POLAR CODE DECODING WITH SOFT DECISION ALGORITHMS

AHMET ÇAĞRI ARLI

JANUARY 2020

POLAR CODE DECODING WITH SOFT DECISION ALGORITHMS

A THESIS SUBMITTED TO

THE GRADUATE SCHOOL OF NATURAL AND APPLIED

SCIENCES OF

ÇANKAYA UNIVERSITY

BY

AHMET ÇAĞRI ARLI

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE

DEGREE OF

DOCTOR OF PHILOSOPHY

IN

THE DEPARTMENT OF

ELECTRONIC AND COMMUNICATION ENGINEERING

JANUARY 2020

Title of the Thesis: **Polar Code Decoding With Soft Decision Algorithms**

Submitted by **Ahmet Çağrı ARLI**

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.

_____

Prof. Dr. Sıtkı Kemal İDER
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 Doctor of Philosophy.

_____

Assoc. Prof. Dr. Orhan GAZİ
Supervisor

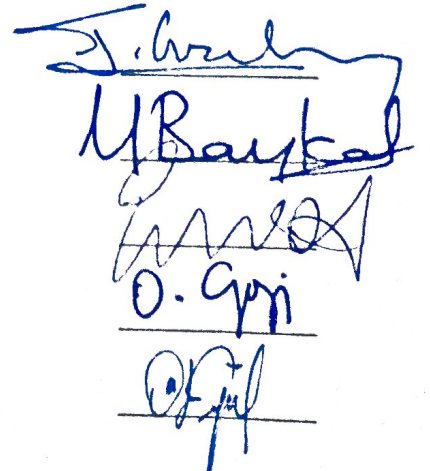**Examination Date: 31.01.2020**

Examining Committee Members

Prof. Dr. Erdal ARIKAN	(Bilkent University)	
Prof. Dr. Yahya Kemal BAYKAL	(Çankaya University)	
Prof. Dr. Emre AKTAŞ	(Hacettpe University)	
Assoc. Prof. Dr. Orhan GAZİ	(Çankaya University)	
Dr. Lecturer Özgür ERGÜL	(Atılım University)	

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 : Ahmet Çağrı ARLI

Signature :

Date : 31.01.2020

ABSTRACT

POLAR CODE DECODING WITH SOFT DECISION ALGORITHMS

ARLI, Ahmet Çağrı

Ph.D., Department of Electronic and Communication Engineering

Supervisor: Assoc. Prof. Dr. Orhan GAZİ

January 2020, 129 pages

Since the born of the field of information theory with the publication of Shannon's famous paper, a mathematical theory of communication, numerous channel codes have been developed to achieve the performance limits drawn by Shannon. Initially, the channel codes are constructed using the binary vector subspaces, i.e., block codes, and the performances of these codes are measured via computer simulations. The codes that show good simulation results are adapted in practical communication systems. A different class of channel codes, convolutional codes are discovered in 1955 by Elias. Convolutional codes show basic differences in encoding and decoding operations than block codes. In 1993, turbo codes, parallel concatenated convolutional codes, are introduced. The astonishing performance of turbo codes is considered as a milestone in channel coding society, and a huge interest for the design of concatenated codes aroused among researchers. The designed codes are decoded in an iterative manner, which was one of the main reasons behind the superior performance of turbo codes. The common idea in the years of 2000 among the researchers was that iteratively decodable concatenated codes was the codes not to be overcame for long years. In 2009 polar codes are introduced by Erdal Arıkan. Polar codes are designed using the concepts of information theory, and their performances are proven mathematically. Polar codes can be considered as the only codes

designed in a non-trivial manner, which can be considered as a breakthrough in channel coding field.

In this doctoral thesis, the decoding of polar codes with soft decision based algorithms is studied. The belief propagation algorithm, which is one of the soft decision based algorithms, has been investigated in details literature. Polar codes can be decoded using belief propagation algorithm. For communication systems decoding latency is a critical issue. Decoding latency can be reduced using parallel processors. Belief propagation algorithm is suitable for parallel processing operations. It is indicated in the literature that polar codes decoded by belief propagation algorithm show worse performance than the polar codes decoded by successive cancelation decoding algorithm. In this thesis, we aim to improve the performance of polar codes decoded by belief propagation algorithm so that once it is achieved, the suitability for parallel processing property of the belief propagation algorithm can gain significance.

The propagation of the unreliable probabilities in belief propagation algorithm worsens the performance of polar codes. To improve the reliability of propagating messages, we made use of the artificially generated weak noise signals. It is seen from the simulation results that the addition of weak noise to the received signal enhances the performance of polar codes decoded by belief propagation algorithm. The proposed approach can be named as noise-aided belief propagation based list, i.e., Na-BPL, polar decoder. With the proposed method, it is seen that the performance of polar codes with belief propagation decoders employing perfect knowledge based early termination approaches to the performance of state-of-the-art successive cancelation list polar decoders. The systematic versions of polar codes are also considered with belief propagation algorithm. The systematic encoding brings extra overheads to the successive cancellation polar decoder. On the other hand, when it is used with belief propagation algorithm, the extra overhead is not seen at the decoder part. In this way, we further improve the performance of the polar belief propagation decoders. Besides, when systematic polar codes are used with Na-BPL, better decoding performance is obtained.

Accurate calculation of splitted channel capacities is a critical issue affecting the performance of polar codes. We considered the genetic algorithm for the design of polar codes. It is indicated in the thesis that when the polar codes designed with genetic algorithm is decoded using Na-BPL, improved performance is achieved, and the obtained performance is only 0.1 dB away from the performance of state-of-the-art polar decoder, i.e., CRC aided SCL polar decoder, when perfect knowledge based early termination is employed. And it is seen that without perfect knowledge based early termination, Na-BPL cannot overcome CRC aided SCL polar decoder, however, its error correction performance is better than SCL.

Keywords: Polar codes, belief propagation algorithm, error correction, list decoding, noise aid, soft decoding, stochastic perturbation.

ÖZ

KUTUP KODLARININ YUMUŞAK KARAR TABANLI ALGORITMALARLA ÇÖZÜMLENMESİ

ARLI, Ahmet Çağrı

Doktora, Elektronik ve Haberleşme Mühendisliği

Tez Yöneticisi: Doç. Dr. Orhan GAZİ

Ocak 2020, 129 Sayfa

Shannon'un haberleşmenin matematiksel teorisini anlattığı ünlü makalesi ile birlikte doğan bilgi teorisi kapsamında Shannon tarafından çizilen performans sınırlarına ulaşmak için çok sayıda kanal kodu geliştirilmiştir. Başlangıçta, kanal kodları ikili vektör alt uzayları, yani blok kodları kullanılarak oluşturulur ve bu kodların performansları bilgisayar simülasyonları ile ölçülmştür. İyi simülasyon sonuçlarına sahip kodlar pratik iletişim sistemlerine uyarlanmıştır.Farklı bir kanal kodu sınıfı olarak, evrişim kodları 1955 yılında Elias tarafından keşfedilmiştir. Evrişimli kodlar, kodlama ve kod çözme işlemlerinde blok kodlara göre temel farklılıklar gösterir. 1993'te turbo kodlar, paralel sıralı evrişim kodları olarak tanıtıldı. Turbo kodların şaşırtıcı performansı kanal kodlama toplumunda bir kilometre taşı olmuştur ve araştırmacılar arasında sıralı kodların tasarımına büyük ilgi duymuştur. Tasarlanan kodlar, turbo kodların üstün performansının arkasındaki ana nedenlerden biri olan yinelemeli bir şekilde çözülür.Araştırmacılar arasında 2000 yıllarındaki ortak fikir, yinelenebilir şekilde deşifre edilmiş birleştirilmiş kodların uzun yıllar üstesinden gelinmeyecek kodlar olmasıydı. 2009 yılında Erdal Arıkan tarafından kutup kodları tanıtıldı. Kutupsal kodlar bilgi teorisi kavramları kullanılarak tasarlanmıştır ve performansları matematiksel olarak kanıtlanmıştır. Kutupsal kodlar,

önemsiz olmayan bir şekilde tasarlanan tek kanallar olarak kabul edilebilir ve kanal kodlama alanında bir atılım olarak düşünülebilir.

Bu doktora tezinde, kutupsal kodların yumuşak karar tabanlı algoritmalarla çözümlenmesi incelenmiştir. Yumuşak karar temelli algoritmalardan biri olan karar yayılım algoritması ayrıntılı olarak literatürde incelenmiştir. Kutup kodları, karar yayılma algoritması kullanılarak çözülebilir. İletişim sistemleri için kod çözme gecikmesi kritik bir konudur. Kod çözme gecikmesi paralel işlemciler kullanılarak azaltılabilir. Bu bağlamda, karar yayılım algoritması paralel işleme operasyonları için uygundur. Literatürde, karar yayılma algoritması ile kodu çözülen kutup kodlarının, ardışık giderim algoritması tarafından kodu çözülen kutup kodlarından daha kötü performans gösterdiği belirtilmektedir. Bu tezde, karar yayılma algoritması tarafından çözülen kutup kodlarının performansını artırmayı hedefledik. Performans iyileştirmesi elde edildiği takdirde, inanç yayılma algoritmasının paralel işleme özelliğine uygunluğu öne çıkacaktır.

