sliced Polar Codes**. IEEE Asia Pacific Conference on Circuits and Systems (APCCAS), 1:476–479, 2016.

[8] KNUTH, D.. **The Art of Computer Programming Vol.3**. Addison Wesley, New York, 3nd edition, 1998.

[9] LIU, Z.; CHEN, K.; NIU, K.; HE, Z.. **Distance spectrum analysis of polar codes**. IEEE Wireless Communications and Networking Conference (WCNC), 1:490–495, 2014.

[10] MORI, R.; TANAKA, T.. **Performance and construction of polar codes on symmetric binary-input memoryless channels**. IEEE Int. Sym. on Inform. Theory (ISIT), 1:1496–1500, 2009.

[11] TRIFONOV, P.. **Efficient design and decoding of polar codes**. IEEE Transactions on Communications, 60:3221–3227, 2012.

[12] ARIKAN, E.. **Systematic polar coding**. IEEE Communications Letters, 15:860–862, 2011.

[13] FOSSORIER, M. P. C.; MIHALJEVIC, M.; IMAI, H.. **Reduced complexity iterative decoding of low-density parity check codes based on belief propagation**. IEEE Transactions on Communications, 47:673–680, 1999.

[14] LEROUX, C.; TAL, I.; VARDY, A.; GROSS, W. J.. **Hardware architectures for successive cancellation decoding of polar codes**. IEEE Int. Conf. on Acou. Speech and Signal Process. (ICASSP), 1:1665–1668, 2011.

[15] TAL, I.; VARDY, A.. **List decoding of polar codes**. IEEE Int. Sym. Inf. Theory (ISIT), 1:1–5, 2011.

[16] B. LI, H. SHEN, AND D. TSE. **An adaptive successive cancellation list decoder for polar codes with cyclic redundancy check**. IEEE Communications Letters, 16:2044–2047, 2012.

[17] SARKIS, G.; GIARD, P.; VARDY, A.; THIBEAULT, C.; GROSS, W. J.. **Increasing the speed of polar list decoders**. Proc. Sig. Process. Sys. (SiPs) IEEE Workshop, 16:1–6, 2014.

[18] BALATSOUKAS-STIMMING, A.; PARIZI, M.; BURG, A.. **LLR-based successive cancellation list decoding of polar codes**. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP), 1:3903–3907, 2014.

[19] CAO, C.; FEI, Z.; YUAN, J.; KUANG, J.. **Low complexity list successive cancellation decoding of polar codes**. IET Commun., 17:3145–3149, 2014.

[20] LIN, B.; SHEN, H.; TSE, D.; TONG, W.. **Low-latency polar codes via hybrid decoding**. 8th Int. Symp. Turbo Codes and Iterative Information Process, 1:223—227, 2014.

[21] VANGALA, H.; VITERBO, E.; HONG, Y.. **Permuted successive cancellation decoder for polar codes**. Int. Symp. Inf. Theory and Its App. (ISITA), 1:438–442, 2014.

[22] VANGALA, H.; VITERBO, E.; HONG, Y.. **Improved multiple folded successive cancellation decoder for polar codes**. General Assembly and Scientific Symposium, 1:1–4, 2014.

[23] KAHRAMAN, S.; VITERBO, E.; ÇELEBI, M.. **Multiple folding for successive cancellation decoding of polar codes**. IEEE Commun. Lett., 5:545–548, 2014.

[24] KAHRAMAN, S.; VITERBO, E.; ÇELEBI, M.. **Folded successive cancellation decoding of polar codes**. Commun. Theory Workshop, 5:57–61, 2014.

[25] LI, Y.; ALHUSSIEN, H.; HARATSCH, E.; JIANG, A.. **A study of polar codes for MLC NAND flash memories**. Int. Conf. on Comput., Netw. and Commun. (ICNC), 1:608–612, 2015.

[26] G. SARKIS, I. TAL, P. GIARD, A. VARDY, C. THIBEAULT, AND W. J. GROSS,. **Flexible and low complexity encoding and decoding of systematic polar codes**. IEEE Trans. Commun, 64:2732—-2745, 2016.

[27] HONG, S. N.; HUI, D.; MARIC, I.. **Capacity-achieving rate-compatible polar codes**. IEEE Transactions on Information Theory, 1:7620–7632, 2017.

[28] SABER, H.; MARSLAND, I.. **An incremental redundancy Hybrid ARQ scheme via puncturing and extending of polar codes**. IEEE Transactions on Communications, 63:3964–3973, 2015.

[29] CHEN, K.; NIU, K.; LIN, J.. **A hybrid ARQ scheme based on polar codes**. IEEE Communications Letters, 17:1996–1999, 2013.

[30] NIU, K.; CHEN, K.; LIN, J.. **Beyond turbo codes: rate-compatible punctured polar codes**. EEE International Conference on Communications (ICC), 17:3423–3427, 2013.

[31] SHIN, D.M.; LIM, S.C.; YANG, K.. **Design of Length-Compatible Polar Codes Based on the Reduction of Polarizing Matrices**. IEEE Transactions on Communications, 61:2593–2599, 2013.

