EFFICIENT DECODING OF POLAR CODES

ALIA AHMED ELETRI ANDI

JUNE 2019

EFFICIENT DECODING OF POLAR CODES

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED
SCIENCES OF
ÇANKAYA UNIVERSITY

BY
ALIA AHMED ELETRI ANDI

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE
DEGREE OF
DOCTOR OF PHILOSOPHY
IN
THE DEPARTMENT OF
ELECTRONIC AND COMMUNICATION ENGINEERING

JUNE 2019

Title of the Thesis: Efficient Decoding of Polar Codes

Submitted by **ALIA AHMED ELETRI ANDI**

Approval of the Graduate School of Natural and Applied Sciences, Çankaya University.

Prof. Dr. Can ÇOĞUN
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Doctor of Philosophy.

Dr. Lecturer Özgür ERGÜL
Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Doctorate of Philosophy.

Assoc. Prof. Dr. Orhan GAZI
Supervisor

Examination Date: 18.06.2019

Examining Committee Members

Prof. Dr. Yahya Kemal BAYKAL	(Çankaya Univ.)
Prof. Dr. Klaus SCHMIDT	(METU)
Assoc. Prof. Dr. Orhan GAZİ	(Çankaya Univ.)
Assoc. Prof. Dr. Özgür ERTUĞ	(Gazi Univ.)
Dr. Lecturer Özgür ERGÜL	(Çankaya Univ.)

STATEMENT OF NON-PLAGIARISM PAGE

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name	:	Alia ANDI
Signature	:	
Date	:	18.06.2019

ABSTRACT

EFFICIENT DECODING OF POLAR CODES

ANDI, Alia Ahmed Eletri
Ph.D., Department of Electronic and Communication Engineering
Supervisor: Assoc. Prof. Dr. Orhan GAZİ

June 2019, 89 pages

Polar Codes are the first mathematically provable capacity achieving error correcting codes which have low complexity encoding and decoding algorithms. For the decoding of polar codes, as a preliminary decoding algorithm, the successive cancellation (SC) decoding algorithm is used. SC algorithm is a sequential decoding algorithm which suffers from error propagation. For this reason, SC algorithm does not show good performance for moderate codeword lengths.

Polar codes with SC decoding show worse performance than that of the modern channel codes, such as LDPC and turbo codes. To improve the performances of the polar codes improved versions of SC algorithm such as SC list (SCL) and SC stack are introduced in the literature, and these algorithms show much better performance than that of the classical SC decoding algorithm although they have larger complexity compared to SC. Besides, cyclic redundancy check codes are concatenated with polar codes which are decoded using

the SCL algorithm, and such a concatenated system shows better performance than the other modern channel codes.

In this thesis, we first propose a tree structure for the successive cancelation (SC) decoding of polar codes. The proposed structure is easy to implement in hardware and suitable for parallel processing operations. Next, using the proposed tree structure, we propose a technique for the fast decoding of polar codes. With the proposed method, it is possible to decode all the information bits simultaneously at the same time, i.e., in parallel. Lastly, we introduce and improved version of the proposed high-speed decoding algorithm. The proposed high-speed decoding approach and its improved version are simulated on computer environment, and their BER performances are compared to the performance of the classical successive cancelation method.

Furthermore, we introduce a new approach to the successive cancelation of polar codes. The proposed approach uses the soft likelihood ratios of the predecessor information bits for the determination of successor information bits. The proposed method can be considered for the construction of joint iterative communication systems exchanging soft likelihoods. It is shown that the proposed soft decoding approach shows better performance than the classical successive cancelation algorithm introduced in Arikan’s original work.

As we know, polar codes are decoded in a sequential manner using successive cancelation algorithm introduced by Arikan. The sequential nature of the decoding process suffers from error propagation. We inspect the effects of error propagation on the performance of polar codes and propose some methods to alleviate the degrading effects of error propagation on the code performance for short and long frame lengths.

Keywords: Polar codes, recursive decoding, fast decoding, successive cancelation decoding, successive cancelation algorithm, soft decoding, sequential decoding, successive cancelation algorithm, error propagation, BEC.

ÖZ

KUTUP KODLARININ VERİMLİ ÇÖZÜMLENMESİ

ANDI, Alia Ahmed Eletri

Doktora, Elektronik ve Haberleşme Mühendisliği

Tez Yöneticisi: Doç. Dr. Orhan GAZİ

Haziran 2019, 89 sayfa

Kapasiteye erişimi ilk defa matematiksel olarak ispatlabilen Polar Kodlar, düşük karmaşıklıkta olan ardışık giderim (SC) yöntemi ile ikili ayrık hafızaya sahip olmayan simetrik kanallar için sunulmuş hata düzeltme kodlarıdırlar. Her ne kadar SC düşük bir karmaşıklığa sahip bir algoritma olsa da, hata yayılması probleminden dolayı iyi performans gösterememektedir.

Ne yazık ki, SC kod çözme işleminin sonlu çerçeve uzunluklarındaki hata düzeltme performansı, LDPC kodları gibi diğer modern kodlarınki kadar iyi değildir. Sonlu çerçeve uzunluğu performansını iyileştirmek için, SC liste (SCL) kod çözme ve SC yığın kod çözme gibi daha gelişmiş algoritmalar yakın zamanda tanıtılmıştır. Bu algoritmalar, temel kod çözücü olarak SC'yi kullanır, ancak aynı anda birden çok yolu keşfederek bir aday kod kelimesi sonuçlanacak şekilde performansını arttırır. SCL çözücüsünü kod çözme işleminin hesaplama ve bellek karmaşıklıkları, basit SC kod çözücüsünden çok daha yüksektir. Kod çözme algoritmasının performansını arttırmak için döngüsel artıklık kontrolü (CRC) yardımı ile SCL yapısı (CRC-SCL) kullanılabilir.

Bu tezde, öncelikle kutup kodlarının kod çözme işlemlerini ardışık giderim (SC) algoritması ile çözmek için bir ağaç yapısı öneriyoruz. Önerilen yapının donanım üzerinde gerçeklenmesi kolaydır ve paralel işleme işlemleri için uygundur. Daha sonra, önerilen

ağaç yapısını kullanarak, kutup kodlarının hızlı bir şekilde çözülmesi için bir teknik öneriyoruz. Önerilen teknik ile tüm bilgi bitlerinin aynı anda, yani paralel olarak çözülmesi mümkündür. Son olarak, önerilen kod çözme algoritması hızını arttıracak bir yöntem sunulmuştur. Önerilen yüksek hızlı kod çözme yaklaşımın ve geliştirilmiş versiyonun, bilgisayar ortamında benzetimi yapılmış ve bit-hata oranı (BER) performansları, klasik ardışık giderim yönteminin performansıyla karşılaştırılmıştır.

Ayrıca, kutup kodlarının art arda canlandırılmasına yeni bir yaklaşım getiriyoruz. Önerilen yaklaşım, ardışık bilgi bitlerinin belirlenmesi için önceki bilgi bitlerinin yumuşak olasılık oranlarını kullanmaktadır. Önerilen yöntem, yumuşak olasılıkları paylaşan ortak yinelemeli iletişim sistemlerinin kurulabilmesi için düşünülebilir. Önerilen yumuşak kod çözme yaklaşımının, Arıkan'ın orijinal eserinde tanıtılan klasik ardışık giderim algoritmasından daha iyi performans gösterdiği gösterilmiştir.

Bildiğimiz gibi, polar kodları Arıkan'ın orijinal eserinde tanıtılan ardışık giderim algoritması kullanılarak sıralı bir şekilde çözülür. Kod çözme işleminin sıralı yapısı, hata yayılımından mustariptir. Bu çalışmada, hata yayılımının kutupsal kodların performansı üzerindeki etkilerini inceliyoruz ve kısa ve uzun veri blokları için hata yayılımının kod performansı üzerindeki düşürücü etkilerini azaltmak için yöntemler öneriyoruz.

Anahtar Kelimeler: Kutup kodları, özyinelemeli kod çözme, hızlı kod çözme, ardışık giderim kod çözücüsü, ardışık giderim algoritması, yumuşak kod çözme, sıralı kod çözme, hata yayılımı, ikili silme kanalı.

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to Assoc. Prof. Dr. Orhan Gazi for his supervision, special guidance, suggestions, and encouragement through the development of this thesis.

I would like to express my truthful respect for the government of Libya represented by the Ministry of Higher Education and Scientific Research for their financial support to complete my study.

TABLE OF CONTENTS

LIST OF FIGURES

FIGURES

LIST OF TABLES

TABLES

LIST OF ABBREVIATIONS

SC	Successive Cancellation
BEC	Binary Erasure Channel
AWGN	Additive White Gaussian Noise
SCL	Successive Cancellation List
DMC	Discrete Memoryless Channel
FER	Forward Error Rate
BER	Bit Error Rate
BLER	Block Error Rate
SNR	Signal-to-Noise Ratio
SCS	Successive Cancellation Stack
LDPC	Low density parity check code
B-DMC	Binary- Discrete Memoryless Channel
BCJR	Bahl-Cocke-Jelinek-Raviv
CRC	Cyclic Redundancy Check
LR	Likelihood Ratio
CD	Compact Disc
DVD	Digital Video Disc
ML	Maximum Likelihood
XOR	Exclusive OR
LLR	Log-Likelihood Ratio
BP	Belief Propagation

BCH	Bose-Chaudhuri-Hocquenghem
BMS	Binary input Memoryless Symmetric
BSC	Binary Symmetric Channel

CHAPTER 1

INTRODUCTION

1.1 Background

Channel coding can be considered as one of the most important topics of digital communication. Channel coding is used in every place of the internet and mobile communications. Channel coding is used to correct the transmission errors taking pace during the data transmission. It is also widely used in optical communication and data storage.

The main topic of information theory is related to transmission of data in a noisy channel. Redundancy is added to the data before the transmission operation to make the communication reliable in the presence of noise. The recipient has access only to a noisy version of the data. If sufficient amount of redundancy is added in a logical manner, then it is possible restore the original data in the receiver. Coding is nothing but generating the redundancy in an intellectual way. Encoding is performed to generate the required redundancy, and the encoded information is passed through a noisy channel. Shannon in its paper [1] defined the reliable communication limits and he provided a mathematical basis for systematically studying the problems that led to success over the past 50 years. The commonality of his approach allows us to study even modern scenarios, such as mobile communications. The basic model that Shannon refers to consists of a source that generates information, a receiver that receives information, and a channel that models the physical transmission of information.

Shannon define the entropy related to a random variable, and the entropy indicates the average amount of information carried by the random variable and it is demoted by $H\ (X)$. The entropy is used to drive the limits of data compression and reliable transmission for a given signal to noise ratio (SNR). The communication channel can be indicated by a conditional probability distribution. Let X and Y denote the input and output alphabet of a communication channel. The channel indicated as $W : X \rightarrow Y$ is characterized by the conditional probability distribution $W\ (y \mid x)$. When x is transmitted through a channel, at the receiver the symbol $y \in Y$ which is an element of another random variable is received, and between output and input random variables we can define the conditional probability distributed function $W\ (y \mid x)$. Shannon showed that it is possible to decrease the transmission error probability to zero asymptotically adding sufficient amount of redundancy to the data to be transmitted. He defined the capacity of the channel $I\ (W)$, which characterizes the maximum possible speed of the reliable transmission. In other words, for any $R < I\ (W)$ it is possible to transmit R bits through a noisy channel with zero error probabilities. It is also possible to deduce that roughly we can say that if the source entropy is less than the channel capacity, i.e. if $H\ (X) < I\ (W)$, then it is possible to do reliably communication through a noisy channel. Another critical result of Shannon is that if the entropy of the source is greater than the capacity of the channel, then reliable communication is impossible.

The second important part of Shannon's work is to separate the source and channel coding operations, as shown in Figure 1, without loss of performance. The source coding module in Figure 1 tries to decrease the redundant information available in the data to be transmitted, this is also called as data compression or source coding On the other hand, channel coding part increases the redundancy amount for reliable data transfer. At the decoder side, first channel decoding operation is performed to recover the original data bits transmitted. Next source decoding, i.e., de-compression, operation is performed. To achieve the theoretical limits of the source and channel coding operations, we should choose the block length of the data to be transmitted large enough. This, in turn, affects the complexity. For practical use of the source and channel codes, we need to pay attention to the computational complexity of the encoder and decoder units so that that can be implemented in electronic circuits. Otherwise, their use in practical systems will be limited.

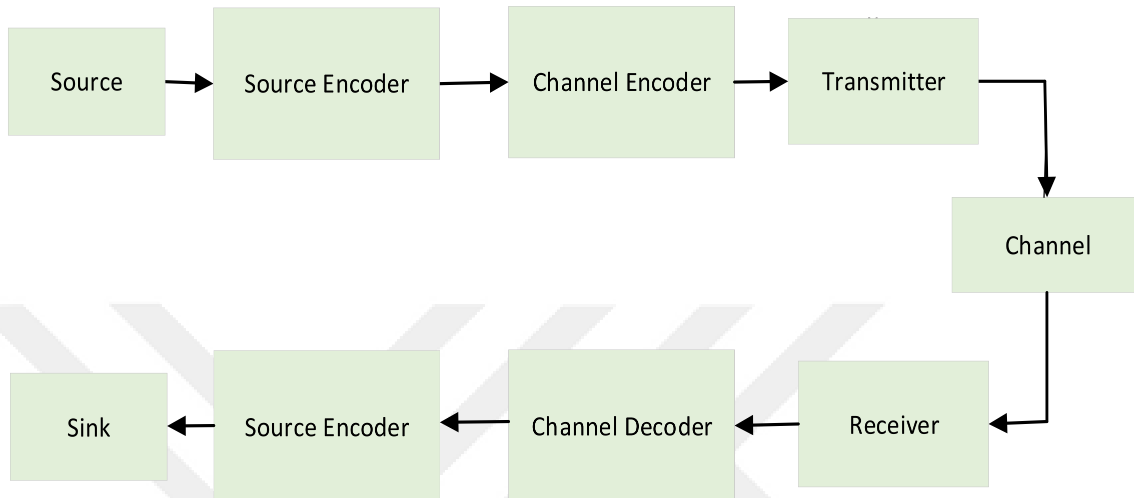

Figure 1 Primary feature of a digital communication system.

1.2 Polar Codes

Polar codes, which are invented by Arikan, can be considered as the first class of mathematically proven channel codes capable of achieving channel with low complexity of encoding and decoding [32]. Polar coding operation is based on the phenomenon of polarization, which is the main principle of polar codes, can be used for many problems of source and channel coding for both single-user and multi-user communication scenarios.

The computation complexity for the encoding and decoding operations of the polar codes can be expressed using $O\ (N\ logN)$ where N is the length of information sequence. The polar code logic can be briefly summarized for binary discrete memoryless channels (DMCs) as follows: (1) For binary DMC, polar encoding is performed such that virtual channels are created between input and output bits and the capacities of these virtually created channels approaches either to '1' or '0' as the length of the information sequence goes to infinity, and this phenomenon is called channel polarization. (2) The information bits are transmitted through high capacity virtual channels to achieve the limits of Shannon.

1.2.1 Successive Cancellation (SC) Decoding

In [32], it was shown that polar codes with SC decoding can achieve the capacity of binary discrete memoryless channels. SC decoding operation fully utilize the polar coding property and it has much less computational complexity when compared to its counterpart belief propagation (BP) decoding algorithm which utilizes the soft information for decoding operations. SC and its improved version SC list are the most popular algorithms used for the decoding of polar codes. However, the inherent sequential decoding nature of the SC algorithm result in long decoding latencies. For a codeword of length N, the total decoding latency of the SC algorithm equals to $2N - 2$ clock cycles. This huge latency created a serious problem for the application of polar codes in real time applications. We will look at SC decoding in detail in Chapter 4.

Currently, researchers are trying to improve the decoding performance of polar codes. There are many improved decoding methods for SC like algorithm in the literature, such as are SCL, SCS which can be considered as two enhanced versions of traditional SC decoding.

1.2.2 Successive Cancellation List (SCL) Decoding

The improved version of the SC algorithm, i.e., SC list algorithm, is proposed to improve the performance of the SC decoding for short and moderate block lengths in [36]. In SCL decoding operation L best decoding paths are simultaneously tracked in contrast to SC decoding operation in which only a single path is tracked. If L is chosen a large number, then significant improvement in performance is achieved when classical SC decoder's performance is considered. SCL is a search algorithm which use a code tree structure with a search depth L. At each level of the SCL algorithm, the number of candidate paths are doubled which corresponds to bit 0 or bit 1. Then SCL selects the best L paths with the highest score metrics and stores them to process for the incoming level. SCL with L decoding paths each having a complexity of $O\ (N)$, has the total complexity $O\ (LN)$.

For the decoding operation at each level, each of the L candidates is saved and these L paths are expanded to new paths and L of them are saved again; there operations have a complexity of $O\ (LN)$ calculations. Besides, considering that the code tree consists of N levels, the direct implementation of the SCL decoder have a complexity of $O(LN^2)$ calculations. In [50], to reduce the computation complexity of SCL algorithm arising from the so-called “lazy copy” technique is alleviated employing memory sharing between candidate paths. Therefore, the complexity of the SCL decoder can be reduced to $O\ (LN\ log\ N)$.

1.2.3 Successive Cancellation Stack (SCS) Decoding

An improved version of SC algorithm called SC stack [51] uses a stack for keeping candidate paths and determines the best candidate path via optimal searching. Whenever the top path has the metric N, the decoding operation stops and decision is made about the transmitted bits. The difference between SCL and SCS lies on the lengths of the candidate paths such that in SCL all the candidate paths have the same lengths, on the other hand in SCS algorithm candidate paths may have different lengths. Let D be the largest stack S in a SCS decoder. In [63], a modified version of SCS, where the number of expanding paths is limited by the parameter L, is introduced.

Considering the parameters D and L employed by a SCS decoder, we can denote a SCS decoder by SCS (L, D). The counting vector $c_{N1} = (c_1,\ c_2, \ldots, c_N\ )$ is utilized to save the number of pushing paths for a certain length, in c_{N1} the parameter c_I indicates the number of matched paths of length-i in the decoding process.

Like in SC and SCL, for the efficient implementation of SCS decoder "lazy copy" method can be utilized for reduced complexity. The computational and memory complexity of SCS can be indicated by $O\ (LN\ log\ N)$ and $O\ (DN)$. Considering the same search width L, we can state that the computational complexity of SCS (L) is smaller than the computational complexity of SCL (L) [36].

1.3 Thesis Contribution

In this thesis work, we consider efficient design of polar decoders. First, we propose low-latency high-performance polar decoders considering the decoding tree of the polar codes. In the sequel, we inspect the error propagation phenomenon of the SC decoders and propose some techniques to alleviate the degrading effects of error propagation. Besides, we also consider a new technique for the SC algorithms for which soft information is used rather than the hard bits.

1.3.1 High-Speed Decoding of Polar Codes

We have proposed an efficient decoding algorithm for polar codes that simultaneously decodes N bits. The BER characteristics of polar codes with this proposed high-speed decoding on binary erasure channels are obtained using computer simulation. The SC decoder needs $(2N-2)$ cycles for the completion of decoding operation. It is clear that the decoding latency created by a large N is not suitable for high-speed real-time applications [37]. Therefore, polar codes need low latency and a high-speed decoding structures. Our proposed algorithm can decode N consecutive bits simultaneously. Therefore, the delay can be reduced, and the speed can be increased.

1.3.2 The Effect of Error Propagation on Performance of Successive Cancellation (SC) Decoder of Polar Codes

Wrong bit decisions in SC decoding can be due to two reasons: channel noise or error propagation due to previous error decisions in bits. At the first wrong decision, since there are no previous errors, the first erroneous decision must be due to the channel parameters. In this thesis, we discuss and analyze the effect of error propagation on the performance of successive cancellation (SC) decoder of polar codes. We have also considered successive cancellation using decoding for polar codes with CRC. With this concatenated structure it is possible to improve the bit error rate performance by estimating the first error location. This new algorithm is more complex than original SC at high rate.

1.4 Thesis Organized

This thesis is outlined as follows. In Chapter 2 Literature Review is introduced.

In Chapter 3 we will state general preliminaries and we briefly introduce the basic concepts of channel polarization and polar codes as well as the main ideas about encoding operations. We end the chapter with some words on code construction issues.

In Chapter 4 We explain the main ideas behind the successive cancellation (SC) decoding algorithm. We explain that polar codes utilizing SC decoding algorithm can achieve the capacity of symmetric channels. It is shown that the complexity of the encoding and decoding algorithms for the polar codes can be indicated by $O(N\ logN)$ where N is the block length. In the sequel, three improved versions of the successive cancellation (SC) decoding algorithms, the successive cancellation list (SCL), the successive cancellation list with CRC and successive cancellation stack (SCS) which improve the performance of polar codes without increasing the code length are explained and some simulation results are provided.

Chapter 5 considers a high-speed decoding method based on successive cancellation (SC) decoding of polar codes using tree structure. The proposed high-speed decoding algorithm that can decode N bits at the same time, i.e., in parallel is discussed in detail in this chapter.

The proposed high-speed decoding approach is simulated on computer environment and its BER performance is compared to that of the classical successive cancellation method. Chapter 6 deals with error propagation issue. In chapter 6, we discuss and analyze the effect of error propagation on the performance of successive cancellation (SC) decoder of polar codes and we introduce a concatenated structure involving successive cancellation and CRC algorithms. This combined structure improves the bit error rate and frame error rate performance by estimating first error location. Chapter 7 presents the conclusion and future work.

CHAPTER 2

LITERATURE REVIEW

The amount of memory needed for storing the computational parameters of the decoder's algorithm and the amount of memory needed for the encoding operation are critical factors for good code design. A code consists of 2^{NR}codewords, where R and N are the rate of the code and block length respectively. In [2], Shannon also characterized the rate-distortion trade-off $R(D)$ which is used to determine the lowest rate for a targeted average distortion D. A typical code requires a memory amount of $O(N^{2NR})$ bits which is inappropriate for practical applications. Dobrushin and Elias in [3], [4], [5], initiated the researches on the channel code development, and they showed that linear codes can be used to approach the capacity limits drawn by the Shannon.

Linear codes are nothing but subspaces of vector spaces. A linear code can be specified in terms of the basis of a subspace. The number of codewords that can be generated is $2^{RN}$where N is the length of the information stream, and for the designed code, the memory requirement can be indicated as $O(N^2)$ bits. In [6], [7] Goblick has shown that the same formulas also hold in the case of source coding. The second problem is the computational complexity of the encoding and decoding operations. A code may not have large memory requirements, but, it may have large amount of encoding and decoding complexity. In the following, we will give a short background of some important developments of channel coding theory. In [8] deep and comprehensive details about channel coding can be found for further reference.

2.1 Channel Coding

Channel codes are usually designed algebraically since the beginning of coding theory. Minimum distance of a code has a large impact on its performance. For this reason, researches focused on the design of codes with large minimum distances and good spectrums. Besides, tolerable encoding and decoding complexity was another factor for the good code design. As we stated before channel codes are used correct the transmission errors at the received data stream. The decoding operation can be divided into two main categories. One is called hard decoding, and the other is called soft decoding. In hard decision decoding, first the received signal is demodulated and then it is passed through the hard decoder which deals with the bits 0 and 1 to recover the transmitted data bits. On the other hand, in soft decoding operations demodulation is not performed, instead, bit probabilities are computed and further processed for the calculation of the bit probabilities of the data bits. Decision on the value of bits is made at the final stage of the soft decoding operation.

Hard decoding tries to find the codeword closest to the demodulated bit stream in terms of Hamming distance. Codes with large minimum distances show better performance, since as long as the number of errors occurred is smaller than half of the minimum distance of the code, the bit error can be corrected and correct transmitted codeword can be determined. Hamming codes, which are single error correction codes, are considered to the first algebraic codes developed. Some other algebraic codes can be listed as Golay codes, BCH codes [9], [10], Reed-Muller codes [11], [12] and Reed-Solomon codes [13]. Over several decades many efficient decoding algorithms are developed for these linear codes. Combined codes or product codes are first studied by Elias in [14] where large codes are constructed using two or more shorter codes. Two codes $C1$ and $C2$ of length $n1$ and $n2$ are combined is such a way resulting in a product code placed into a matrix of size $n1 \times n2$ matrix such that each column is a codeword of $C1$ and each row is a codeword of $C2$. The rows and columns of the product code are decoded separately using a low complexity decoding algorithm. However, the performance of the product code proposed in [14] is much far away the capacity limit. Code concatenation is also studied by Forney in [15] where the information frame is encoded by the code C1 and the resulting stream is again encoder by the code C2. Forney in its work showed that using code concatenation, it is possible to decrease the probability of decoding error in an exponential manner using decoding algorithms having polynomial complexity.

Convolutional codes which are another class of error correcting codes have a vital place in communication world. The Viterbi algorithm, which minimizes the block error probability rather, is used in [16]. An improved version of Viterbi algorithm which is the BCJR algorithm, which minimized the bit error probability rather than the block error probability, is introduced in [17]. Studies show that using convolutional codes reliable transmission, which is related to the exponential decrease of the error probability, can be achieved for code rates lower than the channel capacity. However, to achieve the capacity limit frame length should be chosen large enough and this created a significant load for the complexity of decoding algorithms.

For practical implementation purposes, a decoding algorithm called Fano sequential decoding algorithm, which has linear complexity for rates less than the cutoff rate in block length independent of the constrained length, is proposed in [18]. The cutoff rate of the code is algebraically calculated and it is less than the channel capacity.

Another modern type of codes introduces in the 1960s are low density parity check (LDPC) codes [19]. The parity check matrices of the LDPC codes have a sparse structure. In fact the total number of '1's along each row and column is a constant number. Gallager proposed a decoding algorithm for LDPC codes. However, due to large computational requirements of the LDPC decoders, the LDPC codes did not get sufficient attention by researchers at the time of its introduction due to the low computational capability of the electronic devices.