Karar yayılma algoritmasında güvenilmez olasılıkların yayılması kutupsal kodların performansını kötüleştirir. İletilerin güvenilirliğini arttırmak için yapay olarak üretilen zayıf gürültü sinyallerini kullandık. Simülasyon sonuçlarından, alıcıya gelen sinyale zayıf gürültü eklenmesinin, karar yayılma algoritması tarafından çözülen kutup kodlarının performansını arttırdığı görülmektedir. Önerilen yaklaşım, gürültü destekli karar yayılımına dayalı liste, yani Na-BPL, kutupsal kod çözücü olarak adlandırılabilir. Önerilen yaklaşımla, karar yayılımı kod çözücüleri ile kutupsal kod performansının, en gelişmiş ardışık giderim liste kutup çözücüleri performansına yaklaştığı görülmektedir. Kutupsal kodların sistematik versiyonları da karar yayılma algoritmasına uygulanmıştır. Sistematik kodlama, ardışık giderim kutupsal kod çözücüsüne fazladan ek yükler getirir. Öte yandan, karar yayılma algoritması ile kullanıldığında, kod çözücü kısmında fazladan ek yük görülmez. Bu şekilde, kutupsal karar yayılım kod çözücülerinin performansını daha da geliştiriyoruz. Ayrıca, Na-BPL ile sistematik polar kodlar kullanıldığında, daha iyi kod çözme performansı elde edilir.

Bölünmüş kanal kapasitelerinin doğru hesaplanması, kutupsal kod performansını etkileyen kritik bir konudur. Kutupsal kodlarının tasarımı için genetik algoritmayı da dikkate aldık. Tezde, genetik algoritma ile tasarlanan kutupsal kodların, mükemmel bilgi tabanlı erken tespit yöntemi kullanan Na-BPL ile çözüldüğünde, gelişmiş performans elde

edildiği ve elde edilen performansın, en son teknolojiye sahip kutupsal kod çözücünün, yani CRC ile desteklenmiş SCL kutupsal kod çözücü performansından sadece 0.1dB uzakta olduğu belirtilmiştir. Na-BPL kod çözücü mükemmel bilgi tabanlı erken tespit yöntemi kullanılmadığında CRC ile desteklenmiş SCL kod çözücüyle yarışamıyor fakat hata düzeltme performansı SCL kod çözücünün performansına kıyasla ileridedir.

Anahtar Kelimeler: Kutup kodları, karar yayılım algoritması, hata düzeltme, liste çözümlemesi, gürültü yardımı, yumuşak tabanlı çözümleme, stokastik karışıklık.

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.

This PhD study you will read is one of the strongest concrete indicators of my education. Thanks to my family, especially my mother who brought me into the world, I eagerly continued my education. Especially, I always felt that I had to go to the end for my mother Devrim Gül who left her career and spent best of her years to care my brother and me. I am very happy with the way my grandfather Ahmet Kaan, my grandmother Özgül Kaan and my father Fahrettin Arlı showed, since I was in a family of educators. Also, I want to thank my brother Anıl for his support. Finally, I offer my endless gratitude to my wife, Görkem, who has been my companion for 6 years, for the support she has given me to be able to do my work. As a result, I consecrate my doctorate to my family and country to which I have dedicated my existence.

Lastly, due to his contributions to my thesis work, I would like to thank Prof. Dr. Erdal Arıkan.

TABLE OF CONTENTS

FIGURES

LIST OF TABLES

TABLES

LIST OF ABBREVIATIONS

ABA   Average Based Assumption

ASIC   Application Specific Integrated Circuit

AWGN   Additive White Gaussian Noise

BCH   Bose–Chaudhuri–Hocquenghem

B-DMC   Binary-Discrete Memoryless Channel

BEC   Binary Erasure Channel

BER   Bit Error Rate

BLER   Block Error Rate

BMS   Binary input Memoryless Symmetric

BP   Belief Propagation

BPL   Belief Propagation List

BPSK   Binary Phase Shift Keying

BSC   Binary Symmetric Channel

CMOS   Complementary Metal Oxide Semiconductor

CNT   Carbon Nanotube

CORR   CORRelation based Decision

CRC   Cyclic Redundancy Check

CTC   Convolutional Turbo Code

DE   Density Evolution

DMC   Discrete Memoryless Channel

EEG   Electroencephalogram

eMBB	Enhanced Mobile Broadband
EMG	Electromyography
EXIT	Extrinsic Information Transfer Chart
FEC	Forward Error Correction
FET	Field-Effect Transistor
FG	Factor Graph
5G	Fifth Generation wireless technology
FPGA	Field Programmable Gate Array
GA	Gaussian Approximation
GenAlg	Genetic Algorithm
3GPP	3rd Generation Partnership Project
GPU	Graphical Processing Unit
IEEE	Institute of Electrical and Electronics Engineers
LAN	Local-Area Network
LCD	Leader of Converged Decoders
LDPC	Low Density Parity Check
LMA	LLR-Magnitude aided
LR	Likelihood Ratio
LLR	Log-Likelihood Ratio
MC	Monte Carlo
ML	Maximum Likelihood
mMTC	Massive Machine-Type Communications
MS	Min-Sum
Na-BPL	Noise-aided Belief Propagation List
NND	Neural Network Decoder
OCI	Observation of Consecutive Iterations
PKB	Perfect Knowledge Based
RM	Reed-Muller

RS	Reed-Solomon
RT	Round Trip
SC	Successive Cancellation
SCAN	Soft Successive Cancellation
SCL	Successive Cancellation List
SCS	Successive Cancellation Stack
SMS	Scaled Min-Sum
SNR	Signal-to-Noise Ratio
SR	Stochastic Resonance
3GPP	3rd Generation Partnership Project
TSMC	Taiwan Semiconductor Manufacturing Company
uRLLC	Ultra-Reliable Low Latency Communications
WAN	Wide Area Network
WIB	Worst of Information Bits
XOR	Exclusive OR

CHAPTER 1

INTRODUCTION

1.1 Background

In 1948, C.E.Shannon showed that error-free transmission is possible with any code rate up to channel capacity [1]. Since then, numerous forward error correction (FEC) techniques have been developed to achieve the channel capacity. Hamming, Golay, Bose-Chaudhuri-Hockenham (BCH), Reed-Solomon (RS), Reed-Muller (RM), low density parity check (LDPC), turbo and polar codes are the error correcting codes invented in the past 70 years. FEC methods are mainly used in wireless communication systems to improve the service quality. Moreover, various FEC methods are utilized for the frameworks from 2G to 5G. All channel codes up to polar codes are designed in a trivial manner. In 2009, Arıkan introduced polar codes, the first channel codes whose performance is mathematically proved [2].

Arıkan designed the polar codes in such a way that the probability of error for the transmitted bit can be calculated beforehand, thus considering the probability value, the decision is made to transmit the data bit or not. The bit transmission probabilities are closely related to the channel capacities between the bit to be transmitted and the received symbols and the previously decoded bits. These channel capacities are also named as splitted channel capacities. Splitted channel capacities are used to decide the location of parity bits called frozen bits for polar codes such that both the transmitter and the receiver knows the locations of frozen bits.

Frozen bits are added to the data vector before the encoding operation. In all channel codes except polar codes, parity bits are generated during/after encoding. On the other hand, in

polar codes, parity bits are placed into data vector by taking into account the capacity of the divided channel before encoding. Channels with low capacities are used for parity bits, while high capacity channels are used to bear data bits, and this smart channel usage makes the polar codes perform better than turbo and LDPC codes. In his article [2], Arıkan also introduced the successive cancellation decoding of polar codes. After its introduction, polar codes have received great interest and have become an important topic of future communication standards. In addition, polar codes are selected to provide channel coding on the control channels of the new radio (NR) of the third generation partnership project (3GPP) [3]. It has been shown that polar codes can be used for source coding [4, 5, 6] as well as channel coding.

Different decoding schemes are developed for polar codes in time after its release. There are four state-of-the-art polar decoding methods: successive cancellation (SC) [2], belief propagation (BP) [7], linear programming (LP) [8], and maximum likelihood (ML) [9]. In ML decoding, likelihood of all possible codewords are calculated. Then, the most probable codeword is determined. Since, ML decoding is based on searching all possible codewords, it is not applicable after a certain codeword length N, since in total there are 2^N possible codewords are available, and for large N values it becomes impossible to handle the amount of calculations needed. SC decoding of polar codes is offered for its practical applicability in [2]. With its low error correction performance for short code lengths, SC algorithm offers lowest computational complexity. Performance of SC decoder is enhanced greatly with the introduction of SC list [10] decoder keeping the complexity low. As a general form of SC decoder and its soft decision based version, BP-based polar decoder is proposed in [7]. BP decoder processes soft information in an iterative manner. Moreover, instead of performing sequential decoding like SC based decoders, BP decoder utilizes parallel decoding. Besides, parallel decoding can be implemented on electronic devices such as Field Programmable Gate Array (FPGA) and Graphical Processor Unit (GPU) easily. Consequently, despite its low performance on error correction with its soft decision based decoding process and parallel decoding capability, BP-based polar decoder is a promising candidate for communication standards. Decoding FEC codes with soft decision based decoders is widely used in communication systems to lower block/frame and bit error rate when compared hard decision based

decoders. Having the soft decision based decoders, LDPC and turbo codes are employed in 4G communication standards. Besides, LDPC and polar codes are going to be used in 5G frameworks.

Belief propagation is a message-passing algorithm based on graphical models such as Bayesian network and Markov random models. It is also widely used in coding theory with iterative approach as a soft decision based decoding algorithm. Recursive messaging causes decoders to converge, such that one can estimate the data bits correctly.