[32] ESLAMI, A.; PISHRO-NIK, H.. **A practical approach to polar codes**. IEEE International Symposium on Information Theory Proceedings (ISIT), 1:16–20, 2011.

[33] ESLAMI, A.; PISHRO-NIK, H.. **On finite-length performance of polar codes: stopping sets, error floor, and concatenated design**. IEEE Transactions on Communications, 61:919–929, 2013.

[34] KIM, J.; KIM, J.H.; KIM, S.H.. **An Efficient Search on Puncturing Patterns for Short Polar Codes**. International Conference on Information and Communication Technology Convergence (ICTC), 1:182–184, 2015.

[35] ZHANG, L.; ZHAN, Z.; WANG, X.; YU, Q.; CHEN, Y.. **On the Puncturing Patterns for Punctured Polar Codes**. IEEE International Symposium on Information Theory, 1:121–125, 2014.

[36] WANG, R.; LIU, R.. **A novel puncturing scheme for polar codes**. IEEE Communications Letters, 18:2081–2084, 2014.

[37] MILOSLAVSKAYA, V.. **Shortened polar codes**. IEEE Transactions on Information Theory, 61:4852–4865, 2015.

[38] LIN, S.; COSTELLO, D.. **Error Control Coding**. Pearson, New York, 2nd edition, 2004.

[39] UCHOA, A. G. D.; HEALY, C. T.; LAMARE, R. C. DE. **Structured Root-Check LDPC Codes and PEG-Based Techniques for Block-Fading Channels**. EURASIP Journal on Wireless Communications and Networking, 1:1–20, 2015.

[40] HEALY, C. T.; LAMARE, R. C. DE. **Design of LDPC Codes Based on Multipath EMD Strategies for Progressive Edge Growth**. IEEE Transactions on Communications, 64:3208–3219, 2016.

[41] HEALY, C. T.; LAMARE, R. C. DE. **Knowledge-aided informed dynamic scheduling for LDPC decoding**. IEEE Int. Conf. Commun. Workshop (ICCW), 1:2212–2217, 2015.

[42] KAI NIU; KAI CHEN. **CRC-Aided Decoding of Polar Codes**. IEEE Communications Letters, 16:1668–1671, 2012.

[43] SARKIS, G.; GIARD, P.; VARDY, A.; THIBEAULT, C.; GROSS, W. J.. **Fast Polar Decoders: Algorithm and Implementation**. IEEE Journal on Selected Areas in Comunication, 32:946–957, 2014.

[44] LIN, J.; XIONG, C.; YAN, Z.. **A High Throughput List Decoder Architecture for Polar Codes**. IEEE Transactions on Very Scale Integration (VLSI) Systems, 24:2378–2391, 2016.

[45] DIZDAR, O.; ARIKAN, E.. **A High-Throughput Energy-Efficient Implementation of Successive Cancellation Decoder for Polar Codes Using Combinational Logic**. IEEE Transactions on Circuits and Systems-I: Regular Papers, 63:436–447, 2016.

[46] MISHRA, A.; RAYMOND, A. J.; AMARU, L. G.; SARKIS, G.; LEROUX, C.; MEINERZHAGEN, P.; BURG, A.; GROSS, W. J.. **A successive cancellation decoder ASIC for a 1024-bit polar code in 180nm CMOS**. IEEE Asian Solid State Circuits Conference (A-SSCC), 1:205–208, 2012.

[47] ANDREWS, J. G.; BUZZI, S.; CHOI, W.; HANLY, S. V.; LOZANO, A.; SOONG,A. C. K.; ZHANG, J. C.. **What Will 5G Be?** IEEE Journal on Selected Areas in Communications, 32:1065–1082, 2014.

[48] CHEN, S.; ZHAO, J.. **The Requirements, Challenges, and Technologies for 5G of Terrestrial Mobile Telecommunication**. IEEE Communications Magazine, 52:36–43, 2014.

[49] DEMESTICHAS, P.; GEORGAKOPOULOS, A.; KARVOUNAS, D.; TSAGKARIS, K.; STAVROULAKI, V.; LU, J.; XIONG, C.; YAO, J.. **5G on the Horizon: Key Challenges for the Radio-Access Network**. IEEE Vehicular Technology Magazine, 8:47–53, 2013.

[50] MCC SUPPORT. **Final Report of 3GPP TSG RAN WG1 87**. 3GPP 3GPP, 2017. Acesso em: Novenbro de 2017.

[51] WASSERMAN, D. R.. **Designing polar codes to minimize the BER of CRC-aided list decoding**. Information Theory and Applications Workshop (ITA), 1:1–3, 2016.

[52] ZHANG, Q.; LIU, A.; PAN, X.; PAN, K.. **CRC Code Design for List Decoding of Polar Codes**. IEEE Communications Letters, 21:1229–1232, 2017.

[53] MURATA, T.; OCHIAI, H.. **On design of CRC codes for polar codes with successive cancellation list decoding**. IEEE International Symposium on Information Theory (ISIT), 1:1868–1872, 2017.