Turbo codes which are introduced in [20] by Berrou, Glavieux and Thitimajshima [20] was a breakthrough among coding society. Turbo codes approached to the capacity limits the most when all the other codes till the introduction of turbo codes are considered. Turbo codes are constructed using two convolutional codes concatenated in parallel and an interleaver is used between convolutional codes. Turbo codes are decoded in an iterative manner using the BCJR algorithm. The BCJT algorithm is employed for each components codes, and at each iteration component codes exchange soft information. The complexity of the BCJR algorithm changes in a linear manner considering the length of codewords. The reason behind the astonishing performance of the turbo codes lies on the use of interlaver between the component codes and the exchange of the soft information between component codes.

Making use of the sparse matrices McKay and Neil introduced new codes in [21] and they showed that the proposed codes show good performance with belief propagation algorithm having low complexity. The codes introduced by McKay and Neil can be considered as special case of LDPC codes and the decoding algorithm they considered was an equivalent version of the decoding algorithm introduced by Gallager. In sequel, about at the same time Sipser and Spilman [22] constructed extender codes and proposed a simple decoding algorithm for the proposed codes. In [23], [24] Wiberg, Loeliger and Kotter designed a joint structure involving turbo and LDPC codes. It can be concluded that different approaches to the channel coding problem sometimes invents different versions of the same algorithms as in the case of MacKay and Neal and the probabilistic decoding of Gallager. The superior performance of turbo codes which lead to the subsequent re-discovery of LDPC codes aroused interest in LDPC codes and messaging algorithms.

A number of papers about the analysis of message passing algorithm were released by Lubi, Mitzenmacher, Shokrollahi, Spilman and Steman in [25], [26], [27], [28]. In [25], [27], the authors introduced the "peeling decoder" which is suboptimal decoder used for binary erasure channels (BECs).

They designed capacity achieving codes peeling decoders for the BEC. In sequel, the structure of the peeling decoder is explained as a process on a tree. This new structure was simpler than the ones in [25], [27]. Richardson and Urbanke introduced the density evaluation approach in [30] for the analysis of BEC [29] for discrete memoryless symmetric channels and introduced a class of algorithms falling into the category of message-passing algorithms. Joint use of combined density technique and a number of optimization techniques for the analysis of belief propagation is introduced in [31]. A variety of turbo like and LDPC like codes are also introduced empirically by the researchers to achieve the capacity for various type of communication channels. However, none of these codes allocate bandwidths for certain channels and guarantee the transmission of a data bit.

In this thesis, we work on the polar codes. The polar codes are introduced by Arikan in [32]. Polar codes achieve the capacity of communication channels and they have low complexity encoding and decoding algorithms. Polar codes can be considered as the most serious development in channel coding history. Polar codes are designed mathematically and their capacity achieving properties can be algebraically proven which was not the case for all the codes invented up to date. In fact, Arikan published a number of papers which prepares necessary background for the invention before the introduction of polar coding and channel polarization papers appearing in [33] and [34]. Shortly after the introduction of polarization, a number of articles and papers were published about its performance and the expansion of its areas of application. In [35], the authors compared polar codes to that of the Reed-Muller codes under belief-propagation decoding.

In [37]-[51], the successive cancellation (SC) decoding algorithm, and its improved versions called successive cancellation list (SCL) and the successive cancellation stack (SCS) decoding algorithms showing improved performances are introduced considering the use of the same parameter sets for all the codes.

In [50] the fundamental logic of channel polarization, construction of polar codes and the decoding algorithm (SC) are presented. In sequel, successive cancellation list (SCL) decoding algorithm achieving the performance of maximum likelihood (ML) decoder with an acceptable complexity in [50] is introduced. Following the introduction of the SCL algorithms, another improved version of the SC algorithm called successive cancelation stack (SCS) is proposed in [51], and SCS algorithm shows good performance at high SNR values with tolerable computational complexity. Another modified architecture of SC decoder is presented in [53]. In [54], authors utilized the semi-parallel method and making use of the advantage of the recursive structure of polar codes, they designed a unified scheme having a single encoder and decoder unit that can be used over the multi-channel. The efficient design of polar codes for BMS channels is studied in [56]. The BER performance of polar codes over different type of channels such as AWGN and BSC is studied in [57] for different codeword lengths. Speed-up techniques have been proposed in [58]- [61], and in [58] an algorithm is proposed which increases the throughput of the modified SC algorithm three times. In [58], list decoder is modified in such a way that the decoding latency is reduced, and the proposed method is implemented in software.

In [60], the speed of the polar decoder is increased at least 8 times compared to the classical SC decoder. Polar decoding using SCFlip is studied in [62] where the authors employ the use of an optimized metric for the determination of the flipping positions in a SCFlip decoder, and this approach improves SCFlip ability to locate the first bit error position. In [63], choice of CRC polynomials for embedded network applications is described and a set of good general-purpose polynomials are suggested and the suggested polynomials include a set of 35 new polynomials in addition to the 13 previously known polynomials which provide good performance for 3 and 6-bit CRC with a data word length ranging up to 2048 bits.

2.2 Turkish Literature Review Search

According to YOK eight thesis have been done on polar codes, Table 1. Lists Turkish thesis literature search.

Table 1 Turkish thesis literature search.

Author	Year	Title	Thesis Type	University
AHMET GÖKHAN PEKER	2018	Belief propagation decoding of polar codes under factor graph permutations	Master	Orta Doğu Teknik Üniversitesi
ŞÜKRÜ CAN AKDOĞAN	2018	A study on the set choice of multiple factor graph belief propagation decoders for polar codes	Master	Orta Doğu Teknik Üniversitesi
ONUR DİZDAR	2017	High throughput decoding methods and architectures for polar codes with high energy-efficiency and low latency	Doctorate	İhsan Doğramacı Bilkent Üniversitesi
ALTUĞ SÜRAL	2016	An FPGA implementation of successive cancellation list decoding for polar codes	Master	Bilkent Üniversitesi
TUFAIL AHMAD	2016	Polar codes for optical communications	Master	Bilkent Üniversitesi
ALİ ALİBRAHEEMİ	2015	Performance analysis of polar codes	Master	Cankaya Üniversitesi
SİNAN KAHRAMAN	2014	Efficient maximum likelihood decoding: From space-time block codes to polar codes	Doctorate	Istanbul technical Üniversitesi
SAYGUN ÖNAY	2014	Polar codes for distributed source coding	Doctorate	Bilkent Üniversitesi
SEMİH ÇAYCI	2013	Lossless data compression with polar codes	Master	Bilkent Üniversitesi
BERKSAN ŞERBETCİ	2012	Generator matrix selection for finite-length polar codes	Master	Bogazici university
ÜSTÜN ÖZGÜR	2009	A performance comparison of polar codes with convolution turbo codes	Master	Bilkent Üniversitesi

CHAPTER 3

POLAR CODING

Channel coding has been an ever-growing field since Shannon's seminal work in 1948 [1]. Since then, the goal of achieving capacity with relatively low complexity has been a major issue for coding society. Arıkan has reached this goal for the class of binary input discrete memoryless channels (B-DMC) [32]. He discovered the method of "channel polarization" for constructing capacity achieving codes for B-DMC.

In this chapter, we briefly introduce the basic concepts of channel polarization and polar codes as well as the main ideas about encoding operation. We end the chapter with some words on code construction issues. Throughout this chapter we will restate the main results of [32].

3.1 Preliminaries

We start with presenting the notation that is going to be used throughout this thesis. Afterwards, we introduce the main parameters which capture the notion of channel rate and reliability, namely symmetric capacity and Bhattacharyya parameter, as well as their main properties. The expression $W: X - Y$ denotes a B-DMC having input alphabet X, output alphabet Y, and transition probabilities $W(y|x), x \epsilon X, y \epsilon Y$. The input alphabet X consists of $\{0,1\}$, on the other hand, the output alphabet may have more symbols. We write W^N to denote N use of the channel W; thus, we have

$$W^N = X^N \rightarrow Y^N$$

with

$$W^N(y_1^N|x_1^N) = \prod_{i=1}^{N} W(y_i|x_i)$$

.

For a B-DMC W , we can define two channel parameters which are the symmetric capacity

$$I(W) \triangleq \sum_{y \epsilon Y} \sum_{x \epsilon X} \frac{1}{2} W(y|x) log \frac{W(y|x)}{\frac{1}{2}W(y|0) + \frac{1}{2}W(y|1)}. \tag{3.1}$$

and the Bhattacharyya parameter

$$Z(W) \triangleq \sum_{y \epsilon Y} \sqrt{W(y|0) + W(y|1).} \tag{3.2}$$

These parameters indicate the transmission rate and reliability. $I(W)$ is the highest transmission rate through the channel W for which inputs have uniform distribution. $Z(W)$ is considered to be an upper bound on the probability of maximum-likelihood (ML) decision error for the transmission through W. The range set of $Z(W)$ consists of [0,1]. The base of the logarithm in $I(W)$ is chosen as 2 which indicates that the unit of $I(W)$ is bits/sample. In a similar manner, we can conclude that the unit for code rates and channel capacities will be bits.

Intuitively, one would expect that $I(W) \approx 1$ iff $Z(W) \approx 0$, and $I(W) \approx 0$ iff $Z(W) \approx 1$. For any B-DMC W, the upper and lower bounds for $I(W)$ can be calculates as

$$I(W) \geq log \frac{2}{1 + Z(W)} \tag{3.3}$$

$$I(W) \leq \sqrt{1 - Z(W)^2}. \tag{3.4}$$

Two well-known examples of the discrete memoryless channels are the binary symmetric channel (BSC) and the binary erasure channel (BEC). A BSC is a B-DMC W with output alphabet $Y = \{0,1\}$, and transition probabilities $W(0|0) = W(1|1)$, and $W(1|0) = W(0|1)$. For BEC for each received bit $y \epsilon Y$, we have the transition probabilities

$W(y|0)W(y|1) = 0, W(y|0) = W(y|1)$, and $y = e$ is said to be an erasure symbol with the transition probability defined as $W(e|0) = W(1|e) = \alpha$ where e denotes the erased output.

The notation a_1^N is used as shorthand for the row vector $(a_1, \dots, a_N)$. For row vector a_1^N, the expression a_i^j $1 \leq i, j \leq N$ is used to denote the subvector $(a_i, \dots, a_j)$. Given a_1^N and $\mathcal{A} \subset \{1, \dots, N\}$, the expression $a_{\mathcal{A}}$ is utilized to denote the subvector $(a_i : i\epsilon\mathcal{A})$.In addition, $a_{1,o}^j$ is used for the subvector with odd indices $(a_k: 1 \leq k \leq j; k\ odd)$. $a_{1,e}^j$ denotes the subvector having even indices $(a_k: 1 \leq k \leq j; k\ even)$.

We will use GF(2) for the construction of codes, i.e., vector spaces thought the thesis. All vectors, matrices, will be constructed using the elements of GF(2), and mod-2 addition and multiplication operations will be performed between the elements of vectors and matrices. For a_1^N ,b_1^N vectors over GF (2), the expression $a_1^N \oplus b_1^N$ indicates element by element mod-2 summation of two vectors. The Kronecker product for an m-by-n matrix $A = [A_{ij}]$ and an r-by-s matrix $B = [B_{ij}]$ is defined in (3.5)

$$A \otimes B = \begin{bmatrix} A_{11}B & \cdots & A_{1n}B \\ \vdots & \ddots & \vdots \\ A_{m1}B & \cdots & A_{mn}B \end{bmatrix} \tag{3.5}$$

which is an mr-by-ns matrix. The Kronecker power $A^{\otimes n}$ is calculated in a recursive manner as $A \otimes A^{\otimes(n-1)}$ for all $n \geq 1$. The standard Landau notation $O(N)$ is used to denote the asymptotic behavior of functions.

3.2 Channel Polarization

We start by briefly giving the main notion of channel polarization. Let us have a B-DMC W with symmetric capacity $I(W)$. Now let us get two independent copies of this channel W. If we use these channels as they are, we have two symmetric capacities of $I(W)$. Channel polarization is an operation that allows us to combine those two channels, creating a vector "super channel". Afterwards, we split this vector channel back into two new channels with unequal symmetric capacities; the worse channel will have $I(W^-) \leq I(W)$, while the better one will have $I(W^+) \geq I(W)$.

The same procedure can be applied for $N > 2$; as we make N larger and larger, the symmetric capacity terms of the new channels tend more and more towards 0 or 1. Also, for infinitely large N, the fraction of extremely good channels, with symmetric capacity arbitrarily close to 1, goes to $I(W)$, while the fraction of extremely bad channels, with symmetric capacity arbitrarily close to 0, goes to $(I(W) - 1)$.

So, out of N independent copies of a given B-DMC W, we have created a second set of N polarized channels. This polarization phenomenon gives us the opportunity to pass information only through the extreme good channels with $I(W)$ close to 1.

3.2.1 Channel Combining

Channel combining is the procedure of using N (where $N = 2^n$; $n \geq 0$) independent copies of a given B-DMC W in order to recursively create a vector channel $W_N : X^N \rightarrow Y^N$.

The first channel is formed as $W_1 = W(n = 0)$. In the next recursion two channels are combined such that $W_2: X^2 \rightarrow Y^2$ $(n = 1)$, as indicated in Figure 2 where the channel transition probabilities are given as:

$$W_2(y_1, y_2|u_1, u_2) = W(y_1|u_1 \oplus u_2)W(y_2|u_2). \tag{3.6}$$

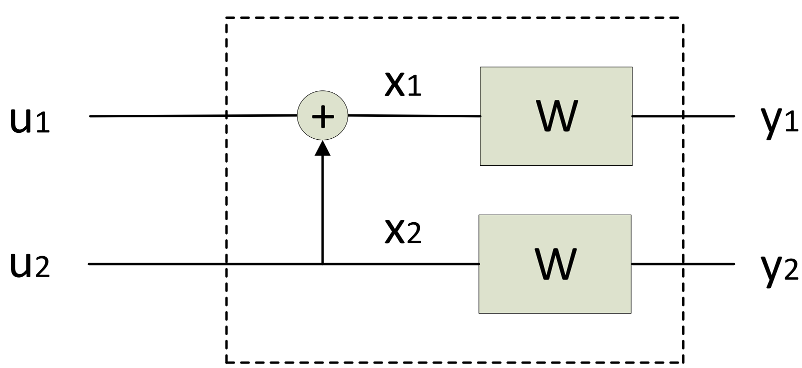

Figure 2 The W_2 channel.

Similarly, for $N = 4$, we combine two independent copies of W_2 and construct $W_4 \colon X^4 \rightarrow Y^4$, construct $W_4 \colon X^4 \rightarrow Y^4$, as shown in Figure 3.

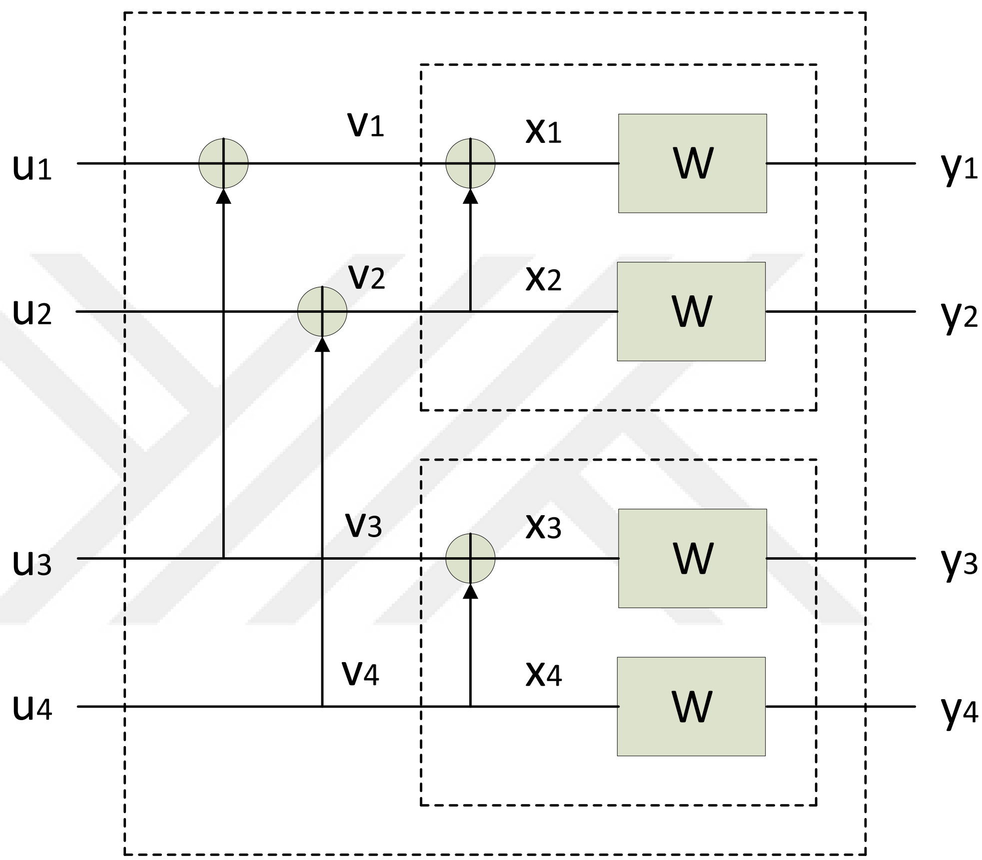

Figure 3 The relation between channel W_4 and W_2 and W.

The mapping $u_1^4 \rightarrow x_1^4$ from the input of W_4 to the output of W_4 can be expressed as $x_1^4 = u_1^4 G_4$ where we have

$$G_4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix} \tag{3.7}$$

and $W_4(y_1^4 \mid u_1^4) = W^4(y_1^4 \mid u_1^4 G_4)$ [32].

After analyzing the whole mechanism for $N = 2$, we will generalize the procedure for $N = 2^n$ at the end of this chapter.

3.2.2 Channel Splitting

We have synthesized the vector channel W_N from independent copies of B-DMC W, the next step of channel polarization is to split W_N back into a set of channels $W_N^{(i)}: X \rightarrow Y^N \times X^{i-1}, 0 \leq I \leq N$, the transition probabilities of these channels are given as:

$$W_N^{(i)}\left(y_1^N, u_1^{i-1} \middle| u_i\right) \triangleq \sum_{u_{i+1}^N \in X^{N-i}} \frac{1}{2^{N-i}} W(y_1^N | u_1^N) \tag{3.8}$$

We combined two independent copies of W (i.e. W_2) to create the vector channel, $W_2: X^2 \rightarrow Y^2$ and we can polarization these channels by splitting resulting in two B-DMCs, $W^-: X \rightarrow Y^2$and $W^-: X \rightarrow Y^2 \times X$, defined as

$$W^-(y_1, y_2 | u_1) = \sum_{u_2^N \in X} \frac{1}{2} W_2(y_1, y_2 | u_1, u_2) \tag{3.9}$$

$$W^+(y_1, y_2, u_1 | u_2) = W_2(y_1, y_2 | u_1, u_2). \tag{3.10}$$

We will see in the next section that the channel seen from u_1 to (y_1, y_2) is worse than the parent one in terms of symmetric capacity (that is the reason of the W^- notation). Similarly, the channel seen from u_2 to (u_1, y_1, y_2) is better than the parent one in terms of symmetric capacity (W^+notation).

3.2.3 Channel Transformation

In this section, explain the calculation of the $I(W_N^{(i)})$ and $Z(W_N^{(i)})$ in a recursive manner for a given channel. At the end of the recursion we will have the splitted channel expression W^N which denotes the channel vector $(W_N^{(1)}, \dots, W_N^{(N)})$.

When the single-step polar transform is applied $(W, W) \rightarrow (W^-, W^+)$ on a two DMCs W, we get

$$I(W^+) + I(W^-) = 2I(W) \tag{3.11}$$

$$I(W^+) \leq I(W^-) \tag{3.12}$$

where we have equality if and only if $I(W) = 1$ or 0.

It is clear from (3.11) that the single-step channel transformation does not change the total capacity. The inequality (3.12) with (3.11) implies that although the total capacity does not change under a single-step transformation, individual channel capacities tend to get far away from each other. In case that the channels are independent of each other, than we have $I(W^+) = I(W^-) = I(W)$. The separation splitted channel capacities from each other in such a manner $I(W^+) > I(W) > I(W^-)$ is named as channel polarization. For single-step polar transform $(W, W) \rightarrow (W^-, W^+)$ the Bhattacharyya parameters of the splitted channels can be calculated and bounded as in

$$Z(W^+) = Z(W)^2, \quad Z(W^-) = 1 - Z(W)^2 \tag{3.13}$$

$$Z(W^-) \leq 2Z(W) - Z(W)^2 \tag{3.14}$$

$$Z(W^-) \geq Z(W) \geq Z(W^+) \tag{3.15}$$

We have the equality for (3.14) if W is a BEC. If $Z(W)$ equals 0 or 1, or equivalently, if $I(W)$ equals 1 or 0, in this case we have $Z(W^+) = Z(W^-)$

As in the polarization of the channel capacities, the it can be shown that the reliability parameters also polarized and preserving the bound

$$Z(W^+) + Z(W^-) \leq 2Z(W) \tag{3.16}$$

with equality if W is BEC.

For channel transformation $(W, W) \to (W^-, W^+)$, if we have BEC with erasure probability ϵ, then the splitted channels W^- and W^+ are also BECs with erasure probabilities $2\epsilon - \epsilon^2$ and ϵ^2. Similarly, if W^- or W^+ is a BEC, then it can be shown that W is also a BEC (for proof see [32]).

If we have single-step polar transformation $(W, W) \to (W^-, W^+)$ and W is BEC with erasure probability ϵ, then

$$I(W^-) = I(W)^2 \tag{3.17}$$

$$I(W^+) = 2I(W) - I(W)^2 \tag{3.18}$$

and

$$Z(W^-) = 2Z(W) - Z(W)^2 \tag{3.19}$$

$$Z(W^+) = Z(W)^2 \tag{3.20}$$

where $I(W) = 1 - \epsilon$ and $Z(W) = \epsilon$.

A. General Case of Channel Transformation

In this section, we describe the general recursive procedure of constructing W_N out of two independent copies of $W_{N/2}$, and we show that this procedure can be performed for $N = 2^n$ for BECs in a recursive manner. Generalized procedure is shown in Figure 3.3 where we can see that for the input vector u_1^N, the channel W_N is first transformed into vector s_1^N in such a way that $s_{2i-1} = u_{2i-1} \oplus u_{2i}$ and $s_{2i} = u_{2i}$, for $1 \leq i \leq N/2$, R_N is called the reverse shuffle operation. R_N maps its input $s_1^N = (s_1, s_2, \dots, s_N)$ to the output $v_1^N = (s_1, s_3, \dots, s_{N-1}, s_2, s_4, \dots, s_N)$ which is the input to the two independent copies of $W_{N/2}$.

As in the previous cases, the overall mapping from the input of the vector channel W_N to the input of the raw channels W^N can be written as $x_1^N = u_1^N G_N$, and the relationship between the transition probabilities of the vector channel and the raw channels can be written as $W_N(y_1^N \mid u_1^N) = W^N(y_1^N \mid u_1^N G_4)$ where G_N is the generator matrix of size N. Later, when we examine the encoding procedure more thoroughly, we will show that $G_N = B_N F^{\otimes n}$, for $N = 2^n$ and $n \geq 0$, where B_N is the bit-reversal matrix, and F is defined as

$$F \triangleq \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}.$$

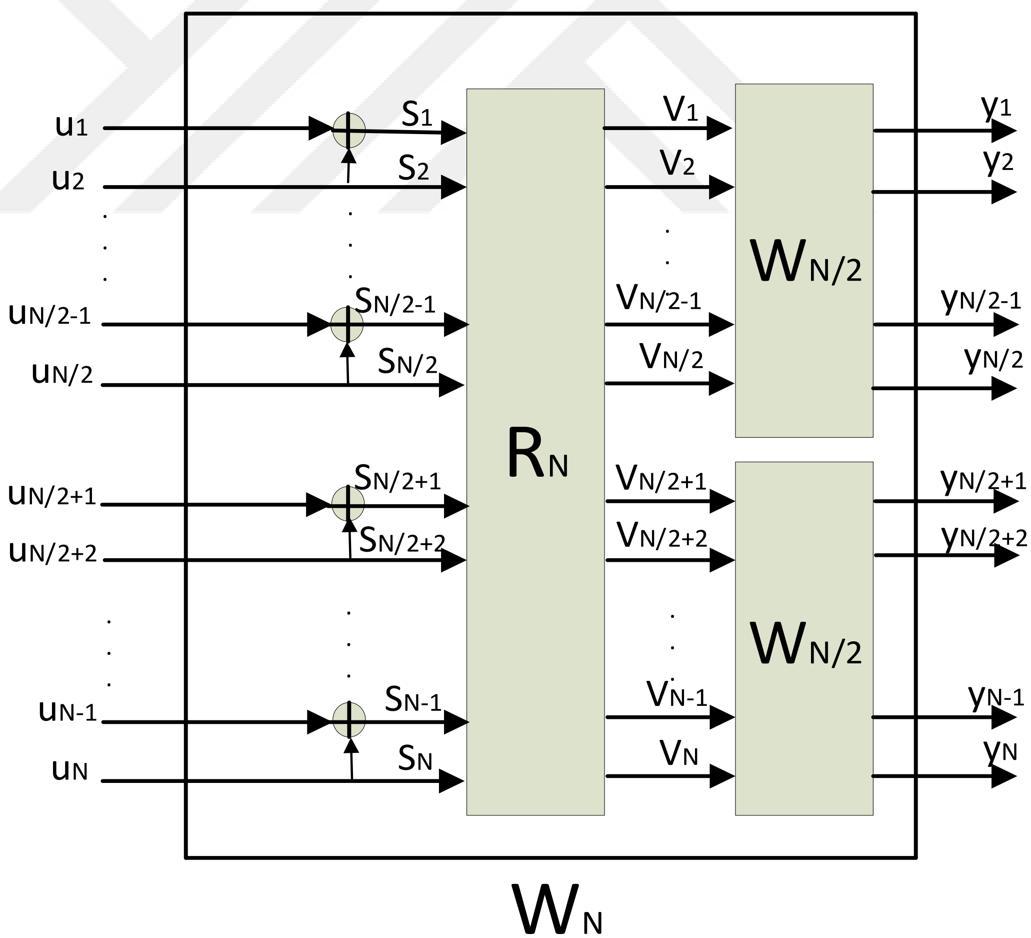

Figure 4 The channel W_N and its relation to $W_{N/2}$.