Turbo decoder processes soft information in an iterative manner and this is the main reason behind the astonishing performance of turbo codes. Like turbo and LDPC decoders, BP-based decoding algorithms are based on an iterative message-passing structure and parallel decoding capabilities that provide fast convergence of the decoder. As an example, Forney has shown that RM codes can be represented graphically and can be decoded by using BP algorithms [11].

Motivating from the fact that a Reed-Muller code, RM(N, K), is a polar code of length 2^N with code rate K / N and it differs only in the choice of generator vector [4], hence, polar codes can be decoded using BP algorithm in an iterative manner as RM codes. Arıkan presented a BP polar decoder [7] that simply identifies the check nodes and variable nodes on Tanner graph of polar code and related message propagation equations.

The performance of polar codes for ML and BP decoders is compared with RM codes for short lengths in [9]. Thanks to its larger minimum distance over polar code, RM codes yield a better error correction at high signal-to-noise ratio (SNR), while, polar codes outperforms RM codes at low SNR. For all code-word lengths, ML has the best error correction performance beating BP decoder. Since ML decoding follows a process searching all possible code-words, its large complexity will be a problem to implement for $N > 64$. With its acceptable computational complexity BP decoding of RM and polar codes can be considered. It is also important to state that minimum distance advantage of RM code over polar code tends to disappear as N gets larger values.

Polar codes with length $N \leq 2^{14}$ do not show substantial error correction performance when compared to LDPC and turbo codes [12] if polar decoder type is selected as SC or BP. However, 1 dB gain over LDPC and turbo codes for code lengths $N < 1000$ is achieved when state-of-the-art cyclic redundancy check (CRC) aided SC List decoder introduced

in [10] is used. The excellent BER/BLER performance of polar codes with short lengths, i.e., $N < 1000$, makes them a good candidate to be utilized for the control plane of 5G frameworks [3].

1.2 Motivation

The 3GPP decided that LDPC codes are more suitable for the physical data channels of 5G enhanced mobile broadband (eMBB) communication service with its high throughput while polar codes are considered for uplink/downlink physical control channel with its low computational complexity [3]. Design of LDPC codes for 5G New Radio (NR) are presented in [13]. Polar codes are also one of the candidates of other 5G frameworks, ultra-reliable low latency communications (URLLC) [14] and massive machine-type communications (mMTC). Framework requirements change relatively with respect to the coverage area, data rate, energy saving and cost needs. It can be anticipated that the fate of polar codes will be determined in URLLC and mMTC frameworks when the next releases of 3GPP are announced. It is shown in [13, 15] that polar codes are appropriate choice for URLLC and mMTC scenarios.

CRC aided SCL polar decoder is planned to be used in 5G eMBB frameworks, and polar decoding with BP-based decoder is a good candidate for low latency applications. One of the most important advantage of BP decoding over SC / SCL decoding is its suitability for parallel processing. Unlike SC / SCL, previously estimated bits are not used to decode the next bits. Even more, all bits are resolved simultaneously after a series of consecutive iterations. However, parallel processing increases the complexity in terms of the number of logical elements and memory requirements. The memory requirement comes from the usage of messages from the previous iteration of the node during current iteration.

The aim of this thesis is to offer a BP-based polar decoding scheme in order to compete CRC-aided SCL polar decoder. We assume that if we can improve the performance of BP polar decoder such that it achieves the same performance with CRC-aided SCL decoder, then it can be a candidate for future communication standards due to its suitability for parallel processing and due to its flexibility for integration with other communication units having soft information processing property.

1.3 Outline

The thesis consists of five chapters. Chapter 2 focuses on explanations of polar codes, polar encoder and polar decoder types. Furthermore, details of BP polar decoder are provided. In the chapter 3, a wide range literature survey that combines and classifies journal papers, conference papers, books, master and PhD theses and patents is provided. Approximately one hundred studies are examined carefully and these studies on BP polar decoder are presented with a reasonable classification. Classification is done to separate studies that focuses on performance improvement, complexity reduction and increasing decoding speed. Besides, every aspect of a BP polar decoder is argued and references of each subject are provided to lead readers better understanding. As the main part of the thesis, in chapter 4 noise-aided belief propagation list decoder is proposed. Before the proposition of the method, step-by-step construction of the decoder is explained. Proposed design is supported with comparative simulation results. Error correction of proposed decoder is also enhanced by introducing the systematic version of the decoder without adding complexity. Additional performance boost to our proposed decoder is provided by applying a new polar code construction that is based on genetic algorithm. Finally, thesis will end with the conclusion part.

CHAPTER 2

POLAR CODES

In communication theory, there are two different types of channel as continuous and discrete channels. A continuous communication channel can be regarded as an analogue channel and has two inputs and one output. Input signal $s(t)$ and channel noise $n(t)$ are inputs whereas $r(t)$ is the output of the continuous channel, Fig. 2.1.
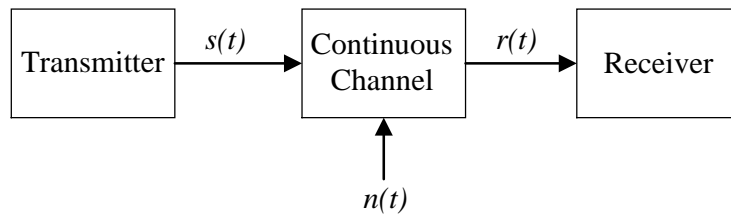
Figure 2.1 A continuous communication channel

On the other hand, a discrete channel has a discrete input and a discrete output where a probability is defined for each bits/symbols to appear at the input and output of the discrete channel. Input alphabet is defined with X, while output alphabet is defined as Y. As seen from the Fig. 2.2, relation between X and Y is defined with transition probabilities; denoted as $Pr(y_n|x_n)$.
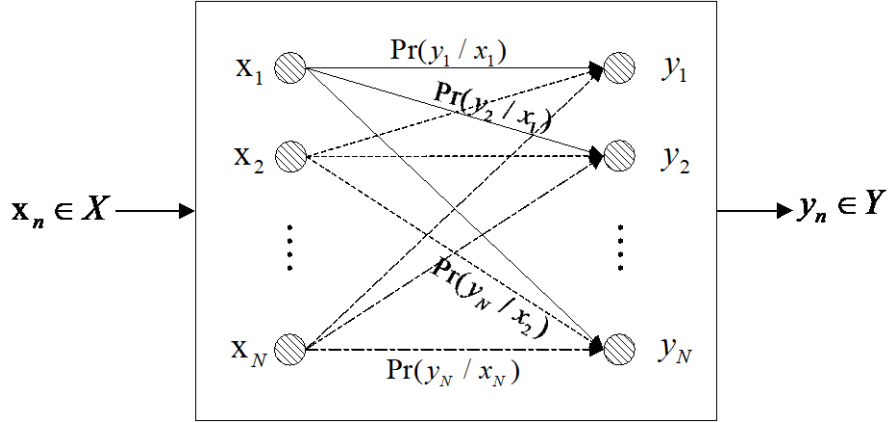
Figure 2.2 A sample discrete channel

Now, we can introduce the philosophy of polar codes and its counterparts in a step-by-step approach. Polar codes are defined on binary-discrete memoryless channels [2] for the first time. When a discrete memoryless channel, denoted as *W*, taken into consideration, input and output of the channel can be represented with discrete random variables, **u** and **y**, respectively. A discrete channel as shown in Fig. 2.2, can be separated into smaller channels like demonstrated in Fig. 2.4a. For instance, one discrete memoryless channel (DMC) is needed to send one symbol, while four separate DMCs can be utilized to handle four symbols.

2.1 Channel Polarization

Claude E. Shannon [1] provided a mathematical model that computes channel capacity. Capacity of a channel can be defined as the maximum mutual information between the input and output of any channel. In our case, each separate DMCs has its own capacity that changes between 0 and 1. Polar coding can be formed by combining these separate discrete channels such that capacities are transformed to examine channels. In this scope, channel polarization allows us to achieve extreme channels by transforming non-zero capacity channels. Extreme channels in other words polarized channels are the channels with zero capacity or the channels with capacity i.e. $C(W) = 1$ where $0 \leq C(W) \leq 1$. Transformation process is called as channel combining. DMC channels that form bigger channels without and with transformation process are demonstrated in Fig. 2.3a and Fig. 2.3b, respectively. **G** is the transformation matrix that will be introduced in the next sub-section as generator matrix of the polar code.
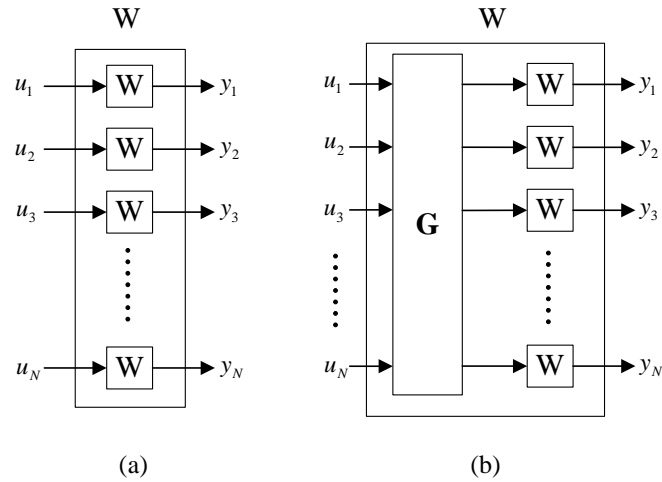
Figure 2.3 (a) Channel combining, (b) channel combining with channel polarization