Applying the single-step polar transformation to the channels of the previous step we obtain

$$\left(W_{\frac{N}{2}}^{(i)}, W_{\frac{N}{2}}^{(i)}\right) \rightarrow \left(W_N^{(2i-1)}, W_N^{(2i)}\right). \tag{3.21}$$

The splitted channels for any $N = 2^n, n \geq 0$ and $1 \leq i \leq N$ can be written in terms of the previously splitted channels as

$$W_{2N}^{(2i-1)}\left(y_1^{2N}, u_1^{2i-2} \middle| u_{2i-1}\right) = \sum_{u_{2i}} \frac{1}{2} W_N^{(i)}\left(y_1^N, u_{1,o}^{2i-2} \oplus u_{1,e}^{2i-2} \middle| u_{2i-1} \oplus u_{2i}\right). W_N^{(i)}\left(y_{N+1}^{2N}, u_{1,e}^{2i-2} \middle| u_{2i}\right) \tag{3.22}$$

$$W_{2N}^{(2i)}\left(y_1^{2N}, u_1^{2i-2} \middle| u_{2i}\right) = \frac{1}{2} W_N^{(i)}\left(y_1^N, u_{1,o}^{2i-2} \oplus u_{1,e}^{2i-2} \middle| u_{2i-1} \oplus u_{2i}\right). W_N^{(i)}\left(y_N^{2N}, u_{1,e}^{2i-2} \middle| u_{2i}\right). \tag{3.23}$$

If we investigate the effect of general transformations on the rate and reliability parameters, we see that for the transformation $(W_{\frac{N}{2}}^{(i)}, W_{\frac{N}{2}}^{(i)}) \rightarrow \left(W_N^{(2i-1)}, W_N^{(2i)}\right)$, $N = 2^n, n \geq 0$ and $1 \leq i \leq N$, we have the relations

$$I\left(W_{2N}^{(2i-1)}\right) + I(W_{2N}^{(2i)}) = 2I\left(W_N^{(i)}\right) \tag{3.24}$$

$$Z\left(W_{2N}^{(2i-1)}\right) + Z\left(W_{2N}^{(2i)}\right) \leq 2Z\left(W_N^{(i)}\right). \tag{3.25}$$

Equality in (3.25) occurs if W is a BEC. For general transformation, we also have

$$I\left(W_{2N}^{(2i-1)}\right) \leq I(W_N^{(i)}) \leq I\left(W_{2N}^{(2i)}\right) \tag{3.26}$$

$$Z\left(W_{2N}^{(2i-1)}\right) \geq Z\left(W_N^{(i)}\right) \geq Z\left(W_{2N}^{(2i)}\right). \tag{3.27}$$

$$\sum_{i=1}^{N} I(W_N^{(i)}) = NI(W) \tag{3.28}$$

$$\sum_{i=1}^{N} Z(W_N^{(i)}) \leq NZ(W) \tag{3.29}$$

If W is a BEC with erasure probability ϵ, we have

$$I\left(W_N^{(2i-1)}\right) = I\left(W_{\frac{N}{2}}^{(i)}\right)^2 \tag{3.30}$$

$$I\left(W_N^{(2i)}\right) = 2I\left(W_{N/2}^{(i)}\right) - I\left(W_{\frac{N}{2}}^{(i)}\right)^2 \tag{3.31}$$

and

$$Z\left(W_N^{(2i-1)}\right) = 2Z\left(W_{N/2}^{(i)}\right) - Z\left(W_{\frac{N}{2}}^{(i)}\right)^2 \tag{3.32}$$

$$Z\left(W_N^{(2i)}\right) = Z(W_{N/2}^{(i)})^2 \tag{3.33}$$

where $I\left(W_1^{(1)}\right) = 1 - \epsilon$ and $Z\left(W_1^{(1)}\right) = \epsilon$.

We can see the effect of polarization of the recursive methods in (3.32) and (3.33) in Figure 5 where BEC with $\epsilon = 0.5$ is employed.

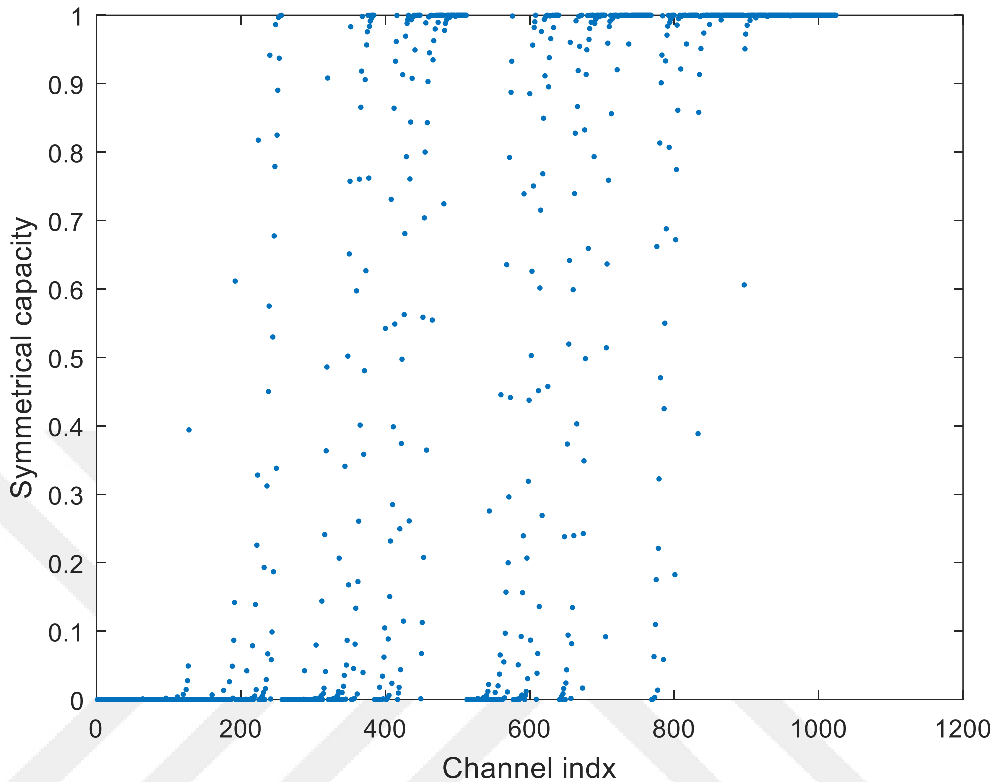

Figure 5 Plot of $I\left(W_N^{(i)}\right)$, $i = 1, \ldots, N = 2^{10}$ for BEC with $\epsilon = 0.5$.

3.3 Polar Codes

Polar codes can be considered a new type of forward error-correcting codes achieving the capacity of binary-input memoryless symmetric (BMS) channels. This implies that transmission through such channels at the highest possible rate is possible using polar codes. Besides, polar codes have low encoding and decoding complexities.

A polar code can be defined via 4-parametrs (N, K, A, u_{A^c}) where:

- N is the codeword length, i.e. block length.
- K is number of information bits, ratio k/N is called the code rate.
- A is the index set, $A \subset \{1, \ldots, N\}$, i.e., which indicates the positions of the information bits.

- u_{A^c} are the frozen bits, i.e., parity bits having 0 values all and their locations are known by the encoder and the decoder.

As mentioned in the previous section, the encoder maps the input data word u_1^N into the codeword x_1^N which is transmitted through the channels W_N, and y_1^N is the received vector. The decoder estimates the information bits $\hat{u}_1^N$ of u_1^N using the received vector y_1^N

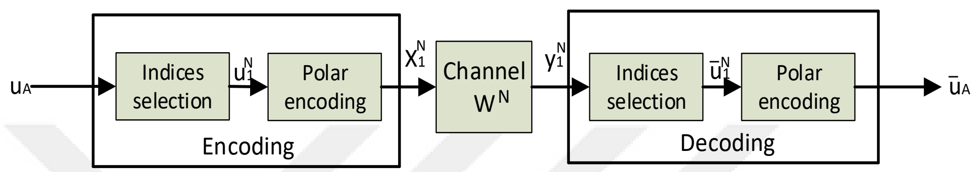

Figure 6 Polar encoding, decoding operations.

u_1^N is encoded into x_1^N using the recursive polar encoding. The codeword can be obtained using $x_1^N = u_1^N G_N$ where G_N is the generator matrix of order N. We can write encoding equation as:

$$x_1^N = u_A G_N(A) \otimes u_{A^c} G_N(A^c) \qquad (3.34)$$

where $G_N(A)$ is nothing bit the submatrix of G_N constructed using the rows whose indices appear A.

The idea of polar coding is to create virtually splitted channels $W_N^{(i)}$ and send data through those channels for which $Z(W_N^{(i)})$ is near 0 or $I(W_N^{(i)})$ is near 1, i.e., pass the data through the high capacity channels and froze the other channels [35].

3.4 Encoding

In this section, we will discuss and analyze the algebraic expressions for the generator matrix G_N. Figure 4 describes the encoding operation. Encoding complexity of polar codes will also be discussed.

3.4.1 Non -Systematic Encoding

Let $N = 2^n, n \geq 0$ and I_k denote the k-dimensional identity matrix for $k \geq 1$. The recursive encoding operation shown in Figure 4 can be algebraically expressed as

$$G_N = \left(I_{\frac{N}{2}} \otimes F\right) R_N \left(I_2 \otimes G_{\frac{N}{2}}\right), \text{ for } N \geq 2 \text{ and } G_1 = I_1 \tag{3.35}$$

Figure 7 is an alternative realization of G_N. Algebraic formula of Figure 7 is given as

$$G_N = R_N \left(F \otimes I_{\frac{N}{2}}\right)\left(I_2 \otimes G_{\frac{N}{2}}\right), \text{ for } N \geq 2. \tag{3.36}$$

We can see that (3.35) and (3.36) are equivalent to each other. From formulas (3.35) and (3.36), we notice that $\left(I_{N/2} \otimes F\right)R_N = R_N(F \otimes I_{N/2})$. Hence, we can write formula (3.36) as

$$G_N = R_N \left(F \otimes G_{\frac{N}{2}}\right). \tag{3.37}$$

For $N/2$, we get

$$G_{N/2} = R_N \left(F \otimes G_{\frac{N}{4}}\right) \tag{3.38}$$

leading to

$$G_N = R_N \left(F \otimes R_N \left(F \otimes G_{\frac{N}{4}}\right)\right) \tag{3.39}$$

By using the identity $(AC) \otimes (BD) = (A \otimes B)(C \otimes D)$ with $A = I_2, B = R_{N/2}, C = F, D = F \otimes G_{N/2}$, (3.39) takes the form

$$G_N = R_N \left(I_2 \otimes R_{\frac{N}{2}}\right)\left(F^{\otimes 2} \otimes G_{\frac{N}{4}}\right). \tag{3.40}$$

where we have

$$G_{N/2} = B_N F^{\otimes n} \tag{3.41}$$

$$B_N = R_N \left(I_2 \otimes B_{\frac{N}{2}} \right). \tag{3.42}$$

B_N represents permutation matrix also known as bit-reversal which is calculated as in (3.42) where I_2 is the 2-D identity matrix, B_2 is initialized as $B_2 = I_2$. $\otimes$ is the Kronecker product, R_N is the permutation operation which maps the input sequence $\{1,2,3,4, \ldots\ldots, N\}$ to $\{1,3, \ldots.., N-1,2,4, \ldots, N\}$ and $n = \log_2 N$.

For example, the $(4,2,\{2.4\}, (0,0))$ code has the encoder operation describes as

$$\mathrm{x}_1^4 = \mathrm{u}_1^4 \mathrm{G}_4 \rightarrow x_1^4 = u_A G_4(A) \otimes u_{A^c} G_4(A^c) \rightarrow$$
$$x_1^4 = (u_2, u_4) \begin{bmatrix} 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix} + (0,0) \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{bmatrix}.$$

For information word $(u_2, u_4) = (1,1)$, codeword is obtained as $\mathrm{x}_1^4 = (0, 1, 0, 1)$.

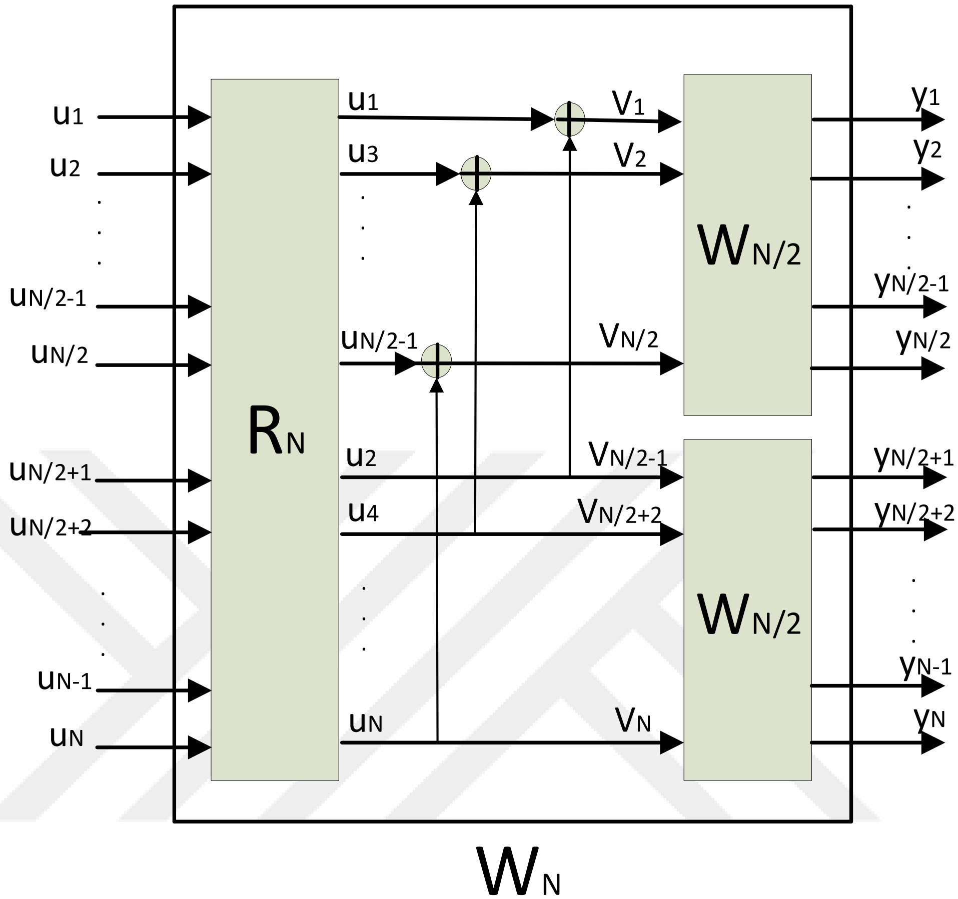

Figure 7 An alternative realization of the recursive construction for W_N.

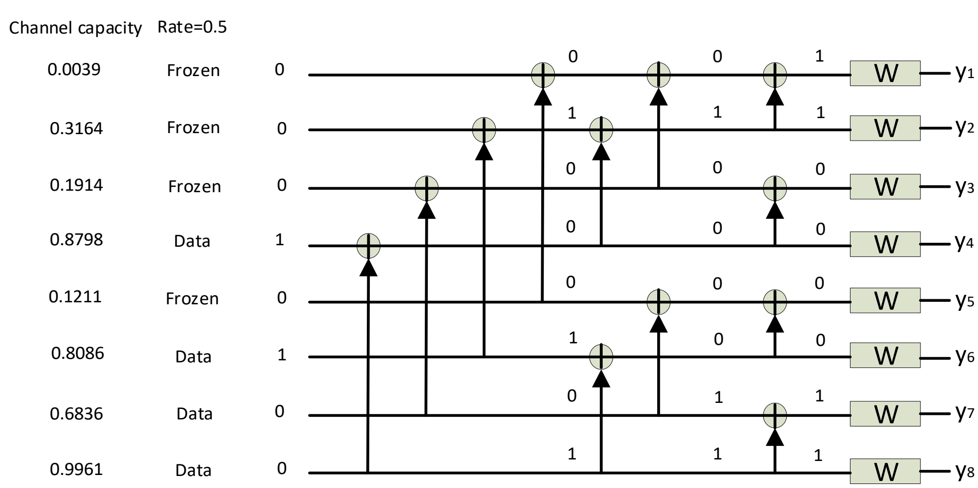

Figure 8 Polar code example: BEC with $\epsilon = 0.5$, $N = 8$ and $rate = 0.5$.

3.4.2 Encoding Complexity

For any B-DMC W $Pe(N, K, A, u_{A^c})$ denotes the probability of block error under SC decoding for an (N, K, A, u_{A^c}) code. The $Pe(N, K, A, u_{A^c})$ is upper bounded as

$$Pe(N, K, A, u_{A^c}) \leq N \max_{i \epsilon A_N} \{Z(W_N^i)\} = O\left(N^{\frac{-1}{4}}\right). \tag{3.34}$$

The most important issue about polar coding is the complexity of encoding, decoding and code construction. Due to the recursive structure of G_N, the encoding complexity of polar codes can be reduced from $O(N^2)$, which is the complexity of vector-matrix multiplication, to $O(N\ logN)$. The encoding is done layer by layer for $n = logN$ layers and within each layer the computational complexity is $O(N)$. Figure 8 illustrates how the encoding is done for an $(N, K) = (8, 4)$ polar code designed for a BEC with erasure probability $\epsilon = \frac{1}{2}$.

CHAPTER 4

SC DECODING FOR POLAR CODES WITH TREE STRUCTURE AND PROPOSED SOFT SUCCESSIVE CANCELATION ALGORITHM

When Arikan introduced polar codes, he used SC algorithm for the decoding of polar codes. SC algorithm is SC suitable considering the structure of polar codes and it has much lower computation complexity than its counterpart algorithms such as BP, and we can say that SC algorithm is the most popular decoding algorithm utilized for polar codes. On the other hand, the serial decoding property causes the long latency problems for SC decoders which is a main drawback for their use in practical communication systems.

In this chapter, we first propose a tree structure for the successive cancelation (SC) decoding of polar codes. The proposed structure is easy to implement in hardware and suitable for parallel processing operations. Next, using the proposed tree structure. We introduce a new approach to the successive cancelation of polar codes. The proposed approach uses the soft likelihood ratios of the predecessor information bits for the determination of successor information bits. The proposed method can be considered for the construction of joint iterative communication systems exchanging soft likelihoods. The performance of the SC algorithm can be improved by using SCL decoding algorithm. A successive cancellation list (SCL) decoding algorithm to improve the performance of polar codes is discussed. Compared with classical successive cancellation decoding algorithms, SCL concurrently produces at most L best candidates during the decoding process to reduce the chance of missing the correct code word. However, the SCL will increase hardware complexity, which prevents the efficient implementation. This chapter, investigates polar codes with a suggested combination of list decoding and CRC to improve the performance of the system further.

The outline of the chapter is as follows. In Section 4.1, we provide SC decoding of polar codes. A tree structure for the successive cancelation decoding of polar codes is proposed in Section 4.2. In Section 4.3, we introduce soft successive cancelation algorithm in details. A tree structure for the successive cancelation decoding of polar codes is introduced in Section 4.4 and concepts of SCL decoding are analyzed. Successive cancellation list decoding algorithm with CRC over BEC channel is introduced in Section 4.5. In Section 4.6, concepts of SCS decoding is discussed in brief. Simulation results are presented in Section 4.7.

4.1 SC Decoding

The encoder maps the input bits u_1^N into the codeword x_1^N , which is transmitted through the channels W_N and $y_1^N$is received signal. Figure 9 shows a polar encoder for $N = 4$. The task of the decoder is to estimate information bits $\hat{u}_1^N$ from the received signal y_1^N. Actually, the duty of the decoder is to estimate the data bits using

$$\hat{u}_i \triangleq \begin{cases} u_i, & if\ i \in\ A^c \\ h_i\left(y_1^N, u_1^{i-1}\right), & if\ i\ \in A \end{cases} \tag{4.1}$$

where $h_i(y_1^N, \hat{u}_1^{i-1})$ is decision function defined as:

$$h_i\left(y_1^N, \hat{u}_1^{i-1}\right) \triangleq \begin{cases} 0, if\ \dfrac{W_N^{(i)}\left(y_1^N, \hat{u}_1^{i-1}|0\right)}{W_N^{(i)}\left(y_1^N, \hat{u}_1^{i-1}|1\right)} \geq 1 \\ 1, otherwise. \end{cases} \tag{4.2}$$

In (4.2) the rational term can be defined as

$$L_N^{(i)}\left(y_1^N, \hat{u}_1^{i-1}\right) \triangleq \frac{W_N^{(i)}\left(y_1^N, \hat{u}_1^{i-1} \middle| 0\right)}{W_N^{(i)}\left(y_1^N, \hat{u}_1^{i-1} \middle| 1\right)} \tag{4.3}$$

where $L_N^{(i)}$ is named as likelihood ratio (LR) [32].

The LRs can be computed recursively using recursive formulas (4.15) and (4.16).

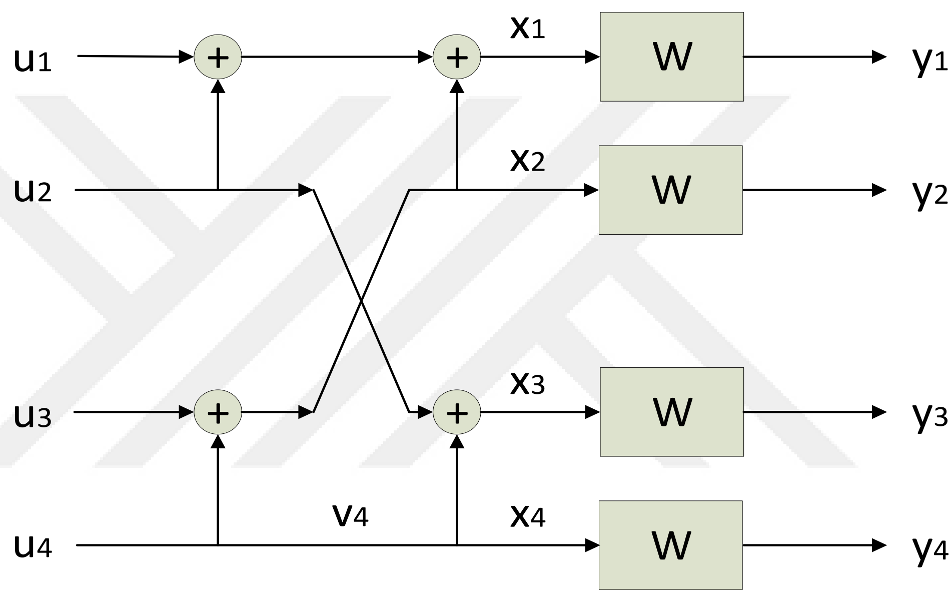

Figure 9 Polar Encoder for $N = 4$.

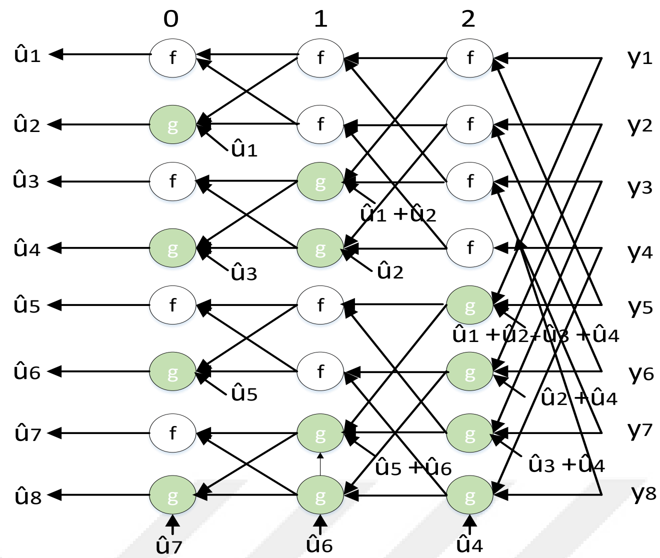

Figure 10 The calculation diagram of the SC decoder for $N = 8$.

The computation graph of the SC decoder for $N = 8$ is shown in Figure 10. This graph contains two types of nodes, namely f nodes and g nodes. Both types of nodes have one output LR and two inputs LRs. The g nodes have an extra input called the partial sum. The partial sums form the decision feedback part of the SC decoder. To calculate LRs, we use formula (4.15) at f nodes and formula (4.16) at g nodes.

In [32], a graphical butterfly structure is suggested for the calculation of the likelihood ratios. However, the structure is complex and not suitable for parallel processing operations. For this reason, we propose a tree structure for the successive cancelation decoding of polar codes in the following sub-section.

4.2 Tree Structure for the Decoding of Polar Codes

The kernel units that are repeatedly used in encoder and decoder structures of polar codes are depicted in Figure 11. The kernel unit can also be considered as the smallest polar encoder and decoder units.

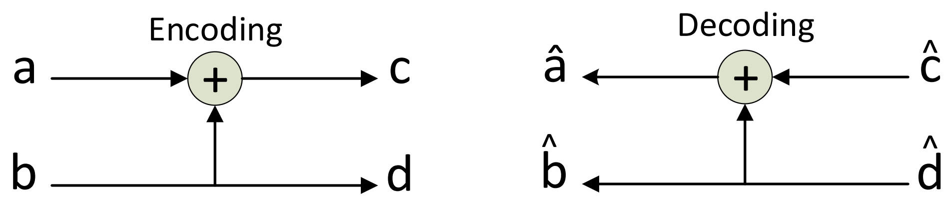

Figure 11 Kernel encoding and decoding units of polar codes.

Considering the decoder unit in Figure 11 where a, b, c, d and $\hat{a}, \hat{b}, \hat{c}, \hat{d}$ are binary variables. Using the decoder unit, we can calculate the likelihood ratios as [64]

$$\hat{a} = \hat{c} \oplus \hat{d} \text{ and } \hat{b} = \hat{d} \tag{4.4}$$

$$P(\hat{a} = 0) = P(\hat{c} = 0)P(\hat{d} = 0) + P(\hat{c} = 1)P(\hat{d} = 1) \tag{4.5}$$

$$P(\hat{a} = 1) = P(\hat{c} = 0)P(\hat{d} = 1) + P(\hat{c} = 1)P(\hat{d} = 0) \tag{4.6}$$

$$LR(\hat{a}) = \frac{P(\hat{a} = 0)}{P(\hat{a} = 1)} \rightarrow$$
$$LR(\hat{a}) = \frac{P(\hat{c} = 0)P(\hat{d} = 0) + P(\hat{c} = 1)P(\hat{d} = 1)}{P(\hat{c} = 0)P(\hat{d} = 1) + P(\hat{c} = 1)P(\hat{d} = 0)} \rightarrow \tag{4.7}$$
$$LR(\hat{a}) = \frac{1 + LR(\hat{c})LR(\hat{d})}{LR(\hat{c}) + LR(\hat{d})}$$

The value of $\hat{a}$ can be decided using (4.7). After deciding the value of $\hat{a}$, we can start to decoding of b. If $\hat{a}$ is decided to be 0, i.e., $\hat{a} = 0$, then using (4.4), we get

$$P(\hat{b} = 0) = P(\hat{c} = 0)P(\hat{d} = 0) \tag{4.8}$$

$$P(\hat{b} = 1) = P(\hat{c} = 1)P(\hat{d} = 1) \tag{4.9}$$

$$LR(\hat{b}) = \frac{P(\hat{b} = 0)}{P(\hat{b} = 1)} = \frac{P(\hat{c} = 0)P(\hat{d} = 0)}{P(\hat{c} = 1)P(\hat{d} = 1)} \rightarrow LR(\hat{b}) = LR(\hat{c})LR(\hat{d}) \tag{4.10}$$

On the other hand, if $\hat{a}$ is decided to be 1, i.e., $\hat{a} = 1$, then using (4.4), we get

$$P(\hat{b} = 0) = P(\hat{c} = 1)P(\hat{d} = 0) \tag{4.11}$$

$$P(\hat{b}=1)=P(\hat{c}=0)P(\hat{d}=1) \tag{4.12}$$

$$LR(\hat{b})=\frac{P(\hat{b}=0)}{P(\hat{b}=1)}=\frac{P(\hat{c}=1)P(\hat{d}=0)}{P(\hat{c}=0)P(\hat{d}=1)}\rightarrow LR(\hat{b})=LR(\hat{c})^{-1}LR(\hat{d}) \tag{4.13}$$

The formulas in (4.10) and (4.13) can be combined as

$$LR(\hat{b})=[LR(\hat{c})]^{1-2\hat{a}}\times LR(\hat{d}) \tag{4.14}$$

The formulas in (4.7) and (4.14) are expressed in a recursive manner in [32] as

$$L_N^{(2i-1)}(y_1^N,\hat{u}_1^{2i-2})=\frac{L_{N/2}^{(i)}\left(y_1^{N/2},\hat{u}_{1,o}^{2i-2}\oplus\hat{u}_{1,e}^{2i-2}\right)L_{N/2}^{(i)}\left(y_{\frac{N}{2}+1}^{N},\hat{u}_{1,e}^{2i-2}\right)+1}{L_{N/2}^{(i)}\left(y_1^{N/2},\hat{u}_{1,o}^{2i-2}\oplus\hat{u}_{1,e}^{2i-2}\right)+L_{N/2}^{(i)}\left(y_{\frac{N}{2}+1}^{N},\hat{u}_{1,e}^{2i-2}\right)} \tag{4.15}$$

and

$$L_N^{(2i)}(y_1^N,\hat{u}_1^{2i-1})=[L_{\frac{N}{2}}^{(i)}\left(y_1^{\frac{N}{2}},\hat{u}_{1,o}^{2i-2}\oplus\hat{u}_{1,e}^{2i-2}\right)]^{1-\hat{u}_{2i-1}}.\,L_{\frac{N}{2}}^{(i)}\left(y_{\frac{N}{2}+1}^{N},\hat{u}_{1,e}^{2i-2}\right) \tag{4.16}$$

Finally, decision is made using,

$$u_i=\begin{cases}0, if \;\; LR(u_i)>1\\ 1, otherwise\end{cases}.$$

It is obvious that the recursive formulas given in (4.15) and (4.16) are similar to those given in (4.7) and (4.14).

Now, let’s explain the proposed tree structure for the decoding of polar codes. To understand the derivation of the proposed algorithm, first let’s give some information about the formation of tree structure used in decoding operation.

In Figure 9, the encoding structure of the polar codes for $N = 4$ is illustrated. In decoding operation, the flow of the signals is reversed. For $N = 4$, the decoding structure with reversed signal flow is shown in Figure 12.

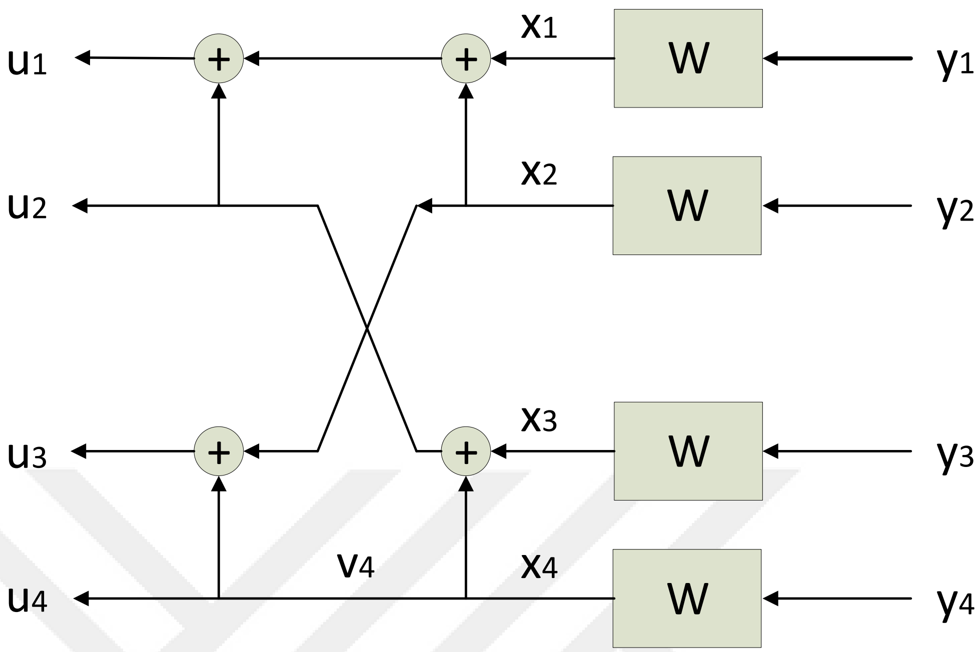

Figure 12 Decoding operation for $N = 4$.

The decoding path for information bit u_1 and its tree equivalent is shown in Figure 13 where the likelihoods are calculated as

$$LR(g_1) = \frac{\left(1 + LR(x_1)LR(x_2)\right)}{LR(x_1) + LR(x_2)}$$
$$LR(g_2) = \frac{\left(1 + LR(x_3)LR(x_4)\right)}{LR(x_3) + LR(x_4)} \qquad (4.17)$$
$$LR(u_1) = \frac{\left(1 + LR(g_1)LR(g_2)\right)}{LR(g_1) + LR(g_2)}$$

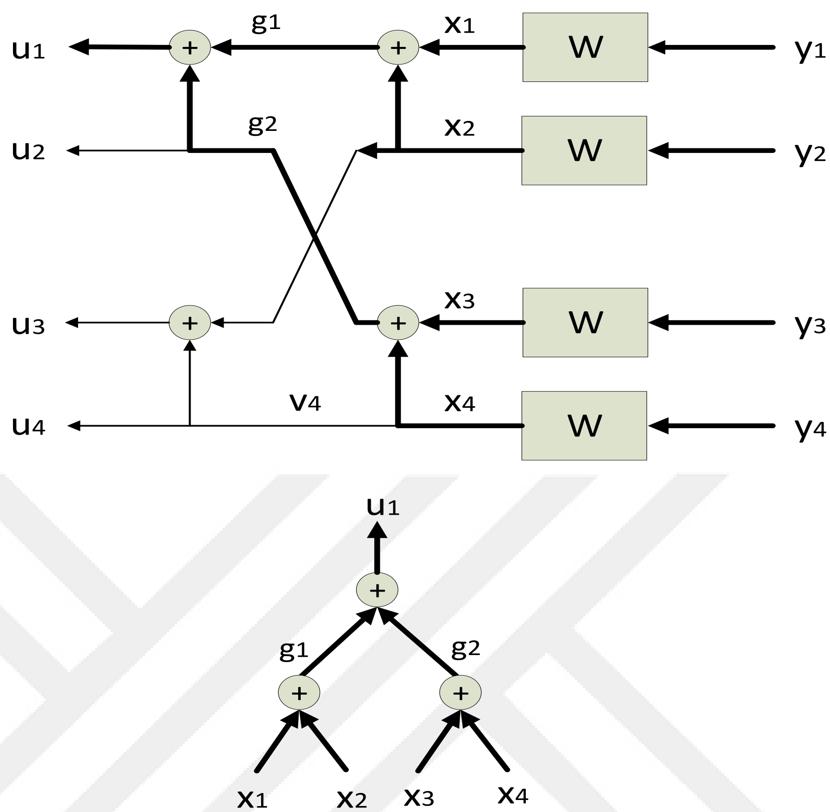

Figure 13 Decoding path for u_1 and its equivalent tree structure.

For better comprehension, let's give one more example. The decoding path for u_3 and its tree equivalent tree structure is depicted in Figure 14 where the likelihoods are calculated as

$$
\begin{aligned}
LR(g_1) &= [LR(\, x_1)]^{1-2(u_1 \oplus u_2)} \times LR(x_2) \\
LR(g_2) &= [LR(\, x_3)]^{1-2u_2} \times LR(x_4) \\
LR(u_3) &= \frac{\left(1 + LR(g_1)LR(g_2)\right)}{LR(g_1) + LR(g_2)}
\end{aligned}
\tag{4.18}
$$

When the tree structure in Figure 14 is inspected in details, we see that some of the nodes have assigned bits, for instance in the tree structure of Figure 14, $u_1 \oplus u_2$ is assigned to the lower left node

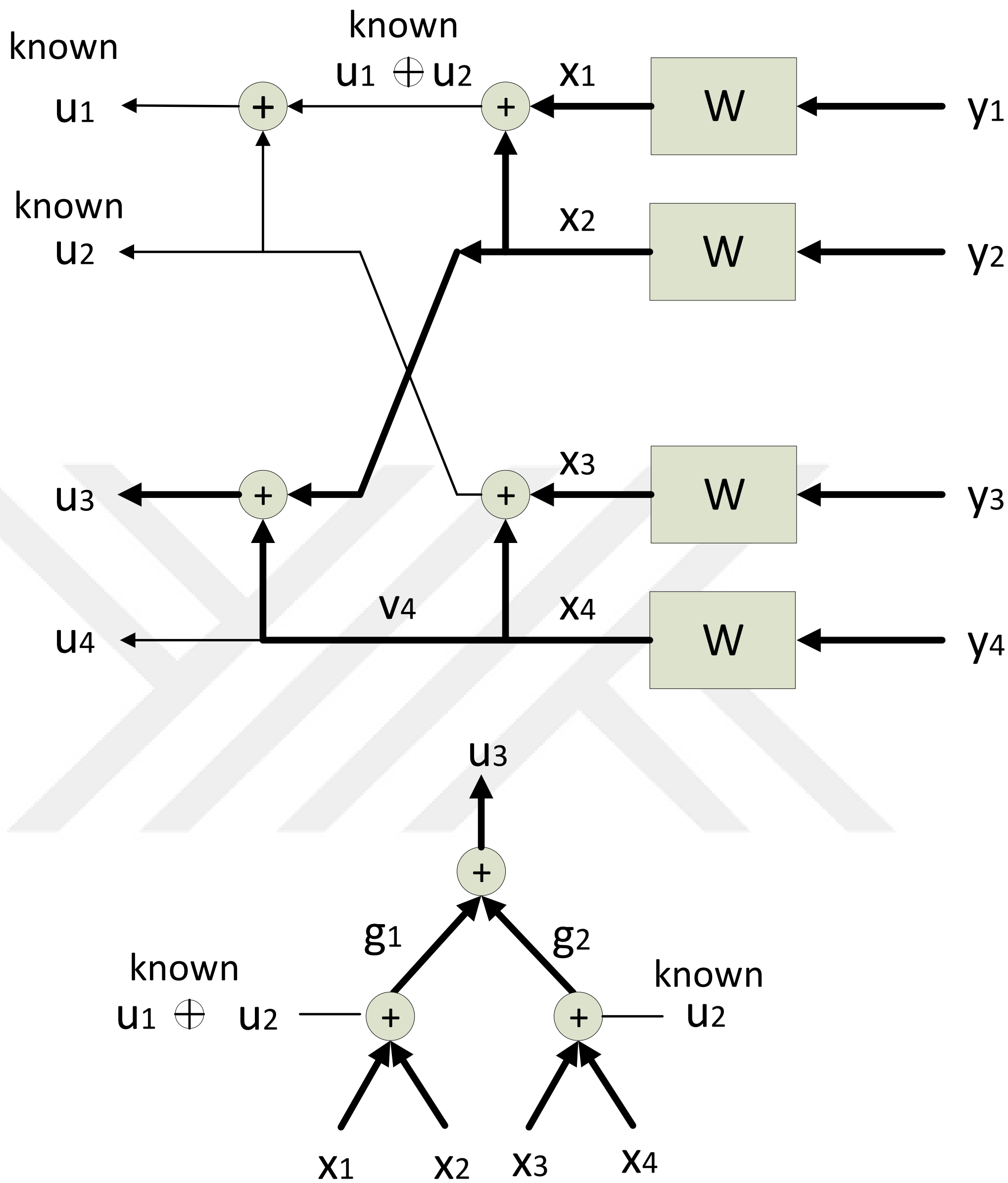

Figure 14 Decoding path for u_3 and its equivalent tree structure.

and u_2 is assigned to the lower right node. And if a bit is assigned to a node, we call these nodes as g nodes, on the other hand, if a node does not have any bit assignment, we call such nodes f nodes. In addition, we employ (4.15) for the likelihood ratio calculation for outputs of the f nodes, and we employ (4.16) for the likelihood ratio calculation for the outputs of the g nodes.

Now, let's generalize the decoding logic illustrated for $N = 4$ in Figures 14 and 15. The decoding operation consists of two stages. The first stage is the distribution of previously decoded bits to the nodes, i.e., deciding on the value of node bits. The second stage involves the calculation of the likelihood values. The decoding operation goes in a recursive manner as, distribution, likelihood calculation, distribution, likelihood calculation and so on. In likelihood calculation stage, node L values are computed for each layer starting from the basement layer and top-most node L-value is used for decision. In distribution stage, decided bits are distributed to the nodes and these bits are called node bits. If there is a bit assigned to a node, then (4.15) is used to compute the node-L value otherwise (4.16) is used. The bit distribution process is performed according to the Algorithm-1.

A numerical example for the distribution operation is depicted in Figure 16 where it is seen that after the distribution 5 decoded bits to the nodes, we see that the node at the top-most level, i.e., level- 0, and the nodes at level-2 have bits, i.e., g nodes are available at level-0 and level-2, since we have $5 = 2^2 + 2^0$. Once the bits are distributed to the nodes, the decoding operation starts which involves the calculation of likelihoods from bottom to top layer.

Algorithm 1: Distribution of decoded bits to the nodes

Input N frame length and received bits.
1: **If** R is odd, then node-check-bit = $d_R then$
2: Left child-node input-bits: $L_b = d_{1,o}^{R-1} \oplus d_{1,e}^{R-1}$
3: Right child-node input-bits: $R_b = d_{1,e}^{R-1}$
4: **else**
5: Left child-node input-bits: $L_b = d_{1,o}^{R} \oplus d_{1,e}^{R}$
6: Right child-node input-bits: $R_b = d_{1,e}^{R}$
7: **end**
8: **If** $R = 1$ then
9: $Terminate$
10: **else**
11: $R = R/2$
12: Go to step-1 and repeat $1 - 6$ for the left-child and right-child nodes
13: **end**

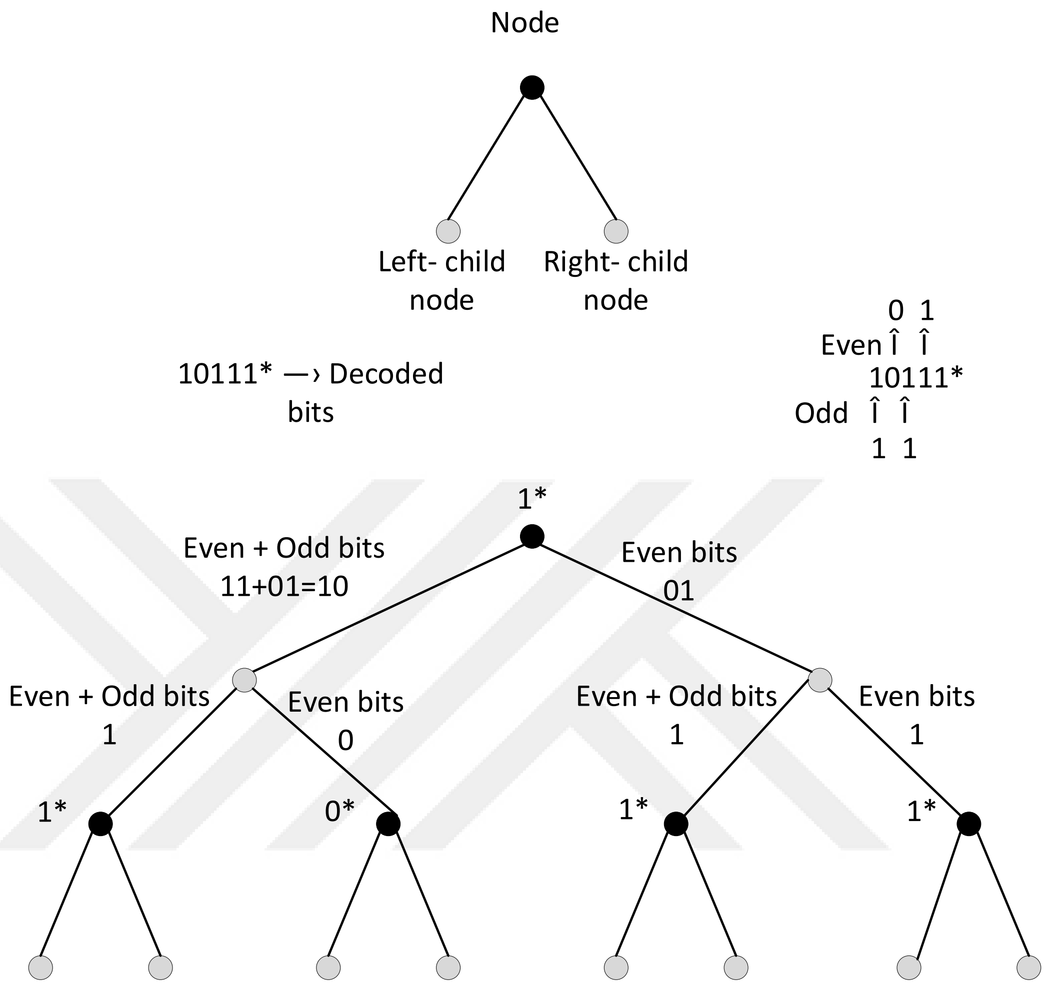

Figure 15 Distribution of the decided bits to the node.

4.3 Proposed Method and Soft Successive Cancelation Algorithm

In this section, we introduce our proposed approach for the soft decoding of polar codes using the proposed soft successive cancelation algorithm.

4.3.1 Using Soft Values for the Calculation of Likelihoods

If the formulas (4.14) or (4.16) are inspected, we see that the previously decoded bit value is used in exponential term, and this means that the previous hard decision is used for the decoding of current bit. From (4.8), (4.9), (4.11) and (4.12), we can write

$$p(\hat{b}=0)=p(\hat{a}=0,\hat{c}=0,\hat{d}=0)+p(\hat{a}=1,\hat{c}=1,\hat{d}=1) \tag{4.19}$$

$$p(\hat{b}=1)=p(\hat{a}=0,\hat{c}=1,\hat{d}=1)+p(\hat{a}=1,\hat{c}=0,\hat{d}=1) \tag{4.20}$$

from which, we obtain

$$LR_s(\hat{b})=\frac{p(\hat{b}=0)}{p(\hat{b}=1)}\rightarrow LR_s(\hat{b})\approx\frac{LR(\hat{a})LR(\hat{c})LR(\hat{d})+LR(\hat{d})}{LR(\hat{a})+LR(\hat{c})} \tag{4.21}$$

where it is seen that instead of the hard value of $\hat{a}$ as in (4.14) and (4.16), soft value of $\hat{a}$, i.e., $LR(\hat{b})$, appears in the calculation of $LR_s(\hat{b})$. Since employment of hard values results in the lose of information, employment of soft values may result in more reliable likelihood calculation for

$$LR_s(\hat{b})=\frac{p(\hat{b}=0)}{p(\hat{b}=1)} \tag{4.22}$$

With the proposed approach, the recursive equation in (4.16) can be modified as in (4.23).

$$L_N^{(2i)}\left(y_1^N,\hat{u}_1^{2i-1}\right)$$
$$=\frac{L_N^{(2i-1)}(y_1^N,\hat{u}_1^{2i-2})L_{N/2}^{(i)}\left(y_1^{N/2},LR(\hat{u}_{1,o}^{2i-2})\oplus LR(\hat{u}_{1,e}^{2i-2})\right)L_{N/2}^{(i)}\left(y_{\frac{N}{2}+1}^{N},\hat{u}_{1,e}^{2i-2}\right)+L_N^{(i)}\left(y_{\frac{N}{2}+1}^{N},\hat{u}_{1,e}^{2i-2}\right)}{L_N^{(2i-1)}(y_1^N,\hat{u}_1^{2i-2})+L_{N/2}^{(i)}\left(y_1^{N/2},LR(\hat{u}_{1,o}^{2i-2})\oplus LR(\hat{u}_{1,e}^{2i-2})\right)} \tag{4.23}$$

When (4.10), (4.13), (4.14) and (4.21) are inspected together, we see that (4.10), (4.13) are nothing but limiting cases of (4.21), that is:

$$\lim_{LR(\hat{a})\to\infty} LR_s(\hat{b}) = LR(\hat{c})LR(\hat{d}) \tag{4.24}$$

$$\lim_{LR(\hat{a})\to 0} LR_s(\hat{b}) = LR^{-1}(\hat{c})LR(\hat{d}) \tag{4.25}$$

To the best of authors knowledge, the formulas given in (4.21), (4.23), and (4.25) are new in the literature.

4.3.2 Likelihood Combination for XOR Functions

In classical successive cancelation decoding of polar codes, the previously decoded bits, i.e., hard values, are used for the decoding of successor bit. For this purpose, the previously decoded bits are used in the decoder structure of the successive cancelation algorithm. And some of the XOR node outputs are decided if there are bit values available at the XOR node inputs. Since in our approach we do not use the hard values but soft values only, we have soft values also at the inputs of the XOR gate, and soft values at the inputs of XOR gate should be combined producing a soft output value at XOR gate output. If $x_1 = (u_1)XOR(u_2)$, then we have

$$\begin{aligned} p(x_1 = 0) &= p(u_1 = 0)p(u_2 = 0) + p(u_1 = 1)p(u_2 = 1) \\ p(x_1 = 0) &= p(u_1 = 0)p(u_2 = 1) + p(u_1 = 1)p(u_2 = 0) \end{aligned} \tag{4.26}$$

from which, we get

$$LR(x_1) = \frac{1 + LR(u_1)LR(u_2)}{LR(u_1) + LR(u_2)} \tag{4.27}$$

which is the likelihood value at XOR gate output if likelihood values instead of the bit values at the inputs of the XOR gates are considered. The decisions on the value of information bits are done at the end of the decoding operation. That is, once we get the likelihood values for all the information bits, then the decoding logic is performed according to

$$u_i = \begin{cases} 0, \text{if } \mathrm{LR}(\mathrm{u_i}) > 1 \\ 1, \text{otherwise} \end{cases}.$$

4.4 Successive Cancellation List (SCL) Decoding Algorithm

SCL decoding algorithm is developed to improve the performance of SC decoding for short and moderate length codewords in [50]. In SCL decoding algorithm, L high probability decoding paths are tracked simultaneously, on the other hand, in SC decoding operation only one decoding path is tracked. If L is chosen sufficiently large, SCL algorithm can achieve the performance of ML decoding , and this is due to the computation of more accurate probability values for the information bits. The performance of the sequential decoding algorithms is sensitive to early decision errors, since it affects the rest of the decoding process. In SC decoding operation, in correct decision nof one information bit cannot be corrected in sequel and this could affect the decoder to make more bit errors.