Channel polarization is provided by a simple operation [2]. Operation includes a modulo-2 addition that can be accomplished by using an exclusive-OR (XOR) gate. Fig. 2.4 shows two channel combining examples. Two binary-DMCs (B-DMC) are combined to create W_2 as shown in Fig 2.4a where two W_2 channels are combined to form W_4 in Fig. 2.4b. It is obvious that for codeword lengths, power of two, encoding can be achieved in a recursive manner.
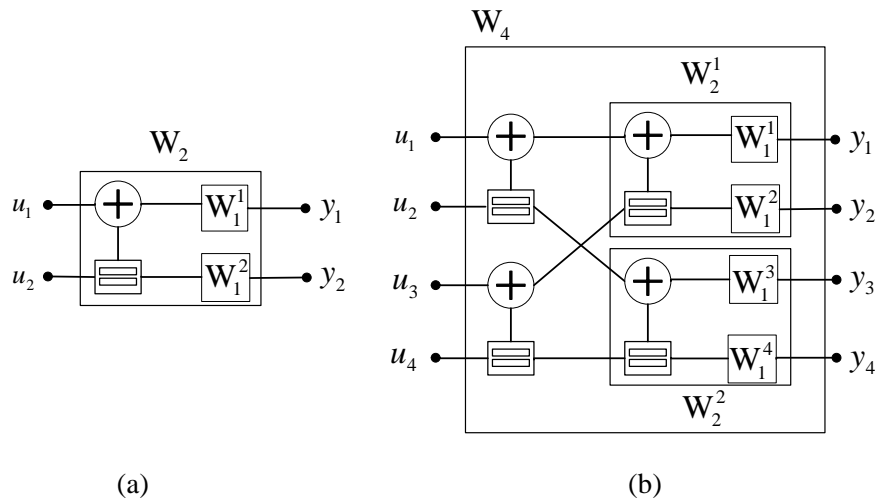
Figure 2.4 Channel polarization examples (a) forming W_2 , (b) forming W_4

Subsequently, channel capacities are polarized such that they close to 0 or 1. After introducing the polarization concept, we will explain how a polar code can be constructed for a Binary Erasure Channel (BEC).

Channel splitting should also be introduced to show polarized channels have better or worse channel capacity than the original B-DMC *W*. Hence, we can split the combined channel into two sub channels and define the bad and good B-DMCs as

$$W^+\left(y_1, y_2 \,\middle|\, u_1\right) = \frac{1}{2} \sum_{u_2 \in \{0,1\}} W_2\left(y_1 \,\middle|\, u_1 \oplus u_2\right) W_2\left(y_1 \,\middle|\, y_2\right) \qquad (2.1)$$

$$W^-\left(y_1, y_2, u_1 \,\middle|\, u_2\right) = \frac{1}{2} W_2\left(y_1 \,\middle|\, u_1 \oplus u_2\right) W_2\left(y_2 \,\middle|\, y_2\right). \qquad (2.2)$$

After some operation, channel capacity comparison between before and the after polar transform can be presented as,

$$C(W^-) \le C(W) \le C(W^+). \qquad (2.3)$$

2.2 Polar Code Construction

Polar code construction is dependent on the type of the communication channel and decoder. A polar code construction method should focus on to polarize *N* independent copies of a given B-DMC channels to 0 or 1 such that synthetic i.e. polarized channels are formed. To measure success of polarization, quality of channel should be examined. Quality of synthetic channels can be evaluated by Bhattacharyya parameter, where it is donated as *Z(W)*. As presented by Arıkan [2], we will demonstrate the Bhattacharyya parameter based polar code construction for BEC. It is important to state that after polar code construction, information and frozen bit places are decided. Usually, frozen bit places are zero/close to zero capacity channels that are known by receiver. Thus, Bhattacharyya parameter can be defined as

$$Z(W) \triangleq \sum_{y \in Y} \sqrt{w(y \,|\, 0) w(y \,|\, 1)} \qquad (2.4)$$

where $y \in Y$ is output alphabet. Polar code design can be done with recursive instructions as shown in Fig. 2.4. Mathematically, Bhattacharyya parameter can be calculated as

$$Z(W_{2N}^{(i-1)}) = 2Z(W_N^{(i)}) - Z(W_N^{(i)})^2 \qquad (2.5)$$

$$Z(W_{2N}^{(2i)}) = Z(W_N^{(i)})^2 \qquad (2.6)$$

As expected, we should start recursive calculation by deciding on the value of W_1^i. W_1^i can be thought as initial step of the polar code construction. That is why, deciding on W_1^i is vital to achieve successful polarization. Since we are dealing with BEC, we can set W_1^i as erasure rate of the channel. As an example, we choose to use erasure rate α as 0.5 for code length $N = 8$. In order to lead better understanding, Bhattacharyya parameter calculation on a polar code is demonstrated in Fig. 2.5.
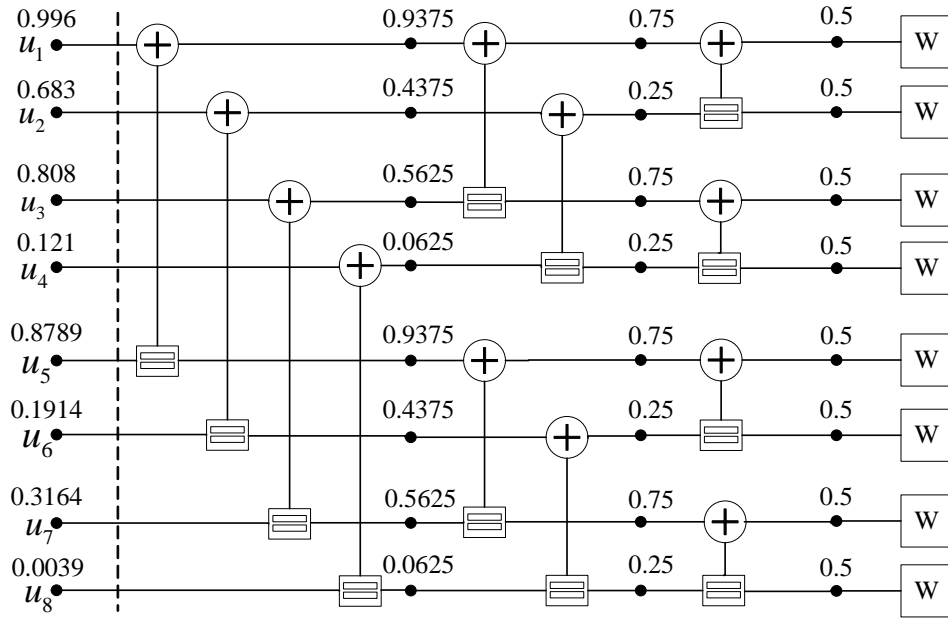
Figure 2.5 Bhattacharyya parameter based polar code construction for $N = 8$

Fig. 2.5. shows that Bhattacharyya parameter $Z(W)$ is high for u_1, u_2, u_3 and u_5 while rest has lower values. Knowing $Z(W)+C(W)=1$, we can conclude that after polarization, u_1, u_2, u_3 and u_5 have lowest capacities and they should be selected as frozen bits. Overall, data sequence can be represented as $\mathbf{u}=[\,f\,f\,f\,d\,f\,d\,d\,d\,]$ where f and d stand for frozen bit and data/information bit. Frozen bit values are usually set as 0. After determining frozen and information bit places for an exact communication channel and decoder, our data sequence $\mathbf{u}$ is ready for encoding operation.

2.3. Polar Encoder

Polar encoder kernel unit is depicted in Fig. 2.6.
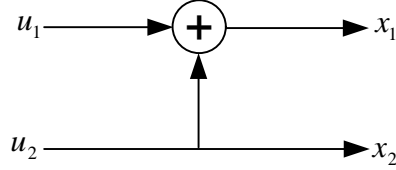
Figure 2.6 Polar encoder for $N = 2$

From Fig. 2.6, we can write that

$$x_1 = u_1 \oplus u_2 \quad x_2 = u_2$$

For the information word $\mathbf{u} = [u_1 u_2]$, after polar encoding operation, the obtained codeword is $\mathbf{x} = [x_1 x_2]$ where $x_1 = u_1 \oplus u_2$ and $x_2 = u_2$. For any encoder, the relation between $\mathbf{u}$ and $\mathbf{x}$ mathematically be expressed as

$$\mathbf{x} = \mathbf{u} \mathbf{G}_N \tag{2.7}$$

where $\mathbf{G}_N$ is the generator matrix. For $N = 2$, $\mathbf{G}_N$ is equal to

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

$$\swarrow \quad \searrow \qquad .$$

$$x_1 = u_1 \oplus u_2 \quad x_2 = u_2$$

The generator matrix of the polar code is an involutory such that

$$\mathbf{G}_N = \mathbf{G}_N^{-1}. \tag{2.8}$$

This property implies that

$$\mathbf{x} = \mathbf{u} \mathbf{G}_N \rightarrow \mathbf{u} = \mathbf{x} \mathbf{G}_N^{-1} \rightarrow \mathbf{u} = \mathbf{x} \mathbf{G}_N.$$

$\mathbf{G_N}$ can be formed in a recursive manner, starting from the mathematical equivalent of kernel polar encoder unit presented in Fig. 2.6. As the kernel matrix $\mathbf{F}$ is defined as $\mathbf{F}=\begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$. Bigger polar code are achieved by $\mathbf{G_N}=\mathbf{B_N}\mathbf{F}^{\otimes n}$ where $\mathbf{B_N}$ is the permutation matrix also known as bit-reversal matrix. "$\otimes n$" denotes Kronecker product of matrices and "$\otimes n$" denotes the nth Kronecker power of a matrix. $\mathbf{B_N}$ can be calculated by using $\mathbf{B_N}=\mathbf{R_N}(\mathbf{I_2}\otimes\mathbf{B_{N/2}})$ where $\mathbf{I_2}$ is the identity matrix, and $\mathbf{R_N}$ is the permutation operation, such that $\mathbf{R_4}$ maps the input $\{1,2,3,4\}$ to $\{1,3,2,4\}$.