In SC decoding operation the most likely path after is saved after every decision level (SC case), on the other hand, in SCL list decoding all possible paths are saved including their calculated likelihood. At the final stage of the decoding operation, the probabilities of all paths in the list are compared to each other, and the most likely path is chosen as the winner path. The complexity of the SCL algorithm depends on list size L, and time complexity is $O(LNlog\ N)$ and space complexity is $O(LN)$.

For efficient use of SCL decoders, we can consider a trade-off between complexity and performance, and we can trace a fixed number of most likely in the list instead of considering every possible path. A list decoder with list size L=4 is depicted in Figure 16 where in the first level all possible paths are saved, and the same thing is done in the second level as well. From level 3 and forward, truncation is performed, and for this purpose, the probabilities of all different paths are computed for each level, but only the four most likely paths survives to be used in the next list step. The least likely paths that are not considered in Figure 15 are represented by the dotted lines, and the full lines are used for the paths saved in the list for that level. Children of a node represent the continuation of a path with either a decided zero or one. Recursive formulas (4.15) and (4.16) are used to compute LRs. SCL decoding algorithm was given in details in [50].

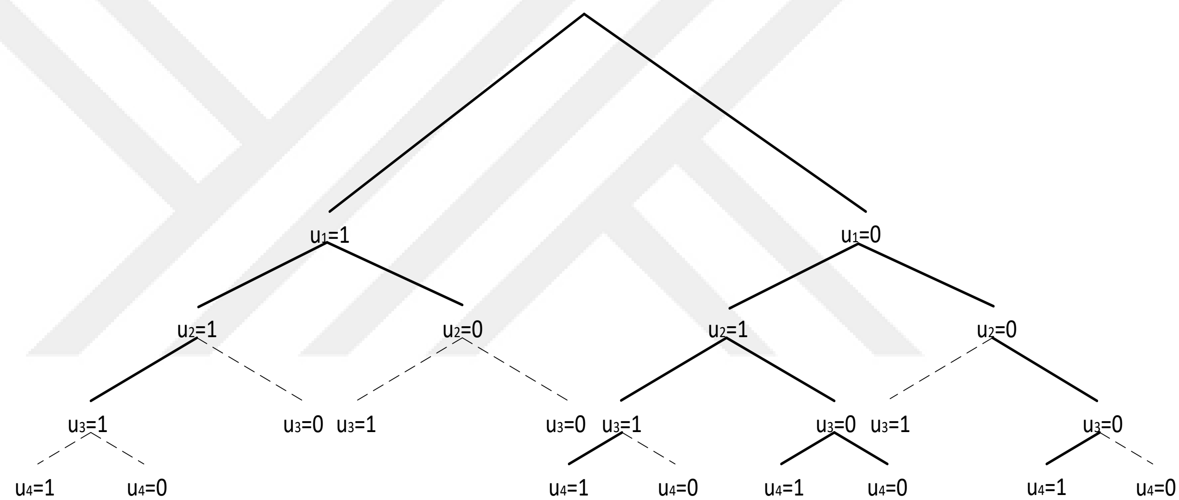

Figure 16 Example of List decoding for L = 4. The paths with low probabilities that are erased are represent by dotted lines.

.

4.5 Successive Cancellation List with CRC

Cyclic Redundancy Check (CRC) codes are a type of channel codes besides the most broadly utilized ones. Excess bits, found through the polynomial division of the code word with a predefined generator polynomial, are concatenated to the transmission frame. The decoder gets the codeword including excess bits. In this study we utilized four bits CRC and the generator polynomial used is $x^4 + x^3 + 1$ in case of BEC, and eight bits Primitive generator polynomials is used in case of AWGN channel, the generator polynomial used is $x^8 + x^4 + x^3 + x^2 + 1$ and $N = 128$.

In polar codes with list decoding, CRC comes in the last step of decoding as shown in Figure 17. The L most likely paths are saved in a list during decoding, and in the last decoding stage, the most likely path out of those L paths is chosen as the decoded code word. It was found in [52] that when errors occur in polar codes the correct code word was often in the final list, but that it was not the code word with the highest likelihood and subsequently is not selected in the last stage of the decoder. In case of CRC concatenated codeword, the CRC check could be used to make the decoder to choose the correct code word out of the list. If a word in the list at the end of decoding has higher probability than the real code word but is not a valid code word according to the CRC check, it is discarded, and the most likely codeword that is valid with its CRC is selected instead. This improves the error rates for polar codes. It should be noted that adding CRCs to polar codes changes the polar code rate slightly.

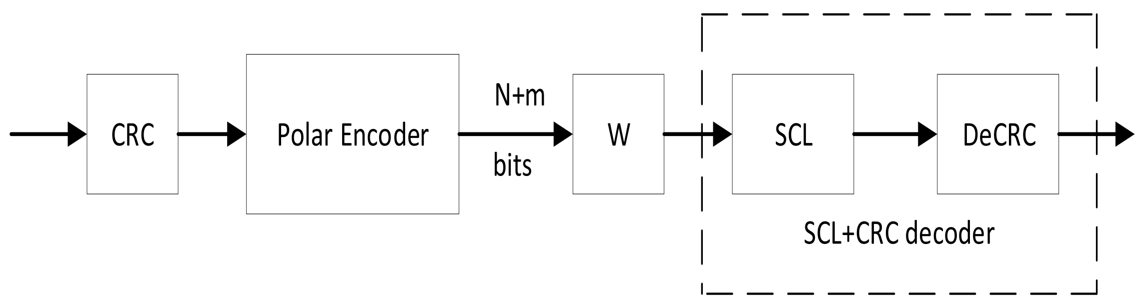

Figure 17 Polar coding and CRC decoding schemes.

4.6 Successive Cancellation Stack (SCS) Decoding

In SCS decoding [51] operation an ordered stack S for storing candidate paths is used and the optimal estimates searching along with the best candidate on the stack is determined.

The decoding process halts and decision is made whenever the top path on the stack having the largest path metric reaches length N . Candidate paths in the SCL decoding operation always have the same length, on the other hand, the candidates in the SCS stack may have different lengths. A modified version of the original SCS is proposed in [63] where an additional parameter L is introduced to control the number of increasing paths of a definite length in the decoding operation.

4.7 Simulation Results

Figure 18 shown the performance of polar codes under SC decoder at block length 2^{10} and 2^{11} in terms of bit error rate (BER) when the communication takes place over the binary erasure channel (BEC) with erasure probability 0.5.

In Figure 19, we compare the performance of polar codes under classical successive cancellation decoder and performance of successive cancellation list using CRC. These results were received when the communication takes place over the binary erasure channel with erasure probability 0.5 and block length 64 bits. Figure 20 compares the performance of polar codes under classical successive cancellation decoder and performance of successive cancellation list using CRC. These results were received when the communication takes place over the AWGN channel with rate 0.5 and block length 128 bits.

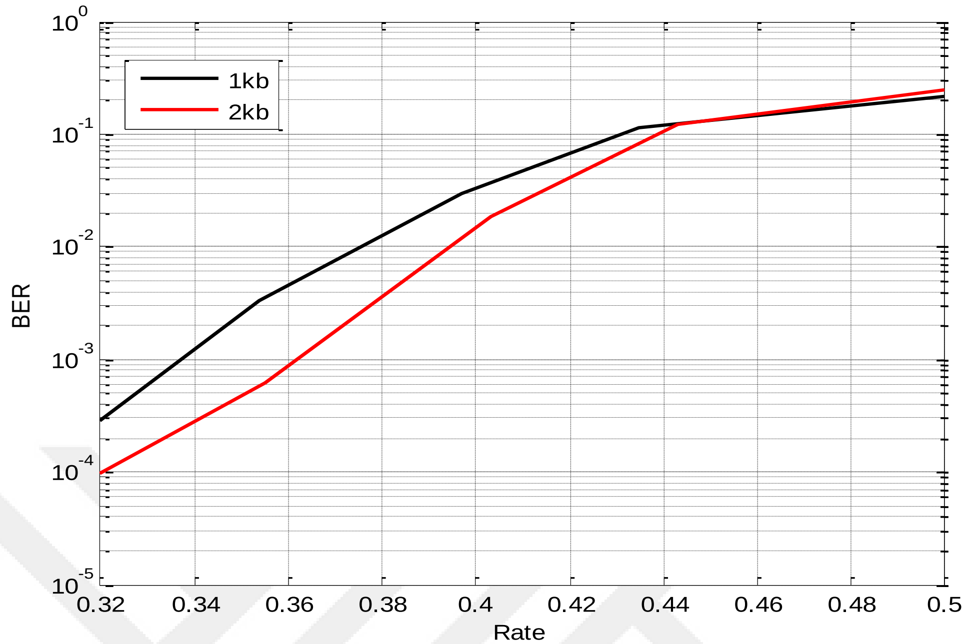

Figure 18 BER performance of polar codes with SC decoding at $N = 2^{10}$ and $N = 2^{11}$ on a BEC with $\epsilon = 0.5$.

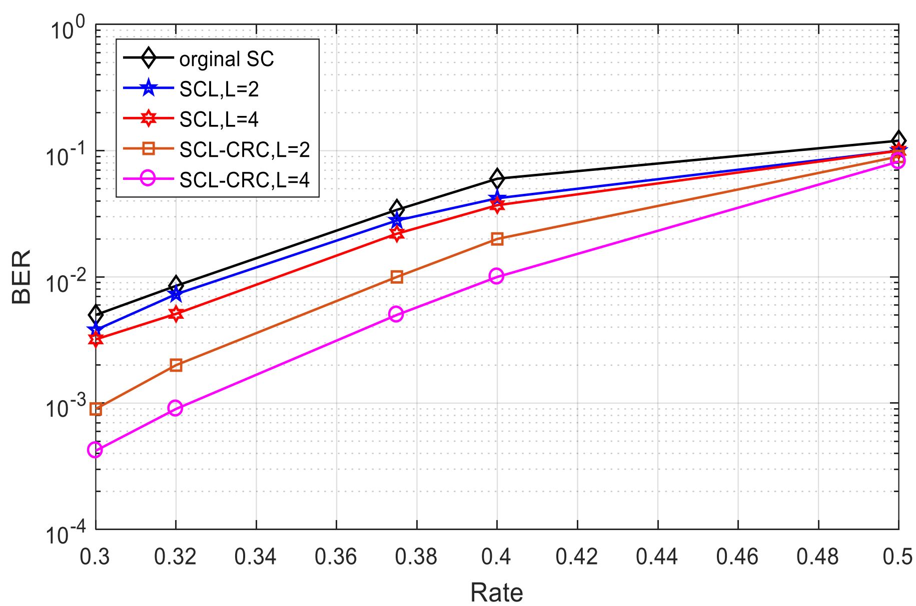

Figure 19 BER performance of SCL using CRC decoder compared to the SC decoder over BEC for $N = 64$ and $\epsilon = 0.5$.

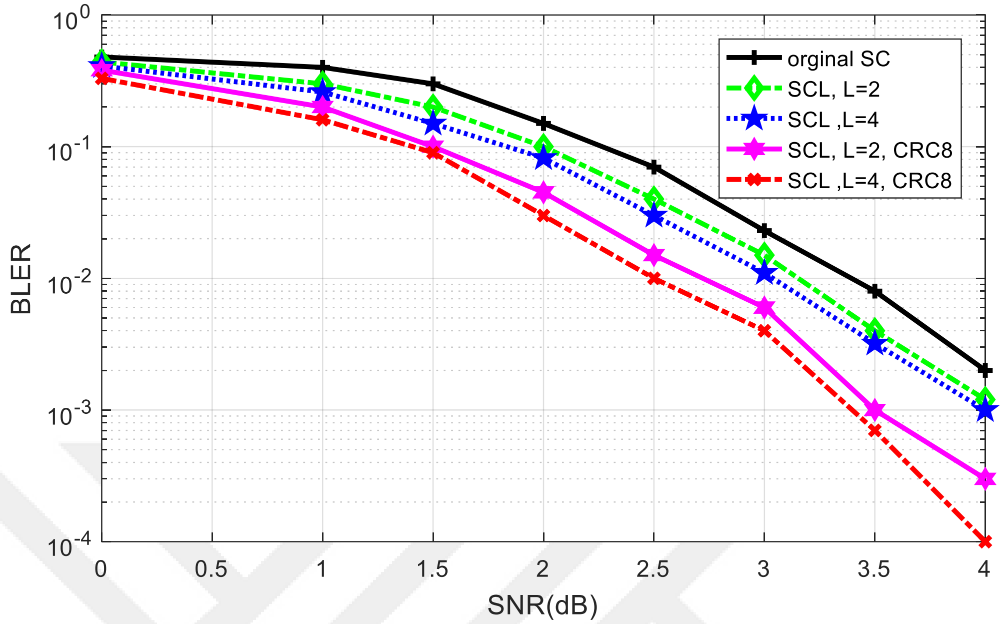

Figure 20 BLER performance of SCL using CRC decoder compared to the SC decoder over AWGN for $N = 128$ and rate 0.5.

Using the proposed approach, we simulated the polar codes with codeword lengths $N = 32$ and $N = 128$ in binary erasure and AWGN channels. For the binary erasure channels, the erasure probability is chosen as $\epsilon = 0.5$. The simulation results for binary erasure channel are depicted in Figure 21 where it is seen that the proposed approach shows better performance than that of the classical successive cancelation algorithm proposed in [32] for short code-word lengths and performances gets closer each other as the code-word length increases. This is the expected result, since (4.20) converges to (4.10) as the code-word length goes to infinity. The simulation results for AWGN channels are depicted in Figure 22. The code rate is takes as 0.5. From the simulation results it is seen that the proposed method shows better performance at small

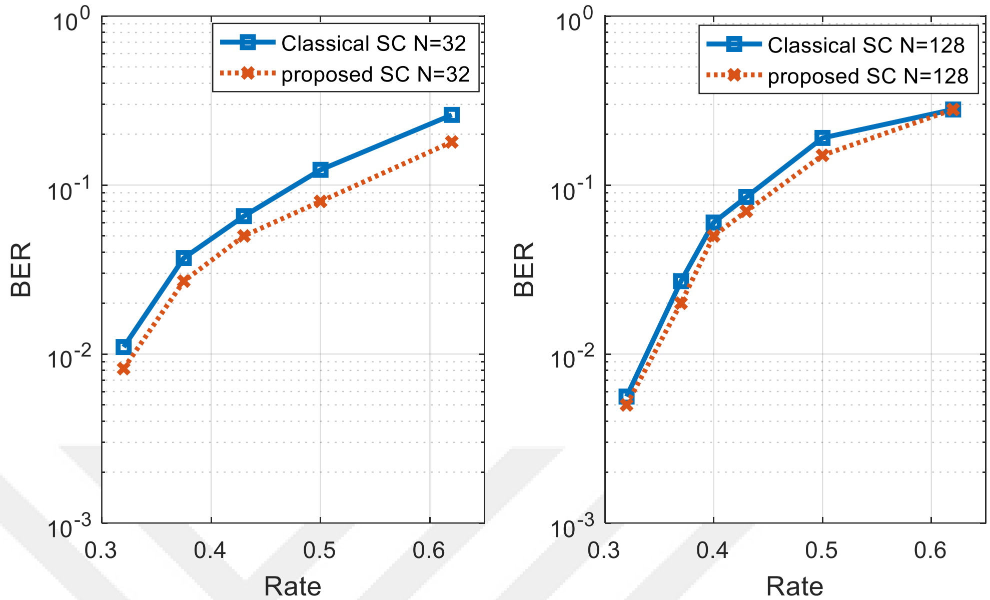

Figure 21 Polar code performance for BEC channel, $N = 32$ and 128, $\epsilon = 0.5$.

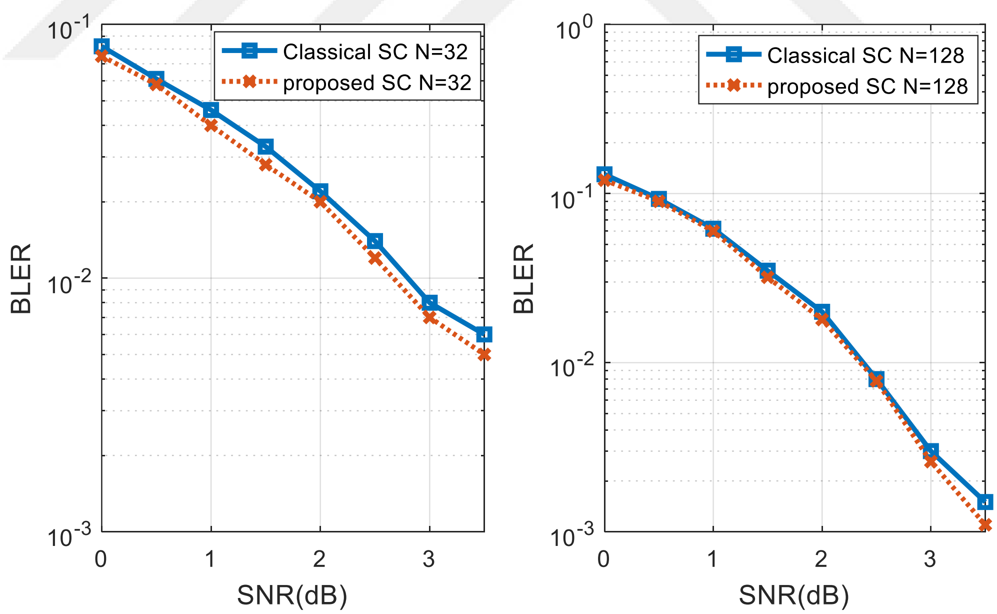

Figure 22 Polar code performance for AWGN channel, $N = 32$ and 128, $rate = 0.5$.

CHAPTER 5

FAST DECODING FOR POLAR CODES

In this chapter, we propose a technique for the fast decoding of polar codes and with the proposed method, it is possible to decode all the information bits simultaneously at the same time, i.e., in parallel. In the sequel, we introduce and improved version of the proposed high-speed decoding algorithm. The proposed high-speed decoding approach and its improved version are simulated on computer environment and their BER performances are compared to the performance of the classical successive cancelation method.

The outline of the chapter is as follows. In Section 5.1, we introduce our proposed high speed polar decoding approach. In Section 5.2, improved version of proposed high-speed polar decoding is introduced. Simulation results are provided in Section 5.3.

5.1 Fast Decoding of Polar Codes

In this section, we propose a novel high-speed decoding technique for successive cancelation polar decoding. The proposed high-speed decoder is based on the introduced tree structure. Our proposed algorithm can decode N successive bits at the same time. Therefore, the latency can be reduced and speed can be increased. The presented N-bit-decoding algorithm is able to decode N bits at the same time (in parallel), i.e., it can decode n^{th} bit without the need of $(n-1)$ previously decoded bits.

As mentioned before, using the tree structure, we can divide the decoding operation into two parts, distribution of the previously decided bits to the nodes, and calculation of node LRs for the decoding of current bit. In the bit distribution stage, the tree is divided into

$log_2(N) + 1$ levels, where N is frame length a power of 2. The index of the top level is 0, and the index of the bottom level is $log_2(N)$. When the previously decoded information bits are distributed to the nodes, we see that nodes at certain levels receive the distributed bits, i.e., g-nodes appears at certain levels. Let us call the levels where g-nodes appear as active levels. In fact, for the distribution of l previously decoded bits, the active levels can be determined using

$$l = \sum_i 2^i \tag{5.1}$$

where i denotes the active level index. The g-nodes in active levels have labels 0 or 1. As an example:

Assume that $N = 8$, and the first 5 bits are decoded. To start the decoding of 6^{th} bit, the previously decoded 5 bits are distributed into the nodes, and the number 5 can be written as

$$5 = \sum_i 2^i \rightarrow 2^0 + 2^2 \tag{5.2}$$

where powers of 2 indicates the locations of g nodes. i.e., they indicate the active levels and the levels with indices 0 and 2 are active levels, and the nodes in these levels are all g nodes, i.e., nodes have bit-labels; the bit-labels either include zero or one. All the other nodes in the tree structure are f nodes which do not have bit-labels. After the first stage of the decoding operation, i.e., distribution of previously decoded bits to the nodes, the second stage of the decoding operation starts. In the second stage of the decoding operation, LRs of the nodes in each level starting from bottom level to top level are calculated.

5.1.1 High-Speed Decoding

For an N-bit information sequence, after channel splitting operation, we have N new channels. For these N channels it is seen that most of the low capacity channels occurs in the first half, i.e., low capacity channels have indices $1, \dots, N/2$. And most of the frozen bits are assigned to these low capacity channels. This is the main motivation of our study. Although some of the channels in the second half have low capacities, their number is few considering the total number of channels. For this reason, we dedicate the first group containing $N/2$ channels to the frozen bits, and use the second group containing the rest of the $N/2$ channels for the data bits. This means that the decoding operation starts for the data bits transmitted through the second part of the communication channels. The number of frozen bits and their percentage considering different code rate is tabulated in Table 1 where it is seen that for code rate 0.5, there is no frozen bit in second half of the information frame.

When $N/2$ frozen bits are distributed on the tree, it is seen that $N/2$ nodes get ’0’ as node-bit. Besides, these $N/2$ nodes exist on the same level which we call as frozen level. When the bit distribution is performed for the decoding of consecutive data bits, it is seen that these $N/2$ nodes always contain ’0’s as node-bits, i.e., frozen level stay the same. And some levels depending on the order of the data bit to be decoded become active levels, and the nodes in these active levels either contain ’0’ or ’1’, and these nodes are nothing but g nodes. The likelihood ratio at these nodes can be calculated separately for node-bit ’0’ and node-bit ’1’ and the larger can be selected to be used for the nodes at upper levels. And this logic can be carried till the top level. In this way, we do not need to know the previously decoded bits, but just need to know the active and frozen levels.

Table 2 Percentage of frozen bits in second half vs code rate.

Rate	Number of frozen bits	Percentage of frozen bits
0.5	0	0
0.434	67	0.13
0.41	91	0.177
0.37	130	0.2539
0.359	144	0.281
0.33	168	0.328
0.32	179	0.349

***Example*:**

Let's illustrate the idea with an example. Assume that $N = 1024$. In this case, the indices of most of the low capacity channels appears in [1 2 3 ... $N/2$]. For this reason, we can freeze the first 512 channels and start decoding the bits with indices greater than 512. Assume that we want to decode the bit with index number 517. Then, we distribute the first 516 decoded bits, 512 of them are frozen bits, to the tree nodes. When the first 516 decoded bits are distributed to the nodes, we get a graph like in Figure 23 where it is seen that level-2 and level-9 active levels, i.e., g. nodes appear in these levels. This is expected since $517 = 2^2 + 2^9$, i.e., powers of 2 indicate the active level indices. The nodes in level-9 contain only 0s. This level can be called as frozen level, since it is filled by zeros via the distribution of frozen bits. On the other hand, the nodes at level-2 may either contain 0 or 1. Since the node bits of level-2 are determined by the distribution of 4 data bits other than the frozen bits, but, we do not need to know the exact values of the node bits at level-2. We can try both 0 and 1 for the calculation of the likelihood for the node output and choose the larger one for the upper nodes. The advantage of the mentioned method is that, we can determine the active levels for the decoding of data bits and data bits can be decoded in a parallel manner without needing the previously decoded bits. Hence, if sufficient place can be found in an electronic device like FPGA, it is possible to decode all the data bits in a concurrent manner. The fast decoding operation can be outlined as in the Algorithm-2.

Algorithm 2: High speed decoding of polar codes.

Input N frame length and received symbols.

1: Determine active frozen level.

2: Let $r = \frac{N}{2} + 1$.

3: Determine active levels for r.

4: Use g-nodes for active levels.

5: Use f-nodes for inactive levels.

6: For g-nodes appearing in the active levels other than the frozen level, use 0 and 1 as bit nodes separately.

7: Calculate LR from the bottom of the tree to the top, use maximum of the two likelihood values for g-nodes.

8: Determine the bit u_r.

9: Increment r, i.e., $r = r + 1$.

10: Go to step-3.

11: When $r = N$ stop.

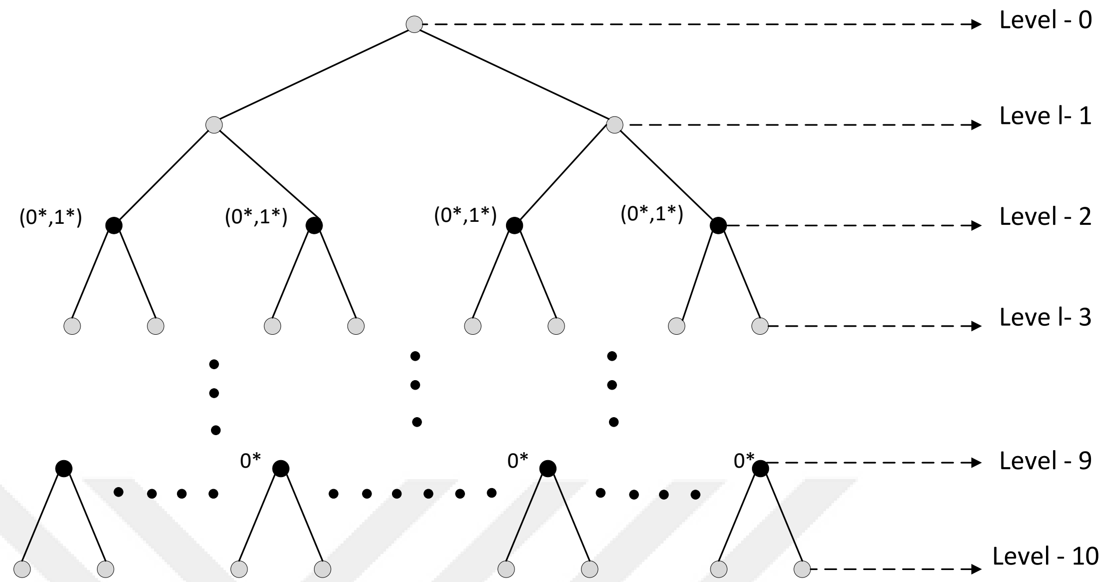

Figure 23 Active and frozen levels for the decoding of data bit u^{517}.

5.1.2 Computation Complexity of The Proposed Approach

The total number of f nodes and g nodes appearing during the decoding operation is the same and it is equal to

$$K_N = \sum_{i=0}^{n} 2^i$$

where $n = \log_2 N$. For g type nodes, we calculate the probabilities for both $u = 0$ and $u = 1$ separately and choose the one containing more information. For this reason, in our approach, the computation complexity for g type of nodes doubles. However, it stays the same for f type nodes. If we indicate the total computational complexity of the SC decoding approach by $O(2K_N)$ where $O(\cdot)$ is the *big-O* notation, then the computational complexity of the proposed method can be expressed by $O(3K_N)$. On the other hand, if we indicate the latency of the SC decoding algorithm by $O(N)$ where N is the number of data bits to be decoded, then the latency of our proposed technique can expressed by $O(1)$.

5.2 Improved High-Speed Decoding

In order to improve the performance of proposed high-speed decoder, we introduce a new decoding scheme called improved high-speed decoding technique. In this new approach, some percent of the information bits are decoded using the classical successive cancelation method and the rest of the bits are decoded using the proposed parallel decoding approach. With this approach, we aim to increase the BER performance of the communication system. An example of the new approach for $N = 1024$ is depicted in Figure 24 where we froze the first $N/2$ bits which corresponding to low capacity channels, and we employed successive cancelation decoding approach to decode the 50% of the rest of the information bits, finally, the remaining information bits are decoded in parallel at the same time using the proposed approach.

Frozen part	Use classical SC decoding for 50% of the information bits	Use the proposed high speed decoding for the rest of 50% of information bits

512 Frozen bits	Decode 256 bits using classical SC decoder	Decod 256 bits using the proposed high speed decoding technique

Figure 24 Improved high-speed decoding approach.

With this new proposed method, the latency of the previously proposed parallel decoding technique is increased, however, we obtain better performance considering all parallel decoding approach. In addition, the total latency is still much less than the latency of successive cancelation method. Since the low capacity channels occurs in the first half of the data frame, by freezing the first half of the transmission frame we prevent the error propagation which affects the decoding of the second half, and by successively decoding some of the bits in the second half, we provide more robust bit decisions which will be used for the decoding of the rest of the bits, and for this reason we obtain better results. We performed computer simulation using the improved high decoding method employing the frame structure in Figure 24. And performance results are depicted in Figures 25 and Figures 26. In case of using AWGN channel, we decreased the part of proposed high-speed decoding to 128 and 64 bits.

5.3 Simulation Results

In Figure 25, we compare the performance of polar codes using the proposed high-speed decoding approach and SC decoding at block length 2^{10} for BECs with erasure probabilities 0.1, 0.2, 0.3, 0.4 and 0.5 in terms of bit error rate BER. From simulation results, we can see that proposed high-speed decoding algorithm gives good performance at low rates, especially for the rates between 0.35 and 0.5. We also performed computer simulations using the improved high decoding method employing the frame structure in Figure 24. And performance results are depicted in Figure 26. It is seen from Figure 26 that at block-length 1024 bits, the improved highspeed decoding method gives better performance with reduced latency over the binary erasure channel BEC with erasure probability 0.5, especially for the rates between 0.35 and 0.5. Figure 27 and 28 shows the performance of improved high speed decoding method and high speed decoding over AWGN channel.

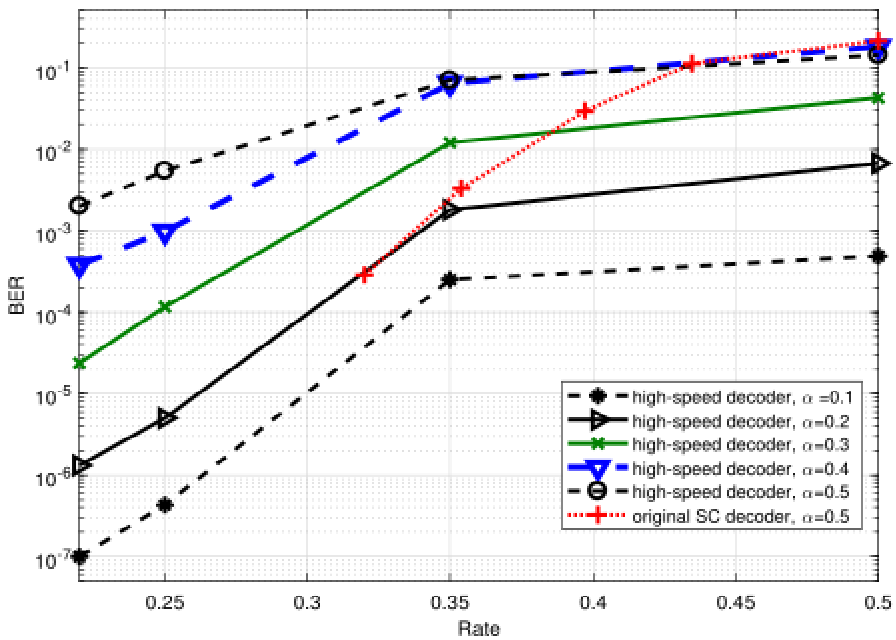

Figure 25 BER vs rate performance comparison.

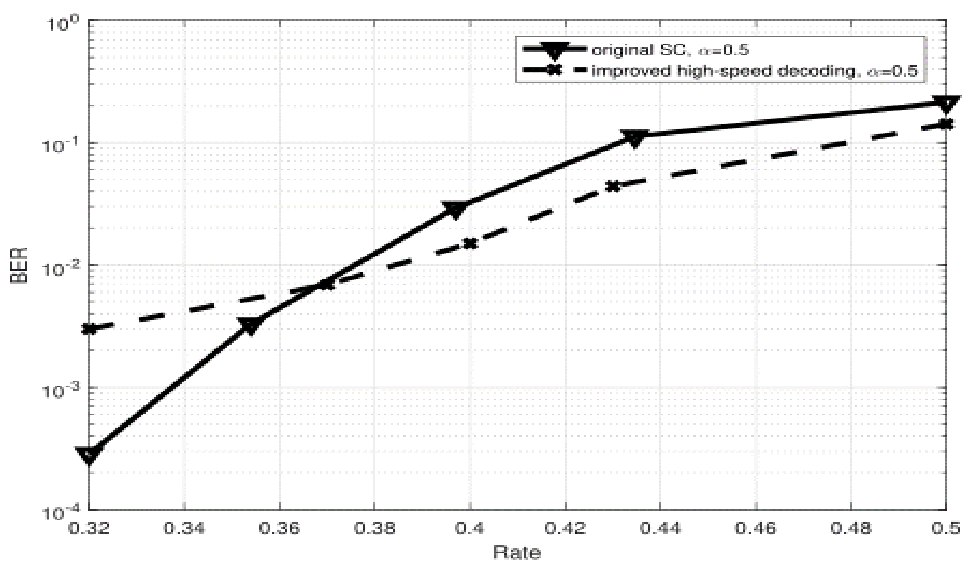

Figure 26 Performance of improved high-speed decoding vs high speed decoding.

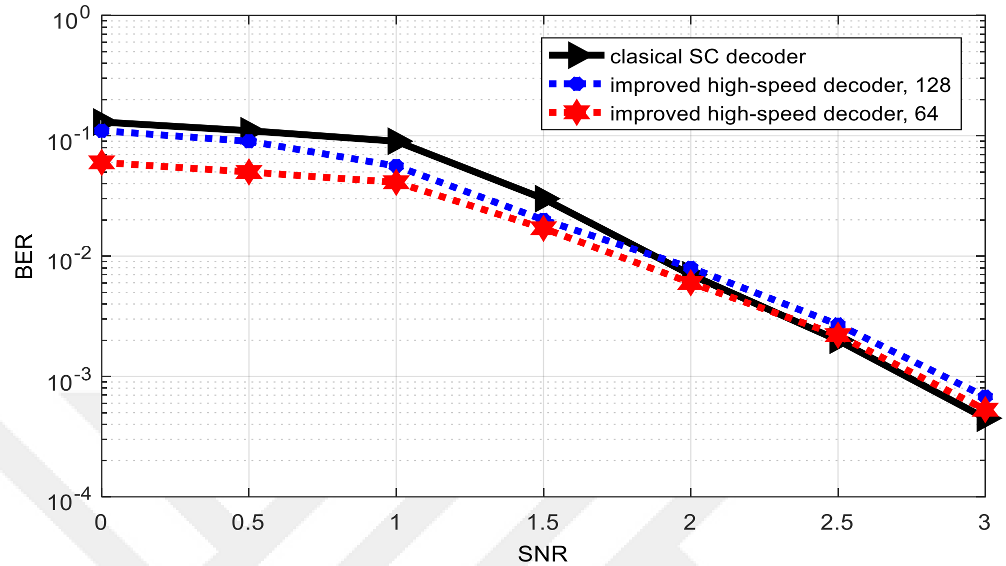

Figure 27 BER vs rate performance comparison for improved high-speed decoder and original SC over AWGN channel.

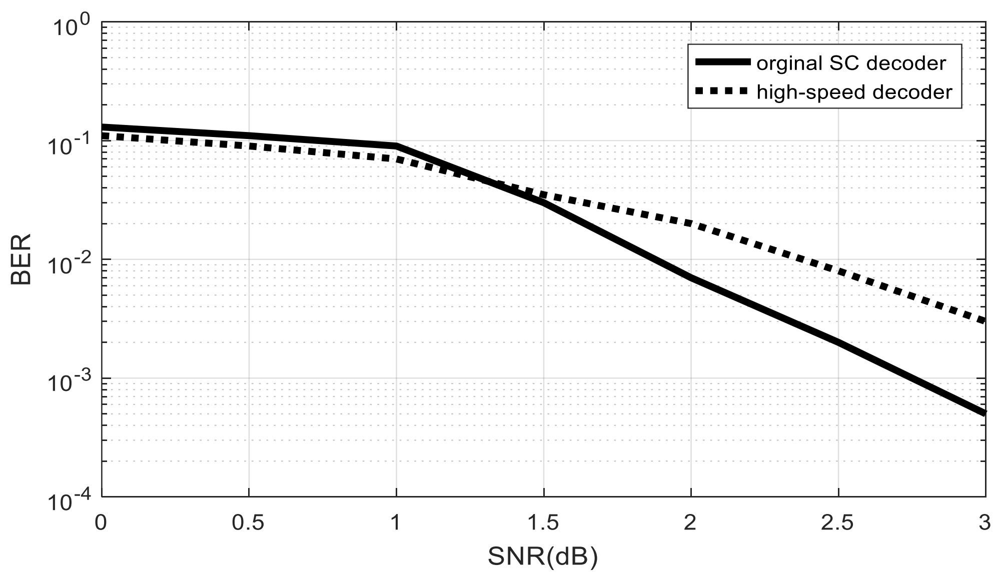

Figure 28 BER vs rate performance comparison for high-speed decoder and original SC over AWGN channel.

CHAPTER 6

EFFECT OF ERROR PROPAGATION ON PERFORMANCE OF SUCCESSIVE CANCELLATION DECODER

Polar codes are decoded in a sequential manner using successive cancelation algorithm introduced in Arikan's original work [32]. The sequential nature of the decoding process suffers from error propagation. In this chapter, we inspect the effects of error propagation on the performance of polar codes and propose some methods to alleviate the degrading effects of error propagation on the code performance for short and long frame lengths.

The outline of this chapter is as follows. In Section 6.1, we propose a short method to determine the values of node-bits in the decoder tree. discussed and analysis the effect of error propagation on the performance of SC decoding in Section 6.2. In order to improve performance of SC decoder we introduce a new approach in Section 6.3. We have introduced training based approach decoding for polar codes in Section 6.4. Simulation results are presented in Section 6.5.

6.1 Determination of Node-Bits

In this section, we propose a short method to determine the values of node-bits in the decoder tree used for the decoding of bit u_{k+1} where $k \epsilon [0 \cdots N-1]$. For this purpose, we first determine the node bits, then calculate the likelihoods of the nodes starting from the bottom ones till the top-most node. For the determination of the node bits, we first write the integer k as sum of powers of 2, i.e.,

$$k = \sum_{i} 2^{i} \tag{6.1}$$

where i refers to the levels whose nodes have assigned bits. Once we determine the level indices i, we partition the previously decoded bit stream starting from the last bit into consecutive sub-streams $\bar{u}_i$ containing 2^i bits, and the node bits are determined using:

$$\bar{n}_i = \bar{u}_i \times G_i \tag{6.2}$$

where G_i is the sub-generator matrix of size $2^i \times 2^i$.

Example:

Assume that $N = 16$ and we want to decode u_{13} and the previous 12 decoded bits are $\bar{u} = [100101110011]$. We can write 12 as $12 = 2^2 + 2^3$, and obtain the sub-streams as $\bar{u}_2 = [1001]$ and $\bar{u}_3 = [01110011]$. Then the node bits are calculated as

$$\bar{n}_2 = \bar{u}_2 \times G_2 \rightarrow \bar{n}_2 = [1001] \times G_2$$
$$\bar{n}_3 = \bar{u}_3 \times G_3 \rightarrow \bar{n}_3 = [01110011] \times G_3$$

6.2 Sequential Decoding and Error Propagation

In successive cancelation decoding of polar codes, for the decoding of $(k+1)^{th}$ −bit we benefit from two kinds of information sources. One is the soft information obtained from the received N −signal, i.e., soft information obtained from the output of the N −channel. The other is the k decision results obtained from the decoding of the previous k bits. This means that the wrong decisions made for the decoding of previous bits affect the decoding of current bit, i.e., bit error propagates throughout the decoding operation.

6.2.1 Bit Errors in Even and Odd Locations

Decoding tree can help us to visualize the distribution of the previously decoded bits to the nodes. For instance, for the decoding of u_8, the distribution of 7 decoded bits $u_1, u_2, \cdots, u_7$ can be achieved using the sub-generator matrices G_1, G_2, G_4, and we get the decoding tree as in Figure 29.

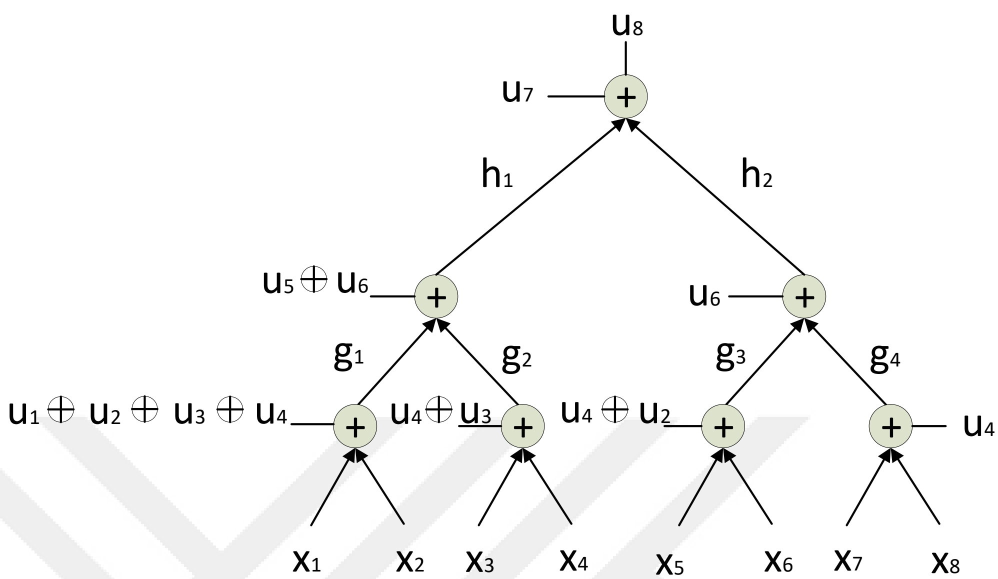

Figure 29 Decoding of bit u_8 for polar codes when $N = 8$.

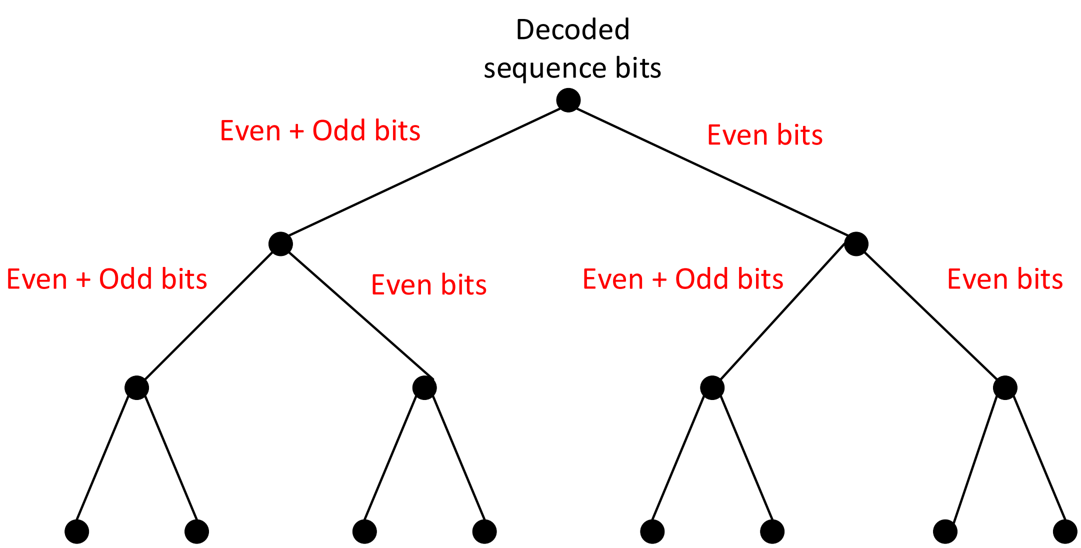

Figure 30 Diffusion of erroneous even bit.

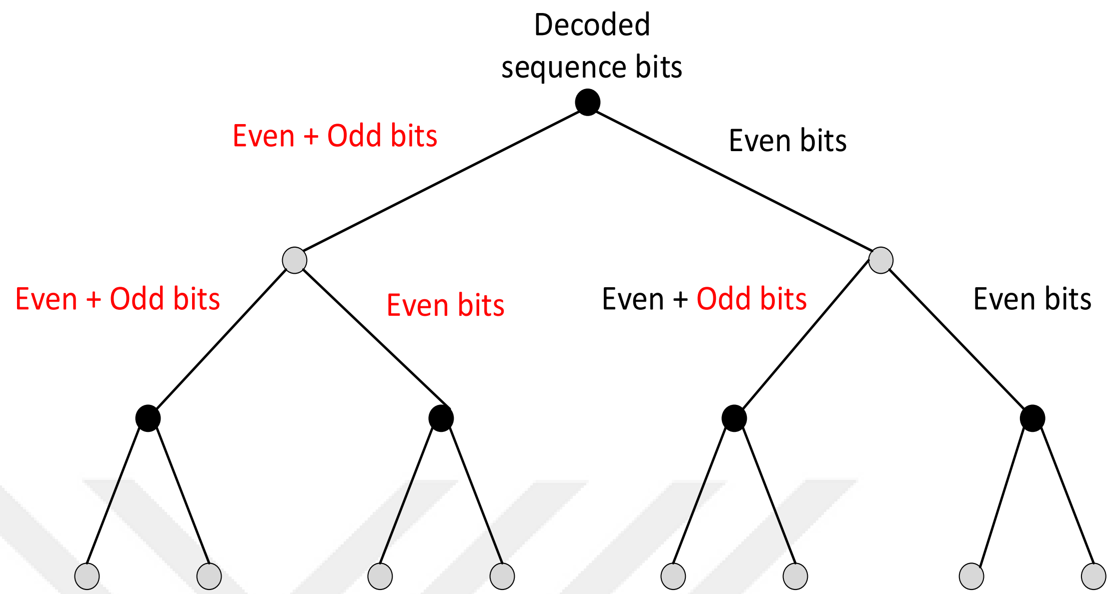

Figure 31 Diffusion of erroneous odd bit.

When Figure 29 is inspected in details, we see that even indexed bits appears at every node-bit combination, on the other hand, odd indexed bits appear only in some of the node-bit combinations. In addition, the bit u_4 appears in every node bit combination at level-2. It is obvious that a wrong decision on the value of bit u_4 affects the probability calculation of the all the nodes of level-2, and nodes at upper levels receive wrong probability values. To test the effects of even and odd indexed bit errors, we introduced single bit errors without BEC, and obtained BER graphs via computer simulations. The obtained graph is depicted in Figure 32. It is clear from Figure 32 that the even indexed bit errors have more degrading effects on code performance. In Figure 32 we compare the performance of SC in two cases, the first case when the first error occurs in odd bits' locations and second case when the first error occurs in odd bits' locations. For more details, see Algorithm 3. For instance, the distribution of the even and odd indexed previously decoded data bits to the nodes of the decoding three for the decoding of current bit is depicted in Figures 30 and 31 where it is seen that even indexed data bits appear in nodes labels more than the odd indexed data bits.

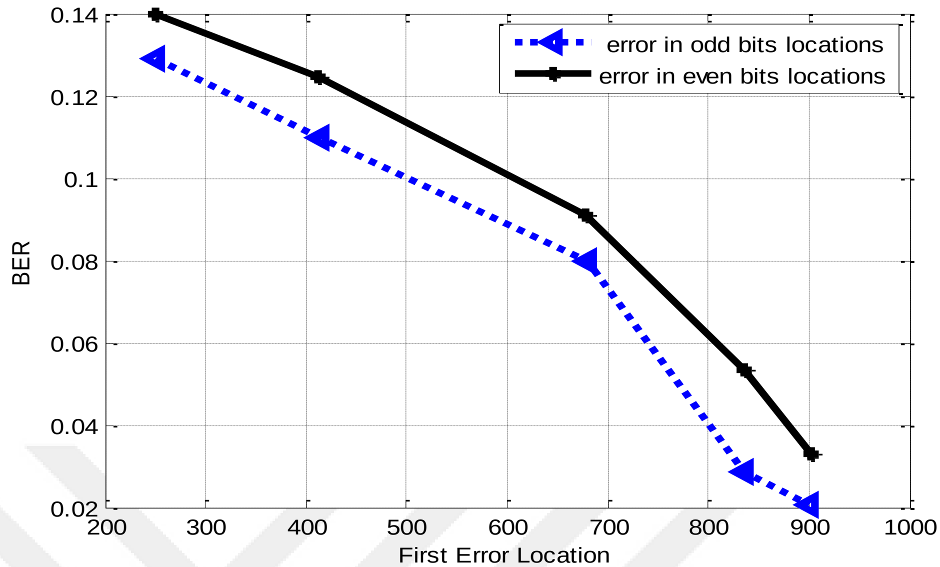

Figure 32 Error propagation vs error location, $Rate = 0.5, N = 1024.$

Algorithm 3: Determining the effect of error propagation in SC decoding
input N frame length, received bits and first error location l. 1: Let $i = 1\ to\ N$. 2: **If** $i \in A^c$, $\hat{u}_i = u_i$. 3: **If** $i \notin A^c$, calculate LR from the bottom of the tree to the top. 4: Use (4.16) for g-nodes. 5: Use (4.15) for f-nodes. 6: Determine the bit u_i. 7: **If** $i = l$, $u_i = u_i \oplus 1$, $i.e.$, create an error. 8: Increment i, i.e., $i = i + 1$. 9: Go to step 2. 10: When $i = N$ stop.

6.3 Successive Cancellation Decoding Improvement

In previous section, we demonstrated that, the performance of successive cancellation decoder of polar codes is affected from the first error locations. We can see that polar codes under SC have better performance when the error occurs in bits which have odd locations. In order to improve performance of SC decoder we propose a new approach. In this new method, after creating an error in one of decoded bit, we distribute the previously decoded bits on the code tree, g-nodes at the top of the tree carries two possible choices 0 and 1, we note the two inputs LRs of every node (α, β) as shown in Figure 33. By using the recursive formula (6.3) for g- nodes and selecting the significant likelihood ratio of $L0$ and $L1$, we can find u_i. For the better understanding of the proposed approach see Algorithm 4.

$$LR = \begin{cases} L0 = (\alpha \times \beta)\ for\ 0 \\ L1 = \dfrac{\beta}{\alpha} \qquad for\ 1 \end{cases} \tag{6.3}$$

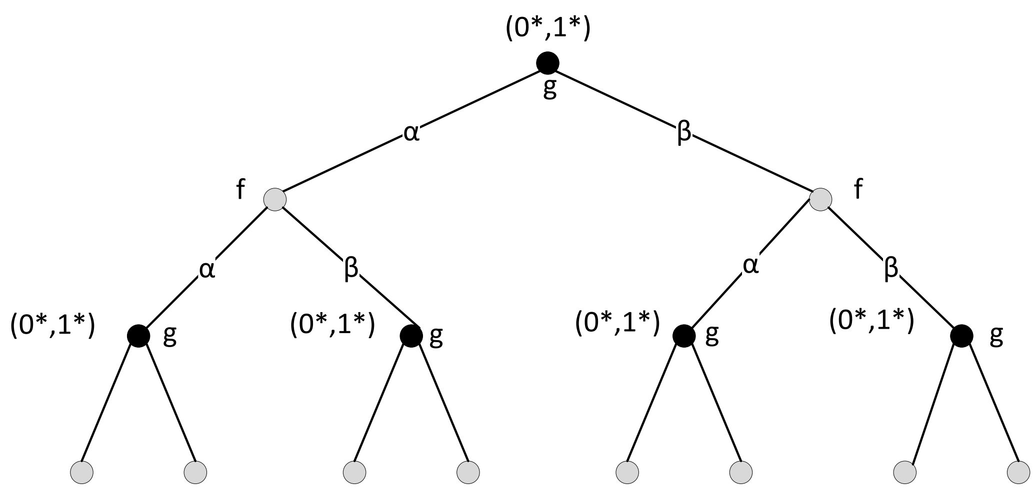

Figure 33 Bits distribution stage of a new approach.