In this scope of coding theory, parity bits are usually added to the codeword after the multiplication of $\mathbf{G}$ matrix where size of the $\mathbf{G}$ matrix is $K\times N$ and $\mathbf{x}=\mathbf{uG}$. N is the code length and K is the number of useful bits in the code. Parity check matrix, $\mathbf{H}$, is also defined for any code. $\mathbf{H}$ is used to check whether decoding is successful or not by using the equality $\mathbf{xH}=0$ where $\mathbf{H}$ has size of $(N-K)\times N$. However, in polar coding, $\mathbf{H}$ and $\mathbf{G}$ matrices are combined into one matrix with size $N\times N$. Moreover, parity bits are added to the data vector before encoding operation is completed. In this thesis, polar code with codeword length N and K information bits is denoted as $\mathcal{P}(N,K)$.

2.4. Belief Propagation Based Decoding of Polar Codes

Using BP algorithm in forward error correction is proposed by R.G.Gallager [16] for the first time in 1962. Besides, Tanner graph representation of this algorithm is offered by R.M.Tanner [17] in 1981. BP decoding scheme is iterative and based on message passing between check nodes and variable nodes placed on the right and left hand side of factor graph. For $\mathbf{x}=\mathbf{uG_N}$ encoding operation, the kernel encoder and decoder units are shown in Fig. 2.7.
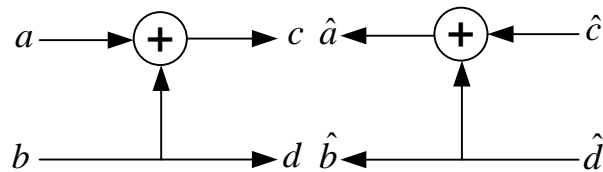
Figure 2.7 Polar encoder and decoder units for $N=2$

Let a,b,c,d be the random variables and assume that the bits c and d are transmitted through a discrete memoryless channel. At the decoder side, the flow of the signals change direction as shown in Fig. 2.7 where $\hat{c}$ and $\hat{d}$ are the outputs of a discrete memoryless channel, i.e., they are the received bits while $\hat{a}$ and $\hat{b}$ are the estimated data bits. Since $\mathbf{G}_N$ equals $\mathbf{G}_N^{-1}$, the encoding and decoding operations can also be interpreted as in Fig. 2.8.
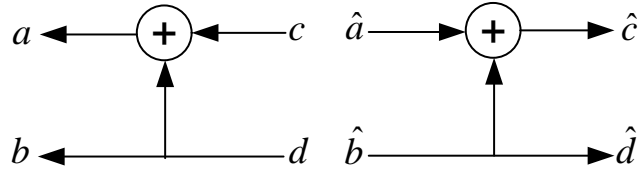
Figure 2.8 Alternative demonstration of polar encoder and decoder for $N=2$

Additionally, as we know that an iterative structure is going to be used with BP decoding algorithm we can combine the Fig. 2.7 and Fig. 2.8 as shown in Fig. 2.9. By this way, factor graph representation of kernel polar encoder/decoder is constructed to be able to allow from both left and right direction propagation of information.
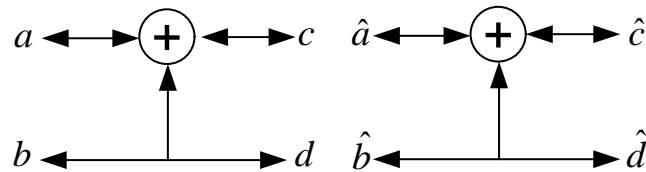
Figure 2.9 Factor graph representation of $\mathbf{G}_2$

Or using parallel arrows, factor graph will be modified as shown in Fig. 2.10a for better understanding. L stands for left propagation while R stands for right propagation.

(a)   (b)
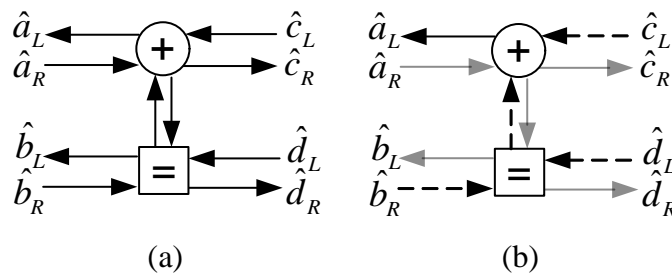
Figure 2.10 (a) Signal flow diagram for BP polar decoder, (b) messages that are used to calculate $\hat{a}_L$

At this point, signal flow on kernel BP polar decoder can be used to derive right and left propagation message equations. First step will be to define XOR and equality check operations in terms of probability functions. Probability functions are going to help us to form likelihood calculation equations. Let x,y,z,w be the random variables, where $p_x(x_1)=Prob(x=x_1)$ is the probability mass function. The XOR operation equals, $z=x \oplus y$, then

$$p_z(0) = p_x(0)p_y(0) + p_x(1)p_y(1)$$

$$p_z(1) = p_x(0)p_y(1) + p_x(1)p_y(0).$$

On the other hand, the equality check box can be defined with operator $\odot$. If $w=x \odot y$, then

$$w = x \odot y = \begin{cases} 0, & if\ x=y=0 \\ 1, & if\ x=y=1 \end{cases}.$$

From that, we can write

$$p_w(0) = p_x(0)p_y(0)$$

$$p_w(1) = p_x(1)p_y(1).$$

Upon these descriptions of the operations, we can try to calculate likelihood ratio (LR) of a_L by using

$$LR(a_L) = \frac{p(a_L = 0)}{p(a_L = 1)}.$$

By using Fig. 2.10b, we can write that

$$a_L = c_L \oplus [b_R \odot d_L].$$

From which, we can write the probabilities for a_L as

$$p(a_L = 0) = p(c_L = 0)p([b_R \odot d_L] = 0) + p(c_L = 1)p([b_R \odot d_L] = 1)$$

$$p(a_L = 0) = p(c_L = 0)p(b_R = 0)p(d_L = 0) + p(c_L = 1)p(b_R = 1)p(d_L = 1)$$

$$p(a_L = 1) = p(c_L = 0)p([b_R \odot d_L] = 1) + p(c_L = 1)p([b_R \odot d_L] = 0)$$

$$p(a_L = 1) = p(c_L = 0)p(b_R = 1)p(d_L = 1) + p(c_L = 1)p(b_R = 1)p(d_L = 0)$$

$$p(a_L = 1) = p(c_L = 0)p(b_R = 1)p(d_L = 1) + p(c_L = 1)p(b_R = 1)p(d_L = 0)$$

14

$$LR(a_L) = \frac{p(a_L = 0)}{p(a_L = 1)}$$

$$LR(a_L) = \frac{p(c_L = 0)p(b_R = 0)p(d_L = 0) + p(c_L = 1)p(b_R = 1)p(d_L = 1)}{p(c_L = 0)p(b_R = 1)p(d_L = 1) + p(c_L = 1)p(b_R = 0)p(d_L = 0)}.$$

By dividing $LR(a_L)$'s denominator and nominator to $p(c_L = 1)p(b_R = 1)p(d_L = 1)$, we can get

$$LR(a_L) = \frac{\dfrac{p(c_L = 0)p(b_R = 0)p(d_L = 0) + p(c_L = 1)p(b_R = 1)p(d_L = 1)}{p(c_L = 1)p(b_R = 1)p(d_L = 1)}}{\dfrac{p(c_L = 0)p(b_R = 1)p(d_L = 1) + p(c_L = 1)p(b_R = 0)p(d_L = 0)}{p(c_L = 1)p(b_R = 1)p(d_L = 1)}}$$

resulting that

$$LR(a_L) = \frac{\dfrac{p(c_L = 0)p(b_R = 0)p(d_L = 0)}{p(c_L = 1)p(b_R = 1)p(d_L = 1)} + 1}{\dfrac{p(c_L = 0)}{p(c_L = 1)} + \dfrac{p(b_R = 0)p(d_L = 0)}{p(b_R = 1)p(d_L = 1)}}.$$

Now equation is ready to be represented by likelihoods such as

$$LR(a_L) = \frac{LR(c_L)LR(b_R)LR(d_L) + 1}{LR(c_L) + LR(b_R)LR(d_L)}.$$

Inspiring from the calculation of $LR(a_L)$, we can form the equations of $LR(b_L)$, $LR(c_R)$ and $LR(d_R)$ by using

$$b_L = [a_R \oplus c_L] \odot d_L$$

$$c_R = a_R \oplus [b_R \odot d_L]$$

$$d_R = [a_R \oplus c_L] \odot b_R.$$

In this scope, with a similar approach using

$$b_L = [a_R \oplus c_L] \odot d_L$$

we calculate $LR(b_L)$ as

$$p(b_L = 0) = p([a_R \oplus c_L] \odot d_L = 0)$$

$$p(b_L = 0) = p([a_R \oplus c_L] = 0)p(d_L = 0)$$

$$p(b_L = 0) = [p(a_R = 0)p(c_L = 0) + p(a_R = 1)p(c_L = 1)]p(d_L = 0)$$

$$p(b_L = 1) = p([a_R \oplus c_L] \odot d_L = 1)$$

$$p(b_L = 1) = p([a_R \oplus c_L] = 1) p(d_L = 1)$$

$$p(b_L = 1) = [p(a_R = 0) p(c_L = 1) + p(a_R = 1) p(c_L = 0)] p(d_L = 1)$$

$$LR(b_L) = \frac{p(b_L = 0)}{p(b_L = 1)}$$

$$LR(b_L) = \frac{[p(a_R = 0) p(c_L = 0) + p(a_R = 1) p(c_L = 1)] p(d_L = 0)}{[p(a_R = 0) p(c_L = 1) + p(a_R = 1) p(c_L = 0)] p(d_L = 1)}$$

$$LR(b_L) = LR(d_L) \frac{1 + LR(a_R) LR(c_L)}{LR(a_R) + LR(c_L)}.$$

In a similar manner, using

$$c_R = a_R \oplus [b_R \odot d_L]$$

we can calculate $LR(c_R)$ as

$$p(c_R = 0) = p(a_R = 0) p([b_R \odot d_L] = 0) + p(a_R = 1) p([b_R \odot d_L] = 1)$$

$$p(c_R = 0) = p(a_R = 0) p(b_R = 0) p(d_L = 0) + p(a_R = 1) p(b_R = 1) p(d_L = 1)$$

$$p(c_R = 1) = p(a_R = 0) p([b_R \odot d_L] = 1) + p(a_R = 1) p([b_R \odot d_L] = 0)$$

$$p(c_R = 1) = [p(a_R = 0) p(b_R = 1) p(d_L = 1) + p(a_R = 1) p(b_R = 0) p(d_L = 0)]$$

$$LR(c_R) = \frac{p(c_R = 0)}{p(c_R = 1)}$$

$$LR(c_R) = \frac{p(a_R = 0) p(b_R = 0) p(d_L = 0) + p(a_R = 1) p(b_R = 1) p(d_L = 1)}{p(a_R = 0) p(b_R = 1) p(d_L = 1) + p(a_R = 1) p(b_R = 0) p(d_L = 0)}$$

$$LR(c_R) = \frac{LR(a_R) LR(b_R) LR(d_L) + 1}{LR(a_R) + LR(b_R) LR(d_L)}.$$

In a similar manner, using

$$d_R = [a_R \oplus c_L] \odot b_R$$

$LR(d_R)$ can be calculated as

$$p(d_R = 0) = p([a_R \oplus c_L] \odot b_R = 0)$$

$$p(d_R = 0) = p([a_R \oplus c_L] = 0)p(b_R = 0)$$

$$p(d_R = 0) = [p(a_R = 0)p(c_L = 0) + p(a_R = 1)p(c_L = 1)]p(b_R = 0)$$

$$p(d_R = 1) = p([a_R \oplus c_L] \odot b_R = 1)$$

$$p(d_R = 1) = [p(a_R = 0)p(c_L = 1) + p(a_R = 1)p(c_L = 0)]p(b_R = 1)$$

$$LR(d_R) = \frac{p(d_R = 0)}{p(d_R = 1)}$$

$$LR(d_R) = \frac{[p(a_R = 0)p(c_L = 0) + p(a_R = 1)p(c_L = 1)]p(b_R = 0)}{[p(a_R = 0)p(c_L = 1) + p(a_R = 1)p(c_L = 0)]p(b_R = 1)}$$

$$LR(d_R) = LR(b_L)\frac{1 + LR(a_R)LR(c_L)}{LR(a_R) + LR(c_L)}.$$

Finally, the derived formulas for the belief propagation algorithm can be written together like shown in (2.9)-(2.12).

$$LR(a_L) = \frac{LR(c_L)LR(b_R)LR(d_L) + 1}{LR(c_L) + LR(b_R)LR(d_L)} \tag{2.9}$$

$$LR(b_L) = LR(d_L)\frac{1 + LR(a_R)LR(c_L)}{LR(a_R) + LR(c_L)} \tag{2.10}$$

$$LR(c_R) = \frac{LR(a_R)LR(b_R)LR(d_L) + 1}{LR(a_R) + LR(b_R)LR(d_L)} \tag{2.11}$$

$$LR(d_R) = LR(b_L)\frac{1 + LR(a_R)LR(c_L)}{LR(a_R) + LR(c_L)} \tag{2.12}$$

Polar encoding and decoding operations can be graphically demonstrated as in Fig. 2.11 via factor graph formation. The factor graph of $\mathbf{G}_8$, $N = 8$, in Fig. 2.11 consists of 2×2 basic computational blocks (BCB), in other words, processing elements (PE). Each stage of graph includes four PEs where iterative process is followed. As noticed number of stages n, determined by $\log_2 N$.
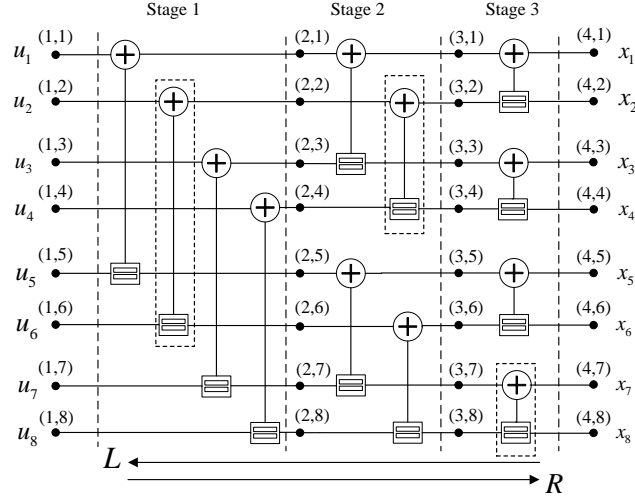
Figure 2.11 Encoder and decoder structure for $N = 8$

Generalized version of the kernel polar decoder, PE, contains two inputs and two outputs as shown in Fig. 2.12 where R represents the messages propagating rightward where L represents the messages propagating to the leftward. Messages are in the form of LR as shown by (2.9)-(2.12).
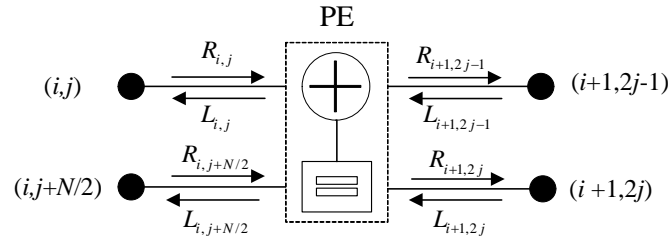
Figure 2.12 Basic processing element (PE) of BP-based polar decoder

The propagating messages for any PE on a factor graph like in Fig. 2.11, use equations (2.13)-(2.16) which are adapted from the derived likelihood ratios in (2.9)-(2.12) of the kernel iterative decoder unit of Fig. 2.12. Thus,

$$L_{i,j} = \frac{1 + L_{i+1,2j-1}L_{i+1,2j}R_{i,j+N/2}}{L_{i+1,2j-1} + L_{i+1,2j}R_{i,j+N/2}} \tag{2.13}$$

$$L_{i,j+N/2} = L_{i+1,2j}\frac{1 + R_{i,j}L_{i+1,2j-1}}{R_{i,j} + L_{i+1,2j-1}} \tag{2.14}$$

18

$$R_{i+1,2j-1} = \frac{1 + R_{i,j}L_{i+1,2j}R_{i,j+N/2}}{R_{i,j} + L_{i+1,2j}R_{i,j+N/2}} \tag{2.15}$$

$$R_{i+1,2j} = R_{i,j+N/2}\frac{1 + R_{i,j}L_{i+1,2j-1}}{R_{i,j} + L_{i+1,2j-1}} \tag{2.16}$$

can be expressed as

$$L_{i,j} = g(L_{i+1,2j-1}, L_{i+1,2j} + R_{i,j+N/2}) \tag{2.17}$$

$$L_{i,j+N/2} = L_{i+1,2j} + g(R_{i,j}, L_{i+1,2j-1}) \tag{2.18}$$

$$R_{i+1,2j-1} = g(R_{i,j}, L_{i+1,2j} + R_{i,j+N/2}) \tag{2.19}$$

$$R_{i+1,2j} = R_{i,j+N/2} + g(R_{i,j}, L_{i+1,2j-1}) \tag{2.20}$$

where $g(x,y)$ function defined as

$$g(x, y) = \frac{1 + xy}{x + y}. \tag{2.21}$$

The initial value of the messages should be determined before the iterations start, for instance, $R_{1,2,...,n+1}$ values should be set to a positive number for frozen indexed nodes. Initial frozen value can be set to 10, 100 or 1000 according to magnitude of code length. Similarly $L_{n+1,j}$ values should be initialized to the channel likelihoods which are calculated as

$$L_{n+1,j} = \frac{p(y_i \mid x_i = 0)}{p(y_i \mid x_i = 1)}. \tag{2.22}$$

$p(y_i \mid x_i)$ is the channel transition probability for channel input x_i and channel output y_i. When the M number of iterations are completed, a hard decision is made according to

$$\hat{u}_i = \begin{cases} 1 & L_{i,1} < 1 \\ 0 & otherwise \end{cases} \tag{2.23}$$

where $\hat{u}_i$ is estimated bit.