Algorithm 4: Effect of error propagation using proposed improved SC Decoding

Input N frame length, received bits and first error location l.

1: Let $i = 1\ to\ N$.

2: **If** $i \in A^c$, $\hat{u}_i = u_i$.

3: **If** $i \notin A^c$, calculate LR from the bottom of the tree to the top.

4: Use (6.3) for g-nodes.

5: Calculate LR from the bottom of the tree to the top, use maximum of the two likelihood values for g-nodes at the top.

6: Determine the bit u_i.

7: If $i = l$, $u_i = u_i \oplus 1, i.e.$, create an error.

8: Increment i, i.e., $i = i + 1$.

9: Go to step 2.

10: When $i = N$ stop.

6.4 Alleviation of Error Propagation via Training Based Approach

In this section, we introduce a training-based approach for the alleviation of error propagation problem. In our proposed approach, we first extract some statistical information for the most probable error locations. For this purpose, we transmit 50 frames and record the index of first erroneous bit. The statistical data for $N = 32, 64, 128$ and 1024 and rate $R = 0.5$ are plotted as histogram as in Figure 34 and 35 from which we see that the first erroneous bits usually appear at the small capacity channels, and they correspond, in general, to the first half of the data block. For different rates the statistical information is extracted via training approach. Since erroneous bits occurring at even indexes have more degrading effects than the erroneous bits at odd indexes, for $N = 32$ for the first 4 data bits, and for $N = 64$ for the first 6 data bits, which are most probably to 50 the errors, we employ cyclic codes with generator polynomials $g_1(x) = x^6 + 1$, $g_2(x) = x^4 + 1$. The rate of the cyclic codes are $R = 0.5$. The parity bits obtained from the cyclic codes are concatenated to the end of the polar codes as side information.

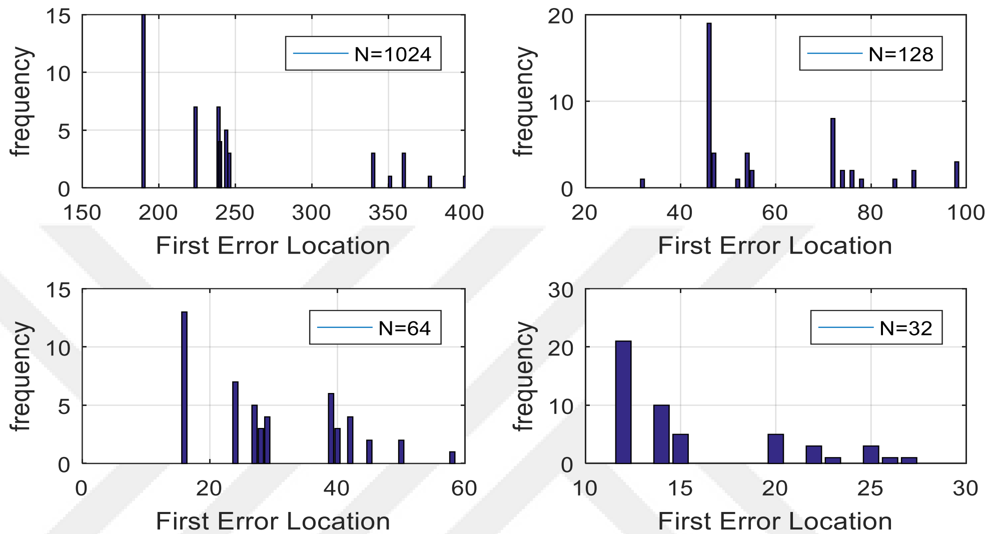

Figure 34 Histogram showing the reputation of fist error location for SC decoder at block length 32, 64, 128, 1024 over a BEC with erasure probability 0.5 and rate 0.5.

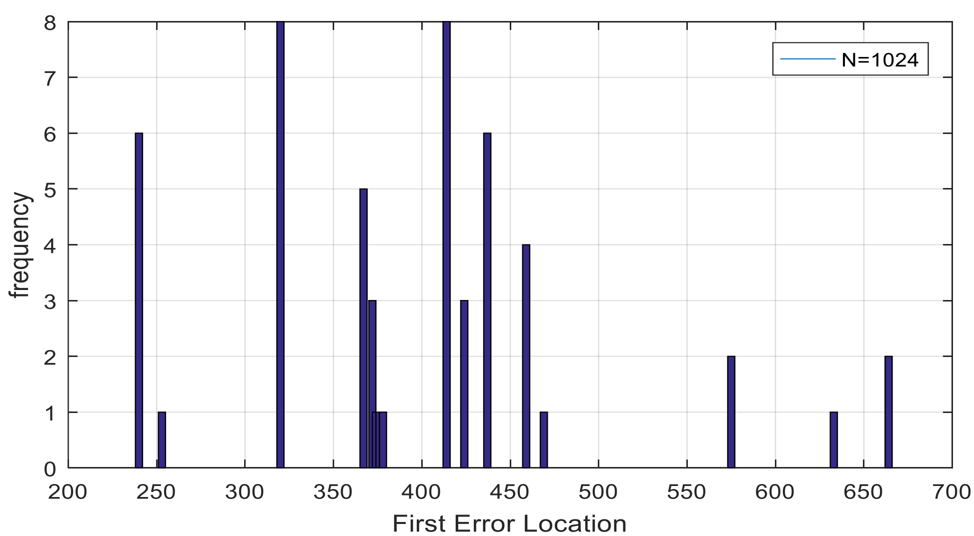

Figure 35 Histogram showing the reputation of fist error location for SC decoder at block length 2^{10} over a BEC with erasure probability 0.5 and rate 0.43.

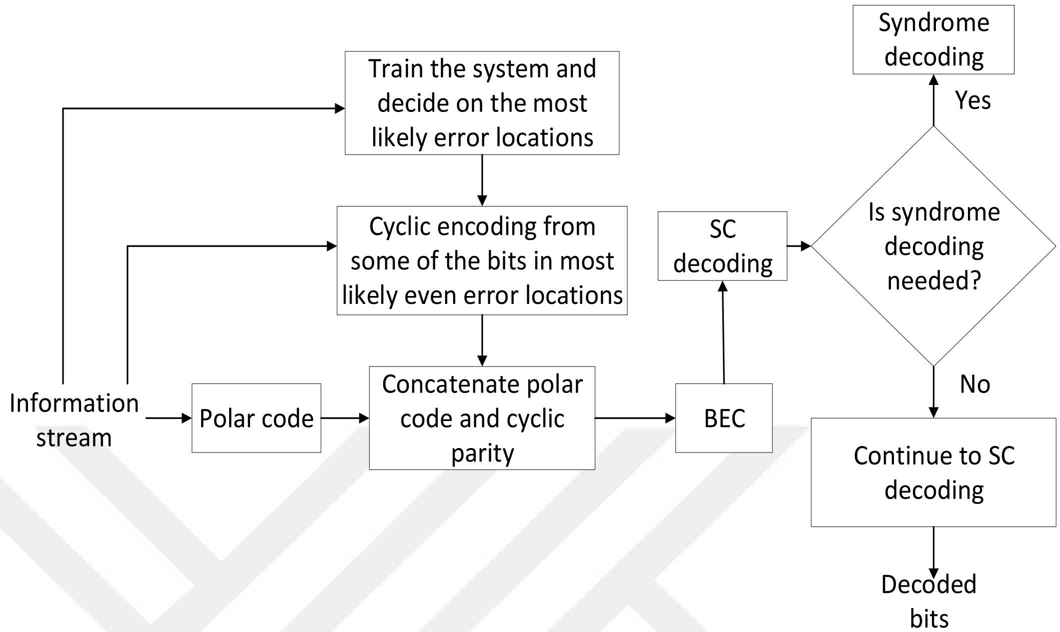

Figure 36 Communication system with CRC.

6.4.1 Cyclic Redundancy Checking (CRC)

Cyclic Redundancy Checking (CRC) is employed for error detection operation. In CRC techniques, a few number of bits calculated from the information stream is appended to the end of the data stream and at the receiver side a computation is performed to detect the presence of any errors occurred on the transmitted bits.

Binary polynomial division method is used in cyclic redundancy checks to detect the errors. For the computation of CRC, first generator polynomial G is chosen. In literature, a number of well-known polynomials are available with good error detecting ability. [63] As an example the generator polynomial $G = (x^3 + x^2 + 1)$ which can be represented with a binary string as $G = 1101$ can be utilized for error detection purposed for the formation of CRC bits. We also know that the first bit of G must always be a 1, so we only need to store $n = 3$ bits, 101. Now assume our original data $S = 100110$. To compute the checksum of S, we must first append n bits to the end to create $D = 100110000$. This is simply binary polynomial division of G into D.

$$\frac{D}{G} = \frac{(100110000)}{1101} = 111001, \text{remainder} = 101$$

So, the $CRC_G(100110000)$ is 101 in binary (5 in decimal). Then our transmitted message S' consists of S concatenated with its CRC, or 100110101. The receiver takes the CRC of what it receives. If there is no corruption, this CRC will be zero, otherwise, an error is detected. In our example, we take $CRC_G\ (100110101) = 0$, but if the last bit is corrupted $CRC_G\ (101110101) = 1$.

6.4.2 Complexity of Proposed SC

The original SC decoding has complexity of $O(N\ log\ N)$. In this proposed algorithm we repeated SC decoding to times in order to find out first error location. Therefore, the complexity will be $O(2N\ log\ N)$. However, this complexity decreases with decreasing of the rate, in low rate the complexity of proposed algorithm proximately equal $O(N\ log\ N)$.

6.5 Simulation Results

In Figure 37, simulation results show the effect of error propagation on the performance of polar codes under a new approach decoder at block length 2^{10} over a BEC with erasure probability zero and rate 0.5. Figure 38 and Figure 39 show the effect of first error location (odd index and even index) on performance of polar codes under SC decoder and under the new approach at block length 2^{10} over a BEC with erasure probability 0.5 and rate 0.5.

We did our simulations for BEC with erasure probability $\alpha = 0.5$. The even indices for the data bits for rate $R = 0.5$ and $N = 32$ are chosen as [12, 14, 20, 22], and they are chosen for rates 0.43, 0.37, 0.32 as [14, 16, 20, 22], [16, 22, 26, 28], [16, 24, 26, 28] respectively. In a similar manner using a training based approach, the even indices for rate

$R = 0.5$ and $N = 64$ are chosen as [16, 24, 28, 40, 50, 58], and they are chosen for rates 0.4, 0.37, 0.32 as [24, 28, 30, 40, 50, 52], [28, 30, 40, 46, 50, 52], [30, 40, 44, 46, 50, 52] respectively. For the chosen data bits at the even indices, we employed cyclic code with rate $R = 0.5$, and the parity bits are concatenated to the end of the polar codeword. At the received side, SC algorithm is run, and when the decoding of the chosen data bits at even indices are complete, a check is performed for the cyclic parity bits. If any error in the chosen bits are detected, syndrome decoding is performed for the chosen data bits and decoding operation is continued for the rest of the bits. The proposed system is depicted in Figure 37.

We also considered the effects of double errors. In Figure 40, the effect of double errors on even locations only, on odd locations only, and one-bit error on even and one-bit error on location are considered. It is clear from Figure 40 that double error at even locations have the most degrading effect, and double errors at odd locations have less degrading effect on the code performance.

In Figure 41, we compare the effects of single and double errors at even and odd locations. It is clear from Figure 42 that the occurrence of a single error at an odd location almost has the same effect as the occurrence of double errors at odd locations. However, this is not the case for errors occurring at even locations. Double errors occurring at even locations have more degrading effect than a single error occurring at an even location.

We also inspected the effects error propagation for SC list decoding of polar codes. The simulation results are depicted in Figure 42 where it is clear that for SC list decoders, errors occurring at even location have more degrading effects than the errors occurring at odd locations.

The simulation results for binary erasure channel are depicted in Figures from 43 to 46 where they are seen that the Training Based Approach shows better performance than that of the classical successive cancelation algorithm proposed in [32]. This is the expected result, since employing cyclic codes for the most probable even error locations, we alleviate the degrading effect of error propagation, and even for very frame lengths we obtain significant performance improvement for short sizes.

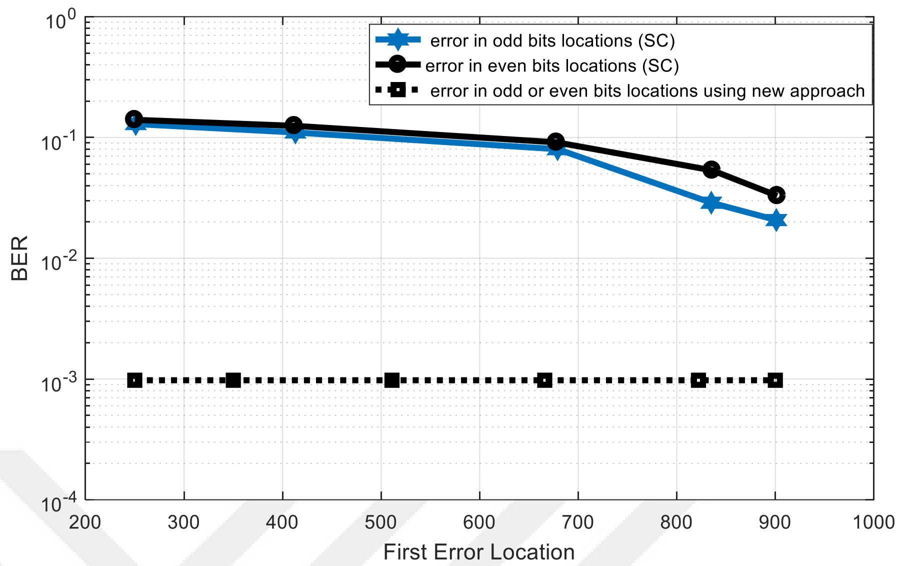

Figure 37 Performance of a new approach decoder compared to the SC decoder block at length 2^{10} over a BEC with erasure probability zero and rate 0.5.

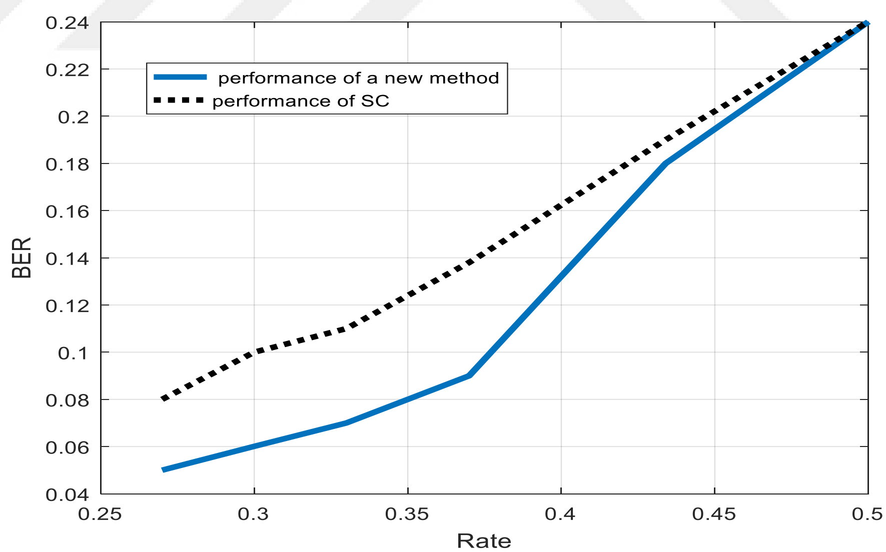

Figure 38 The effect of first error location (odd index) on performance of polar codes under SC decoder and under the new approach at block length 2^{10} over a BEC with erasure probability 0.5 and rate 0.5.

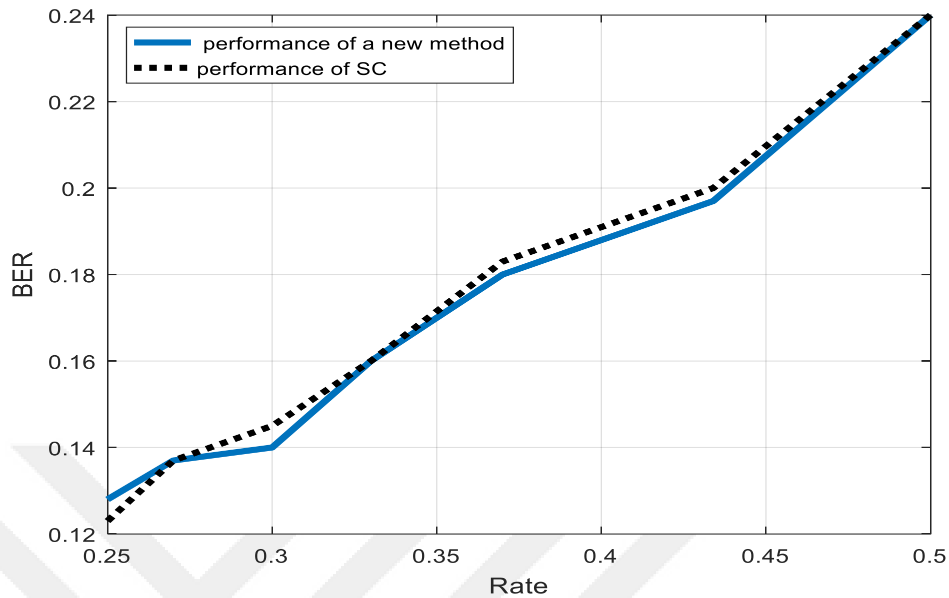

Figure 39 The effect of first error location (even index) on performance of polar codes under SC decoder and under the new approach at block length 2^{10} over a BEC with erasure probability 0.5 and rate 0.5.

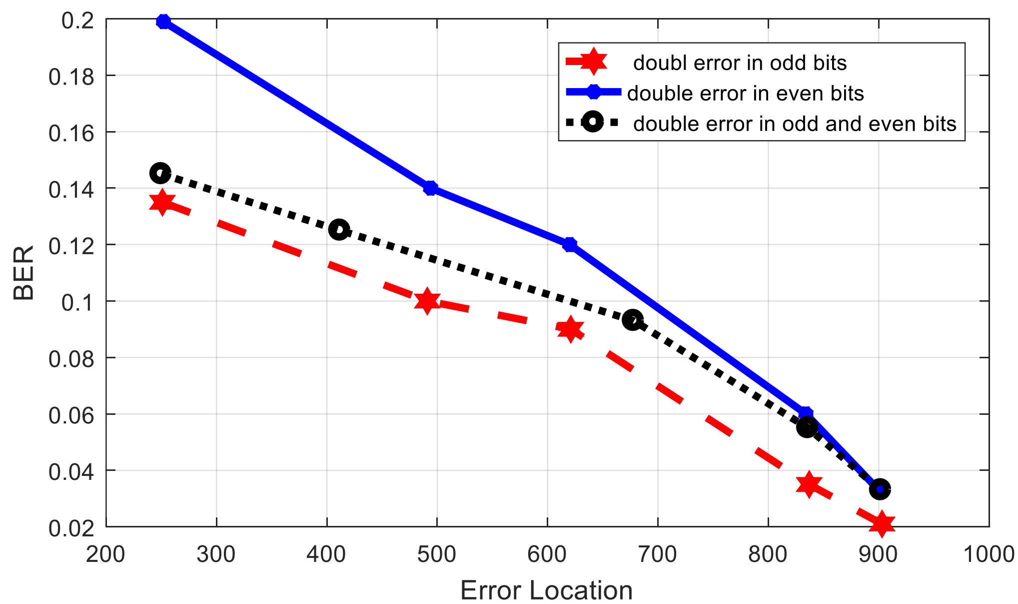

Figure 40 The effect of double errors on even location only, on odd locations only, and one at even and one at odd locations.

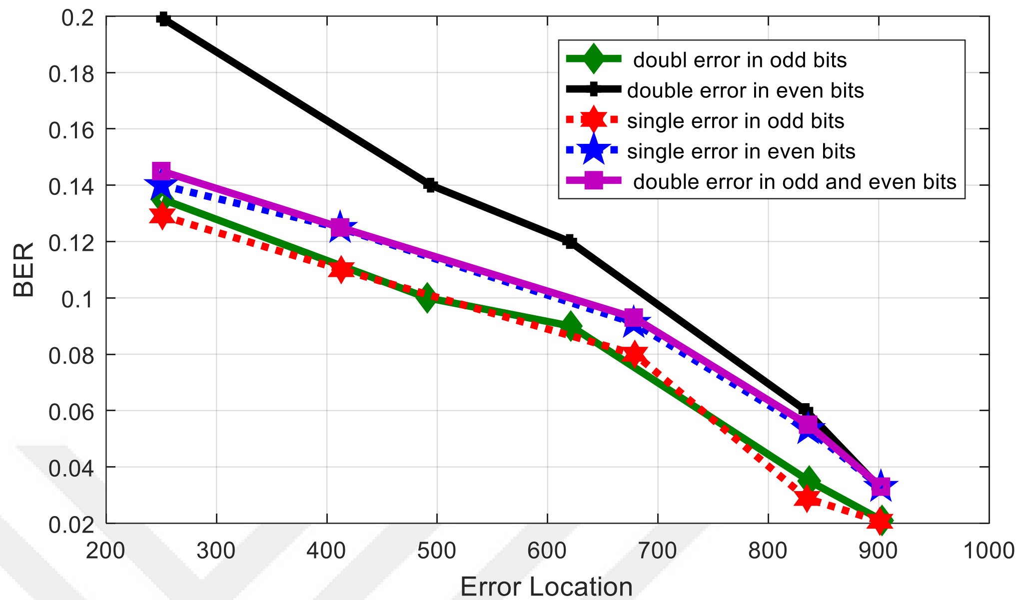

Figure 41 Comparison of the effects of single and double errors on even and odd locations.

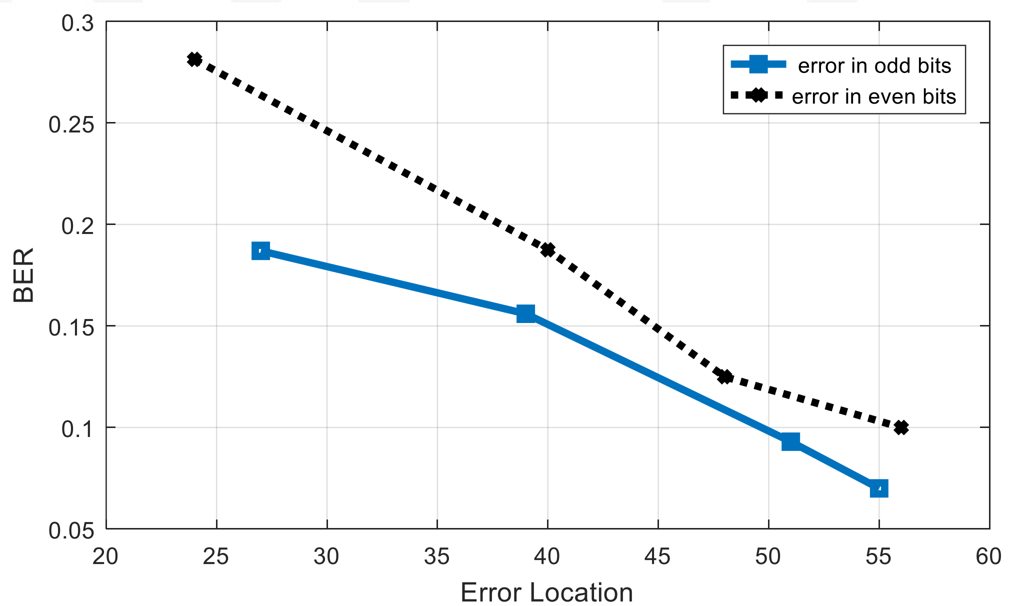

Figure 42 Single error propagation in SCL, $Rate = 0.5, N = 64$.

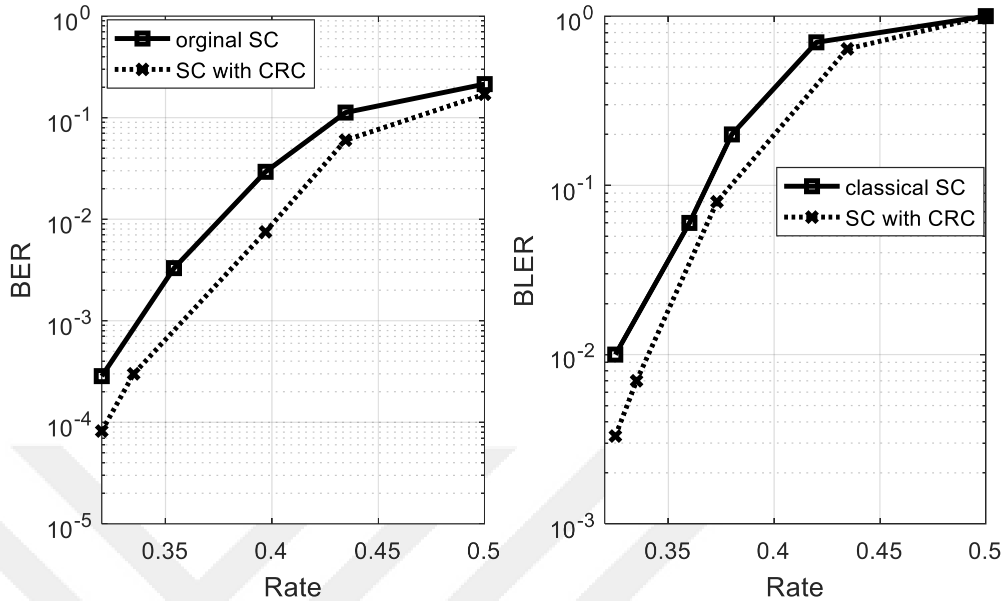

Figure 43 BER and BLER performance of proposed SC using CRC decoder compared to the SC decoder for $N = 1024$ and $rate = 0.5$.