As the code length increases, the implementation of (2.9) to (2.12) requires significant amount of hardware resource and results in latency for BP decoder. To alleviate the large complexity and latency issues, logarithmic version of the formulas (2.9) to (2.12) are considered for hardware implementations, i.e., multiplications are converted to

19

summations while divisions are converted to subtractions. The $g(x,y)$ function defined in (2.18) is expressed in log domain [18] as

$$gl(x, y) \cong \mathrm{s}ign(x)sign(y)\min\left(|x|,|y|\right) \tag{2.24}$$

for BP decoding of LDPC codes. Log-Likelihood Ratio is used to represent new version of the propagation messages. Additionally, initial conditions and likelihood calculations should be expressed in log domain. The log domain equivalent of (2.9) to (2.12) can be written as

$$L_{i,j} = gl(L_{i+1,2j-1}, L_{i+1,2j} + R_{i,j+N/2}) \tag{2.25}$$

$$L_{i,j+N/2} = L_{i+1,2j} + gl(R_{i,j}, L_{i+1,2j-1}) \tag{2.26}$$

$$R_{i+1,2j-1} = gl(R_{i,j}, L_{i+1,2j} + R_{i,j+N/2}) \tag{2.27}$$

$$R_{i+1,2j} = R_{i,j+N/2} + gl(R_{i,j}, L_{i+1,2j-1}) \tag{2.28}$$

where the $gl(.)$ function is defined in (2.24).

Although the main aspect of this thesis is on BP-based decoding of polar codes, we need to introduce SC and SCL polar decoders. Since, our aim is to catch the error correction performance of a SC based technique e.g. state-of-the-art CRC aided SCL polar decoder, we should give a brief introduction to understand the methods.

In SC algorithm [2], information bits are decoded step by step. Briefly, decoding decision on current bit is done by using channel output and all previously estimated information bits. Unlike ML decoding, changing previous decisions is not allowed in SC decoding so that decoding process continues with already estimated bits. This feature of the algorithm, lower its complexity while reducing its error correction performance.

SC list decoding of polar codes is proposed in [10] to lead better error correction. Instead of tracking one best decoding path like in SC, L best decoding paths are followed in parallel in SCL decoder. By utilizing list concept, performance closes to ML decoder's. Even more, SCL decoder enhanced by introducing CRC aid. If list size L is large enough, ML decoding performance is shown to be achieved [10].

CHAPTER 3

BELIEF PROPAGATION BASED DECODING OF POLAR CODES

3.1 Art of BP Polar Decoding

The proven potential of BP-based decoding algorithms has led researchers to improve the performance of the BER / BLER polar BP decoder. As stated before, the BP polar decoder has weak error correction performance. On the other hand, LPDC codes with BP-based iterative decoding gives excellent results. Because of the low performance on BER / BLER and high complexity, a BP-based polar decoder has bandwidth issue when compared to SC-based polar decoders. Over the last five years, new approaches have been developed to adapt the BP algorithm to polar codes with high error correction capabilities. We categorize these new approaches in eight sub-sections, and explain them in detail.

3.1.1 Scaled Min-Sum BP Polar Decoder

Complexity of encoders and decoders is an important indicator whether it is applicable on hardware or not. In order to lower complexity, logarithmic approximations of equations are used in decoders. Logarithmic versions of the node messages are presented with Log-Likelihood Ratio (LLR). LLR values can also be used in log domain implementation of the BP algorithm. Despite its advantage over complexity, using log domain equivalent of the equations causes BER/BLER performance degradation due to approximations. In this scope, a study to improve log domain version of BP algorithm called as scaled min-sum (SMS) decoding algorithm is proposed in [19] where log domain equations of the propagating messages are given as

$$L_{i,j} = s \times sign(L_{i+1,2j-1})sign(L_{i+1,2j} + R_{i,j+N/2})\min\left(\left|L_{i+1,2j-1}\right|, \left|L_{i+1,2j} + R_{i,j+N/2}\right|\right) \quad (3.1)$$

$$L_{i,j+N/2} = L_{i+1,2j} + sign(R_{i,j})sign(L_{i+1,2j-1})\min\left(\left|R_{i,j}\right|,\left|L_{i+1,2j-1}+L_{i+1,2j-1}\right|\right) \qquad (3.2)$$

$$R_{i+1,2j-1} = s \times sign(R_{i,j})sign(L_{i+1,2j}+R_{i,j+N/2})\min\left(\left|R_{i,j}\right|,\left|L_{i+1,2j}+R_{i,j+N/2}\right|\right) \qquad (3.3)$$

$$R_{i+1,2j} = R_{i,j+N/2} + sign(R_{i,j})sign(L_{i+1,2j-1})\min\left(\left|R_{i,j}\right|,\left|L_{i+1,2j-1}\right|\right). \qquad (3.4)$$

Scaling factor s is added to propagation equations to compensate the performance loss. It is shown that with scaling factor $s = 0.9375$, performance improvement is achieved when compared to the conventional BP decoder utilizing $P(1024,512)$ on AWGN channel $s = 0.9375$ [19, Fig. 2]. Error correction performance is increased by using scaling factor but complexity in term of logic gate count is increased with inserted extra multiplication. Besides, critical path delay caused by gate latency is also increased. SMS based BP method is used as standard decoder in the studies [20, 21, 22, 23, 24, 25, 26, 27, 28]. Even more, processing element (PE) of SMS BP polar decoder is optimized by using high-speed parallel prefix. Ling adder is used instead of carry ripple adder to reduce logic gate delay i.e. critical path delay. Overall, critical path of a PE of SMS BP polar decoder is decreased from 2.594 ns to 0.959 ns [26, Table II].

Similar to SMS BP decoder, different approximations of the logarithmic version of the propagating message equations are presented in [29]. All three approximations achieved better BLER performance over MS BP decoder with a margin of complexity increment.

3.1.2   Parity-Check Matrix Based BP Polar Decoders

Parity check matrix of a polar code **H** can be obtained from the columns of the generator matrix of polar code **G** considering the frozen bit locations [8]. It is not applicable to use **H** during BP decoding because of its high density. Instead of using **H** through decoding stages, its use as an early detection and termination method is proposed [30]. Nonetheless, An **H** based BP polar decoder employing an adaptive approach is introduced in [31]. Adaptive approach to generate sparse version of **H** is presented to increase the convergence accuracy of the iterative decoder. Adaptive approach can be outlined as:

- Method is presented for the logarithmic version of the BP polar decoder, so firstly calculate the absolute values of received LLRs,

- Sort these absolute values in ascending order and name resulting sequence as **y**,
- A vector **B** of length $N - K$ denoting the indices of least reliable bits of **y** is generated,
- $\mathbf{H_B}$ is constructed from the columns of **H** using **B**,
- Gauss elimination is applied on $\mathbf{H_B}$ to reduce the matrix to be an identity matrix,
- New matrix $\mathbf{H}_{new}$ is formed.

By using $\mathbf{H}_{new}$ in the decoding, better error correction performance in terms of BER and BLER is obtained when compared with the state-of-the-art SCL decoder. Unfortunately, adaptive matrix construction causes complexity and latency increment. Moreover, operations like absolute value calculation, sorting, Gauss elimination for each received codeword are not feasible for long length codes. Therefore, a simpler parity-check matrix based BP decoder is presented in [32]. Simpler version of the parity-check matrix based BP decoder is achieved by converting its factor graph from high-density factor graph to sparse graph as shown in Fig. 3.1b.

As mentioned before, there are variable nodes denoted by $VN_{1,j}$ and check nodes denoted by $CN_{i,j}$ on the factor graph of polar decoder as depicted in Fig. 3.1. Messages, used to estimate original data, propagate from left to right and right to left. In this scope, variable node messages are multiplied with their previous values during iterations to improve reliability [32]. By using this idea, 1-2 dB gain is achieved over original BP scheme. Moreover, average number of iterations is reduced by 10-25 even though no early detection method is applied.
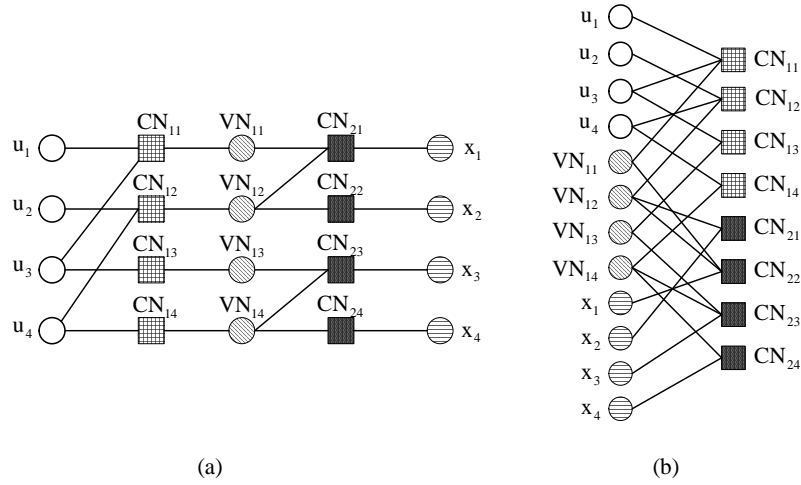
Figure 3.1 (a) Variable and check nodes for **G**$_4$, (b) sparse Tanner graph of BP polar code, **G**$_4$