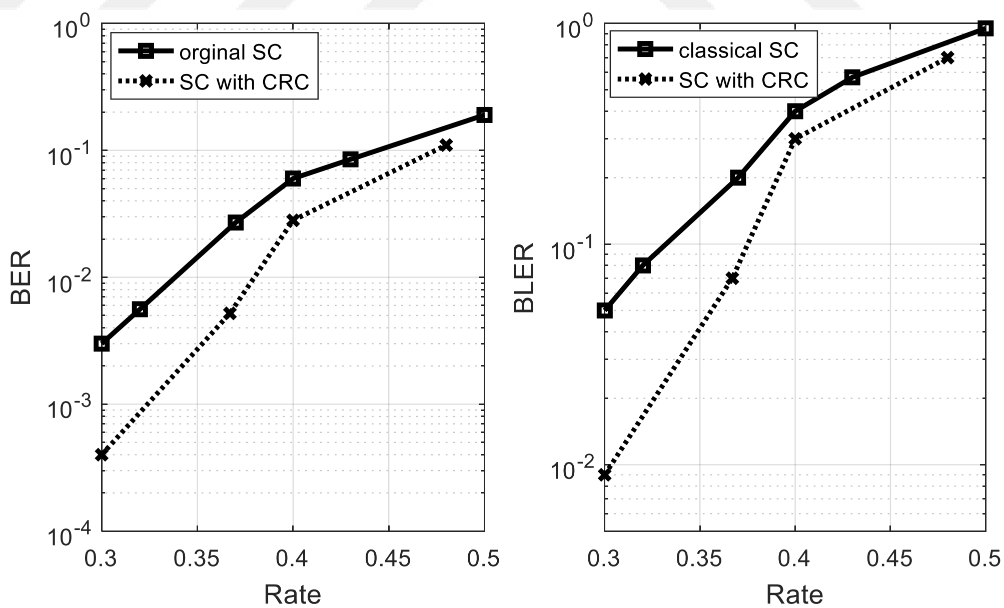

Figure 44 BER and BLER performance of proposed SC using CRC decoder compared to the SC decoder for $N = 128$ and $rate = 0.5$.

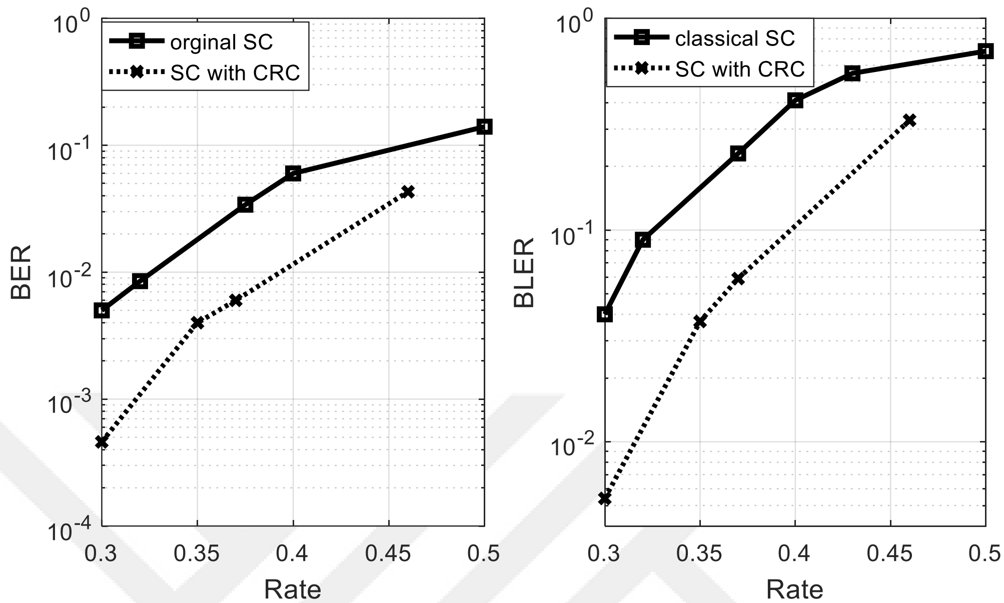

Figure 45 BER and BLER performance of proposed SC using CRC decoder compared to the SC decoder for $N = 64$ and $rate = 0.5$.

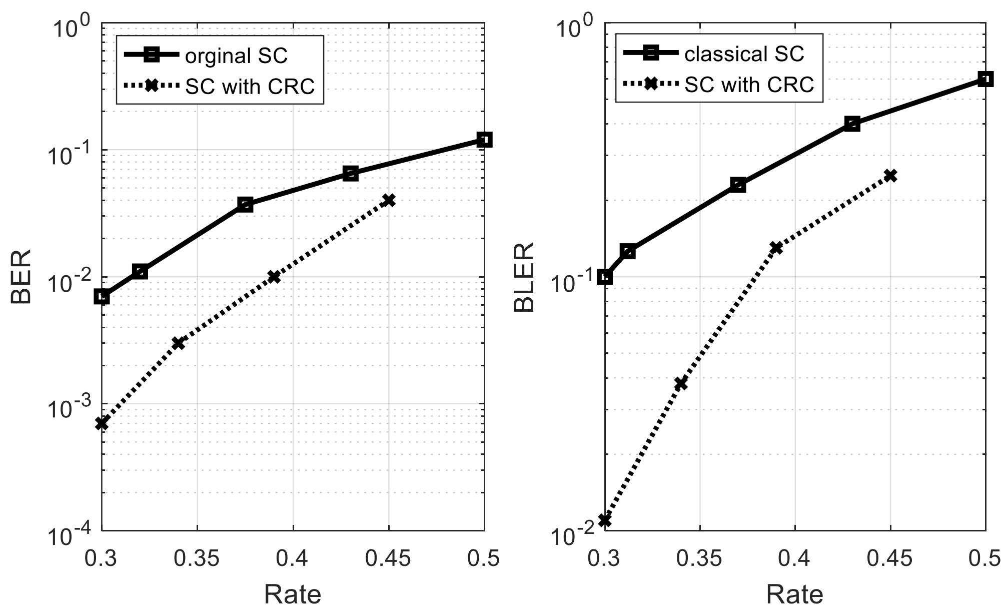

Figure 46 BER and BLER performance of proposed SC using CRC decoder compared to the SC decoder for $N = 32$ and $rate = 0.5$.

CHAPTER 7

CONCLUSION AND FUTURE WORK

7.1 Conclusion

This thesis considers several parts and algorithms for polar codes decoder, including SC decoder, SCL decoder and proposed high-speed decoder. Furthermore, the effects of error propagation on performance of SC decoder are discussed aiming to achieve significant improvement on performance of polar codes.

In this thesis study, we propose a tree structure for the successive cancelation (SC) decoding of polar codes. The proposed structure is easy to implement in hardware and suitable for parallel processing operations. From simulation results in chapter 4, we can see that polar codes give better performance when the block length rises from 2^{10} to 2^{11}. The performance of the SC algorithm can be improved by using SCL decoding algorithm. In this thesis, we proposed a new approach for the calculation of likelihoods used in successive cancelation decoding of polar codes. The proposed approach uses only the soft likelihood values during the decoding process. It is shown via computer simulations that the proposed approach shows better performance than that of the classical successive cancelation algorithm for small code-word lengths. The suggested technique can be used to construct joint communication systems utilizing exchange of soft information rather than hard one. The classical successive cancelation formula in (4.16) can be considered as the limiting case of our proposed formula in (4.23).

A successive cancellation list (SCL) decoding algorithm to improve the performance of polar codes is discussed. Compared with classical successive cancellation decoding algorithms, SCL concurrently produces at most L best candidates during the decoding process to reduce the chance of missing the correct code word. However, the SCL will

increase hardware complexity, which prevents the efficient implementation. This algorithm can reduce the latency significantly without any performance loss.
Combination of list decoding and CRC algorithm is discussed to improve the performance of polar codes further. From simulation results, we can see that SCL does not provide a significant improvement in case of $L = 2$ and $L = 4$. However, simulation results of SCL decoding with CRC over the binary erasure channel show a significant performance improvement, especially at the low rate.

According to simulation results, it is seen than the code length of the polar code N should be chosen large enough to achieve the desired error-correcting performance for practical applications. The decoding of a codeword using SC requires $(2N - 2)$ cycles which make a large latency with large N and it is not suitable for real-time high-speed applications. Therefore, the design of low-latency and high-speed polar decoder is a critical issue especially for practical applications.

In this thesis, we proposed a tree structure for the successive cancelation decoding of polar codes. Using the introduced tree structure, we proposed a high-speed low latency decoding algorithm for polar codes, such that, using the proposed algorithm it is possible to decode all the information bits simultaneously in parallel. BER performance of polar codes using the proposed high-speed decoding method over binary-erasure channels is obtained via computer simulations. From simulation results, we see that proposed high-speed decoding algorithm good performance at high code rates, especially for the rates between 0.35 and 0.5. The proposed high-speed decoding algorithm gives a great improvement in polar codes decoding speed. For frame length $N = 1024$ and $rate = 0.5$, while original SC decoder can decode only one bit for a decoding stage, the proposed high-speed decoder can decode 512 bits all together at the same decoding stage and since in the proposed high-speed decoding algorithm, we do not need to know the previously decoded bits. Next, we suggested an improved version of the proposed high speed decoding technique. The proposed improved high-speed decoding method shows better performance than that of the classical successive cancelation decoding method at high rates with much lower decoding latency. In case of using AWGN channel, proposed high-speed decoder gives better performance at low SNR.

We also proposed a new approach for the alleviation of error propagation that occurs in successive cancelation decoding of polar codes. We discovered that single errors occurring at even indexed bits has more degrading effects than the single errors occurring at odd indexed bit locations. The proposed approach uses a training-based approach to determine the most likely first erroneous bit locations, and employ cyclic codes for the mostly likely even erroneous bit locations. Syndrome decoding is performed at the receiver side in case of error occurs at determined most likely erroneous bits. It is shown via computer simulations that the proposed approach shows much better performance than that of the classical successive cancelation algorithm with a negligible extra overhead, and significant improvements is observed in performance for short frames sizes which are used in practical communication systems.

In Chapter six, we focused on design efficient decoder that reduces the effects of error propagation on performance of SC decoder of polar codes. The results demonstrated that. The performance of SC decoder of polar codes affected by error locations. We see that polar codes under SC have better performance when the first error occurs in bits which have odd indexes. Also, we have introduced a new decoding for polar codes. This algorithm avoids error propagation in the two cases (artificial error in odd and even bits indexes) without erasure channel and improves the BER performance only when first error occurs in odd bits indexes. Furthermore, we have introduced a new decoding using CRC for polar codes. This algorithm improves the bit error rate and block error rate performance by estimating the first error location.

7.2 Future Work

Since their introduction in 2008, numerous improvements have been achievements on the performance of polar codes and improved SC algorithms such as systematic SC and SC list are proposed. However, there are still some problems for the use of polar codes in practical communication systems. Efficient construction of polar codes is an important issue for practical use of them in communication systems. The accurate determination of the frozen bit positions for non-BEC channel is still an open problem. The investigation of efficient selection of frozen bit positions over many practical channel models is an open issue for future works. Second, to overcome the performance of LDPC codes, the list size of SCL decoders should be chosen very large and this leads to a huge cost on silicon area and power consumption. Future research will include the low complexity efficient algorithms and architecture designs employing polar codes with small list sizes showing outstanding error-correcting performances. Third, the performance of the BP algorithm is not comparable to SCL algorithm. However, BP algorithm makes use of the soft information and it is suitable to design joint communication system that can be iteratively processed. For this reason, improved design of BP decoders and joint communication systems involving BP decoders is an open challenge for future studies.

In the following, we motivate some important problems as extensions of our work.

• In Chapter 5, we proposed a technique for the fast decoding of polar codes, we have faced significant problem in this technique. This problem occurs when lost bits comes neighbors to each other, in this case, $LR\ of\ 1 = LR\ of\ 0 = 1$ in next level. So, we could not decide if the bit in this node is 0 or 1. According to previous analysis, we have to find the efficient solution of neighbors lost bits' problem to improve polar codes under high-speed decoder performance.

• In Chapter 6, considering most probably error locations. We can increase the number of most probably error locations that checked by CRC to improve the performance of SC by reducing error propagation.

REFERENCES

1. **Shannon, C. E., (1948b),** "*A mathematical Theory of Communication*", Bell Syst. Tech. J., vol. 27, pp. 623-656. October.

2. **Shannon, C. E., (1959a),** "*Coding Theorems for a Discrete Source with a Fidelity Criterion*", IRE Nat. Conv. Rec., pt. 4, pp. 142-163, March.

3. **Elias, P., (1955),** *"Coding for noisy channels*", in IRE International Convention Record, pp. 37–46, March.

4. **Dobrushin, R. L., (1963),** *"Asymptotic optimality of group and systematic codes for certain channels",* Teor. Veroyat. i primenen., vol. 8, pp. 52–66.

5. **Gallager, R. G., (1968),** "*Information Theory and Reliable Communication*", New York: Wiley.

6. **Goblick, T. J. and Jr., (1962),** *"Coding for discrete information source with a distortion measure*", Ph.D. dissertation, MIT.

7. **Berger, T., (1971)**," *Rate Distortion Theory*", London: Prentice Hall.

8. **Costello, Jr J. and Forney, Jr. G. D., (2007),** *"Channel coding: The road to channel capacity*", Proceedings of the IEEE, vol. 95, no. 6, June.

9. **Bose, R. C. and Ray-Chaudhuri, D. K.,** (1960), *"On a class of error correcting binary group codes",* Info. and Control, vol. 3, no. 1, pp. 68–79, March.

10. **Hocquenghem, A.,** (1959), *"Codes correcteurs d'erreurs",* Chiffres, vol. 2, pp.147–156.

11. **Reed, I. S., (1954),** *"A class of multipl-error-correcting codes and the decoding scheme",* IRE Transactions on Inform. Theory, vol. 4, pp. 38–49.

12. **Muller, E. D., (1954),** "*Application of Boolean algebra to switching circuit design",* IRE Transactions on Electron. Comput., vol. 3, pp. 6–12.

13. **Reed, I. S. and Solomon, G., (1960),** "*Polynomial codes over certain finite fields",* J. SIAM, vol. 8, no. 2, pp. 300–304, June.

14. **Elias, P., (1954),** "Error-free coding," IEEE Trans. Inform. Theory, vol. 4, pp. 29–37, September.

15. **Forney, Jr. G. D., (1966),** "Concatenated Codes", MIT Press.

16. **Viterbi, A J., (1967),** *"Error bounds of convolutional codes and an asymptotically optimum decoding algorithm",* IEEE Trans. Inform. Theory, vol. 13, no. 2, pp. 260–269, April.

17. **Bahl, L., Cocke, J., Jelinek F., and Raviv, J., (1974),** *"Optimal decoding of linear codes for minimizing symbol error rate",* IEEE Trans. Inform. Theory, vol. 20, no. 2, pp. 284–287, March.

18. **Fano, R. M., (1963),** "*A heuristic discussion of probabilistic decoding",* IEEE Trans. Inform. Theory, vol. 9, no. 2, pp. 64–74, April.

19. **Gallager, R. G., (1963),** *"Low-Density Parity-Check Codes",* Cambridge, MA,USA: M.I.T. Press.

20. **Berrou, C., Glavieux, A., and Thitimajshima, P., (1993),** "*Near Shannon limit error-correcting coding and decoding,"* in Proc. of ICC, Geneve, Switzerland, pp. 1064–1070, May.

21. **MacKay, D. J. C. and Neal, R. M., (1995),** *"Good codes based on very sparse matrices",* in Cryptography and Coding. 5th IMA Conference, ser. Lecture Notes in Computer Science, C. Boyd, Ed. Berlin, no. 1025, pp. 100–111.

22. **Sipser, M. and Spielman, D. A., (1996),** *"Expander codes",* IEEE Trans. Inform.Theory, vol. 42, no. 6, pp. 1710–1722, November.

23. **Wiberg, N., Loeliger, H. A., and Kotter, R., (1995),** *"Codes and iterative decoding on general graphs",* European Transactions on Telecommunications, vol. 6, pp. 513–526, September.

24. **Wiberg N., (1996),** "*Codes and decoding on general graphs*", Ph.D. dissertation, Linkoping University, S-581 83, Linkoping, Sweden.

25. **Luby, M., Mitzenmacher, M., Shokrollahi, A., Spielman, D. A., and Stemann, V., (1997),** "*Practical loss-resilient codes",* in Proc. of the 29th annual ACM Symposium on Theory of Computing, pp. 150–159.

26. **Luby, M., Mitzenmacher, M., Shokrollahi, A. and Spielman, D. A., (1998),** "*Analysis of low density codes and improved designs using irregular graphs",* in Proc. of the 30th Annual ACM Symposium on Theory of Computing, pp. 249–258.

27. **Luby, M., Mitzenmacher, M., Shokrollahi, A. and Spielman, D. A., (2001),** *"Efficient erasure correcting codes*", IEEE Trans. Inform. Theory, vol. 47, no. 2, pp. 569–584, February.

28. **Luby, M., Mitzenmacher, M., Shokrollahi, A. and Spielman, D. A., (2001),** "*Improved low-density parity-check codes using irregular graphs*", IEEE Trans. Inform. Theory, vol. 47, no. 2, pp. 585–598, February.

29. **Luby, M. G., Mitzenmacher, M., and Shokrollahi, M. A., (1998),** *"Analysis of random processes via and-or tree evaluation",* in SODA '98: Proceedings of the ninth annual ACM-SIAM symposium on Discrete algorithms, San Francisco, California, United States, pp. 364–373.

30. **Richardson, T. and Urbanke, R., (2001),** "*The capacity of low-density parity check codes under message-passing decoding*", IEEE Trans. Inform. Theory, vol. 47, no. 2, pp. 599–618, February.

31. **Y. Chung, S. Y., G. D. Forney, G. D., Jr., Richardson T., and Urbanke R., (2001),** "*On the design of low-density parity-check codes within 0.0045 dB of the Shannon limit",* IEEE Communications Letters, vol. 5, no. 2, pp. 58–60, February.

32. **Arıkan, (2009),** *"Channel polarization: A method for constructing capacity achieving codes for symmetric binary- input memoryless channels*", IEEE Trans. Inform. Theory, vol. 55, no. 7, pp. 3051-3073, July.

33. **Arıkan, (2006),** *"Channel combining and splitting for cutoff rate improvement",* IEEE Transactions on Information Theory, vol. 52, pp. 628-639, February.

34. **Arikan, (2009)**," *Polarization for arbitrary discrete memoryless channels*", IEEE Information Theory workshop, pp. 144-148, August.

35. **Arıkan, (2008),** *"A performance comparison of polar codes and Reed-Muller codes"*, IEEE Communications Letters, vol. 12, No. 6, June.

36. **Kai, Chen, (2013),** *"Improved Successive Cancellation Decoding of Polar Code*", IEEE Transactions on Communications, vol. 61, No. 8, August.

37. **Bo, Yuan, (2015),** "*Algorithm and VLSI Architecture for Polar Codes Decoder"*, Ph.D. Dissertation, July.

38. **Alexios, Balatsoukas, (2015),***" LLR-Based Successive Cancellation List Decoding of Polar Codes"*, Submitted to IEEE Transactions on signal processing, March.

39. **Xiao, Liang, Chuan, Zhang, Menghui, Xu, Shunqing, Zhang and Xiaohu, You, (2015),** "*Efficient Stochastic List Successive Cancellation Decoder for Polar Codes",* IEEE International system-on-chip conference (SOCC), pp. 421-426.

40. **Orion, Afisiadis, Balatsoukas-Stimming, A. and Andrean, Burg, (2014)**, "*A Low-Complexity Improved Successive Cancellation Decoder for Polar Codes",* IEEE 48th Asilomar conference on signal, systems and computers, pp. 2116-2120.

41. **Chuan, Zhang, Zhongfeng, Wang, Xiaohu ,You and Bo, Yuan, (2014)***," Efficient Adaptive List Successive Cancellation Decoder for Polar Codes",* IEEE 48th Asilomar conference on signal, systems and computers, pp. 126-130.

42. **Hadi, Ammar, Alsusa, Emad and Rabie, Khaled M., (2016),** *"A Method to Enhance the Performance of Successive Cancellation Dcoding in Polar Codes*", IEEE 10th International Symposium on communication systems, Network and Digital Signal Processing (CSNDSP), pp. 1-5.

43. **Vangala, H., Viterbo, E. and Hong, Y., (2014),** *"Permuted successive cancellation decoderfor polar codes",* In Proc. Int. Symp. Inf. Theory and Its App. (ISITA), pp. 438– 442, Melbourne, VIC, October.

44. **Vangala, H., Viterbo, E. and Hong, Y., (2014),** *"Improved multiple folded successive cancellation decoder for polar codes"*, In General Assembly and Scientific Symposium, pp. 1–4, August.

45. **Balatsoukas-Stimming A., Parizi, M. and Burg, A., (2014),** *"LLR-based successive cancellationlist decoding of polar codes",* In Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP), pp. 3903 – 3907, May.

46. **Kahraman, S., Viterbo, E., and Celebi, M., (2014),** *"Multiple folding for successive cancellationdecoding of polar codes",* IEEE Commun. Lett., vol. 3, no. 5, pp. 545–548, October.

47. **Lin, B., Shen, H., Tse, D., and Tong, W., (2014),** *"Low-latency polar codes via hybriddecoding",* In Proc. 8-th Int. Symp. Turbo Codes & Iterative Information Process., pp. 223 – 227, Bremen, Germany, August.

48. **Cao, C., Fei, Z., Yuan, J., and Kuang, J., (2014),** "*Low complexity list successive cancellation decoding of polar codes",* IET Commun., vol. 8, no. 17, pp. 3145 – 3149, November.

49. **Kahraman, S., Viterbo, E., and Celebi, M., (2014),** "*Folded successive cancellation decoding of polar codes",* Theory Workshop, pp. 57–61, February.

50. **Tal, I. and Vardy, A., (2011),** "*List decoding of polar codes",* IEEE Int. Symp. Inform. Theory (ISIT), pp. 1-5.

51. **Niu, K. and Chen, K., (2012),** "*Stack decoding of polar codes",* Electronics Letters, vol. 48, no. 12, pp. 695-696.

52. **Niu, K. and Chen, K., (2012),** *"CRC-Aided Decoding of Polar Codes",* IEEE Communication Letters, vol. 16, no. 10, October.

53. **Dan, Le, Xianyan, Wu and Xiamu, Niu, (2016),** *"Decoding schedule generating method for successive-cancellation decoder of polar code*", IET Commun., Vol. 10, lss. 5, pp. 462-467.

54. **Camille Leroux, Alexandre, Raymond J., Gabi Sarkis, and Warren, J., Gross, (2013),***"A Semi-Parallel Successive-Cancellation Decoder for Polar Codes",* IEEE Transactions on Signal Processing, vol. 61, pp. 289-299.

55. **Arikan, (2009),**" *Polarization for arbitrary discrete memoryless channels",* IEEE Information Theory workshop, pp. 144-148, August.

56. **Ramtin Pedarsani, S. Hamed Hassani, Ido Tal, Emre Telatar, (2011),***" On the Construction of Polar Code*" IEEE International symposium, pp. 11-15.

57. **Navneet Kaur and Arvinder Pal Signgh Kalsi, (2016),**" *Implementation of codes over AWGN and Binary Symmetric Cannel",* Indian Journal of Science and Technology, Vol. 9, no. 19, March.

58. **Gabi Sarkis, Warren, J. Gross, (2013)**," *Increasing Throughput of Polar Decoder",* IEEE Communications Latters, vol. 17, pp. 725-728, April.

59. **Seyyed, Ali Hashemi, Carlo, Condo and Warren, J. Gross, (2016),** "*A Fast Polar Code List Decoder Architecture Based on Sphere Decoding",* IEEE Transactions on Circuits and Systems, vol. 63, pp. 2368-2380.

60. **Gabi Sarkis, Pascal Giard, Alexander Vardy, Claude Thibeault and Warren J. Gross, (2014),** "*Fast Polar Decoders: Algorithm and Implementation",* IEEE Journal on Selected Areas in Communications, vol. 32, pp. 940-957.

61. **Gabi, Sarkis, Pascal, Giard, Alexander, Vardy, Claude, Thibeault and Warren, J. Gross, (2014),"** *Increasing the speed of polar list decoders"* , IEEE Workshop on Signal Processing Systems (SiPS), pp. 1-6.

62. **Chandesris, L., Savin, V. and Declercq, D., (2016),** "*An Improved SCFlip Decoder for Polar Codes",* Global Communication Conference, December.

63. **Koopman, P. and Chakravarty, T., (2004),** *"Cyclic Redundancy Code (CRC) Polynomial Selection for Embedded Networks"*, The International Conference on Dependable Systems and Networks, DSN-2004.

64. **Gazi, O., (2019),** "*Polar Codes: A Non-Trivial Approach to Channel Coding*", Springer, ISBN: 978-981-13-0736-2.