3.1.3   Modified BP Polar Decoder with Check Nodes

Check nodes can be inserted into both kind of nodes i.e., frozen and information nodes. Check node addition into factor graph of the polar decoder can improve error correction performance [33]. In this study, check nodes are used to keep already converged nodes stable. Keeping converged nodes same avoids oscillation errors. Frozen and information check nodes denoted by $c(i,j)$ are depicted in Fig. 3.2.
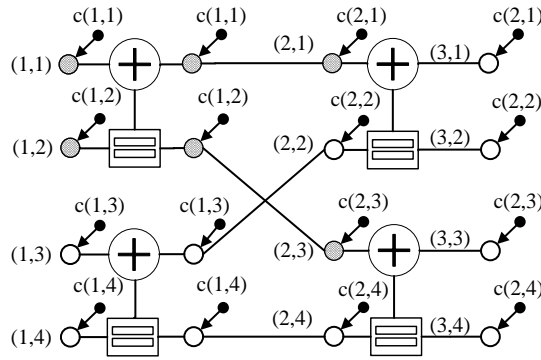
Figure 3.2 Factor graph representation involving frozen and information check nodes for $N = 4$

Frozen nodes, filled with dashed lines in Fig. 3.2, shouldn't be updated during decoding stages. Otherwise, less reliable or falsely converged frozen node likelihoods can be obtained. In order to avoid false convergence of frozen nodes, addition of frozen check

nodes is proved useful [33]. Employment of check node for information nodes is a useful approach such that they increase the reliability of the messages from previous information nodes. It is also demonstrated that the modified BP decoder utilizing $\mathcal{P}(2048,1024)$ achieves 0.5 dB gain over conventional BP polar decoder with 60 iterations at 10^{-6} BER.

Similar to check node concept, threshold value check is introduced to the nodes of BP polar decoder in [34]. A threshold value for each node is calculated using Gaussian approximation. During decoding process, if likelihood of any node reaches this threshold, then this node is set to infinity. This modification lead decoder to reach convergence faster than the conventional approach.

A multi-stage decoding process is proposed in [35] with the name of BP-bit strengthening (BPBS). In this method, bit strengthening is applied when decoding fails after pre-defined number of iterations are performed. Bit strengthening is provided by sub-factor graph based check. In this approach, already converged information bits are set to infinite. After bit strengthening is completed, another decoding process takes place. Overall, error correction performance close to SCL and 0.4 dB better BER performance over conventional BP is achieved.

3.1.4 Concatenated Decoders

Concatenated codes, as a sub-class of error correcting codes, are constructed by combining an inner code and an outer code. Concatenated codes are offered by Forney [36] to avoid exponential decoding complexity for large block lengths of the codes. In theory, two relatively short codes can decrease error probability as low as a long error correcting code can do.

A concatenated code utilizing a RS code as outer code and a polar code as inner code is presented in [37] where BP polar decoder is used for decoding operation. It is shown that with high rate RS code, high probability of block error of concatenated code decays sub-exponentially with increasing block length. Since asymptotic cases are considered in [37], concatenated decoders for practical communication systems that involve polar codes and LDPC codes are studied in [38, 39, 40, 41, 42, 43, 44, 45, 46].

LDPC codes are one of the linear codes consisted of message and check nodes that can be represented using sparse bipartite graphs. Fig. 3.3 shows a sample LDPC code for better visualization. There are n message nodes (nodes on the left) and check nodes (nodes on the right). An LDPC code is designed to generate codewords associated with the n message nodes. Sum of each generated codeword in check nodes equal to zero.

$$x_1 + x_2 + x_3 + x_4 + x_6 + x_8 + x_{10} = 0$$

$$x_1 + x_3 + x_4 + x_7 + x_8 + x_9 + x_{10} = 0$$

$$x_2 + x_4 + x_8 = 0$$

$$x_1 + x_5 + x_7 + x_8 + x_9 + x_{10} = 0$$

$$x_3 + x_4 + x_5 + x_7 + x_9 = 0$$
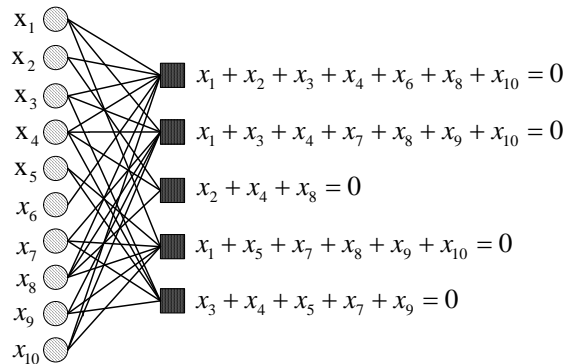
Figure 3.3 An LDPC code

With its near Shannon's capacity and being able to be decodable by iterative BP-based algorithm, make LDPC codes a good candidate to be concatenated with BP-based polar decoders. Besides, LDPC codes are utilized in large scale of frameworks like Worldwide Interoperability for Microwave Access (WiMAX), 10GBASE-T 10 Gbit/s (1,250 MB/s) Ethernet over unshielded twisted pair (802.3AN), Digital Video Broadcasting — Second Generation Terrestrial (DVB-T2), Digital Multimedia Broadcast-Terrestrial/Handheld (DMB-T/H), and Digital Terrestrial Multimedia Broadcast (DTMB).
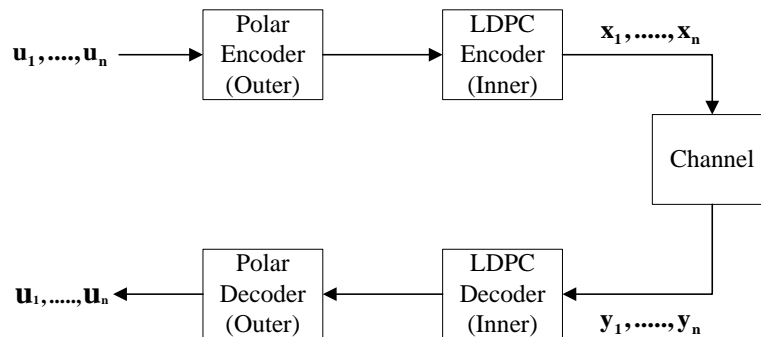
Figure 3.4 Concatenated code structure involving polar and LDPC codes

Fig. 3.4 demonstrates a concatenated encoder and decoder design consisting of polar code as outer code and LDPC code as inner code. The terms **u**, **x**, **y**,**û** represent data, encoded data, received data and estimated data, respectively. It is known that although LDPC code has decent waterfall characteristics, it suffers from error floor. Since polar codes shows no error floor up to 10^{-12} BER, it will be a good idea to concatenate LDPC and BP polar decoders [41]. The proposed method deals with error floor problem of LDPC and implemented on optical fiber links that supports data rates up to 100 Gbps. RS, LDPC and BCH codes have already been standardized for different optical transport networks. Similar error probability is achieved with code rate 0.93 when polar-LDPC code is used. It is also an advantage that both scheme can be decoded using soft decision based BP algorithm when hardware implementation is considered.

[38, Fig.4] demonstrates that polar-LPDC code is a good candidate to cover optical transport network standard's requirements in terms of BER performance and interchangeable code-rates. Hardware implementation of polar-LPDC code [38, 39] is performed in [42]. Moreover, a modification to polar-LDPC cascaded structure of [38, 39] is presented in [43]. Modification is done by adding an influence factor to the junction of two decoders' factor graphs. Soft messages coming from polar decoder are multiplied with this influence factor. Various influence factor values are studied in [43] and better BER performance is observed.

A cleverer concatenated scheme that involves polar and LDPC code is introduced in [44]. In this scheme, LDPC code is only applied to the non-polarized bit channels as depicted in Fig. 3.5. Other channels of the polar code stay untouched. Non-polarized (intermediate) channels are selected by introducing two threshold values that separate good and bad channels from intermediate channels. Three channel types can be defined considering the threshold values δ_1 and δ_2 as

- good channels, u_{good}, $Z(W_N^i) < \delta_1$
- intermediate channels, u_{inter}, $\delta_1 < Z(W_N^i) < \delta_2$
- bad channels, u_{bad}, $Z(W_N^i) > \delta_2$

where $0 < \delta_1 \leq \delta_2 < 1$ and where $Z(W_N^i)$ stands for Bhattacharyya parameter that is accepted as an upper bound for the maximum probability of transmission errors [2]. In order to perceive the effect of partially coding with LDPC code, two different scheduling schemes to be utilized for BP polar decoder are introduced, where one of them is the conventional BP algorithm, and other one is the soft successive cancelation (SCAN) BP algorithm.

Decoding speed of a BP-based decoder is strictly related with the average number of iterations performed. A conventionally scheduled BP polar decoder over-performed the improved BP polar decoder (polar-LDPC concatenated) of [44] in terms of average number of iterations [44, Fig.4]. However, when BP polar decoding is scheduled with SCAN algorithm polar-LDPC structure has lower average number of iterations than simple BP polar decoder. Success of SCAN scheduled polar-LDPC decoder comes from the polar code construction method. Bhattacharyya parameter based code construction method used in [44] is optimized for SC algorithm in [2]. Since SCAN is a soft version of SC algorithm, using SCAN scheduling in BP polar decoder is a beneficial approach. However, SCAN scheduling is a slow algorithm when compared conventional scheduling. SCAN decodes codewords bit by bit while conventional BP has parallel decoding capability.
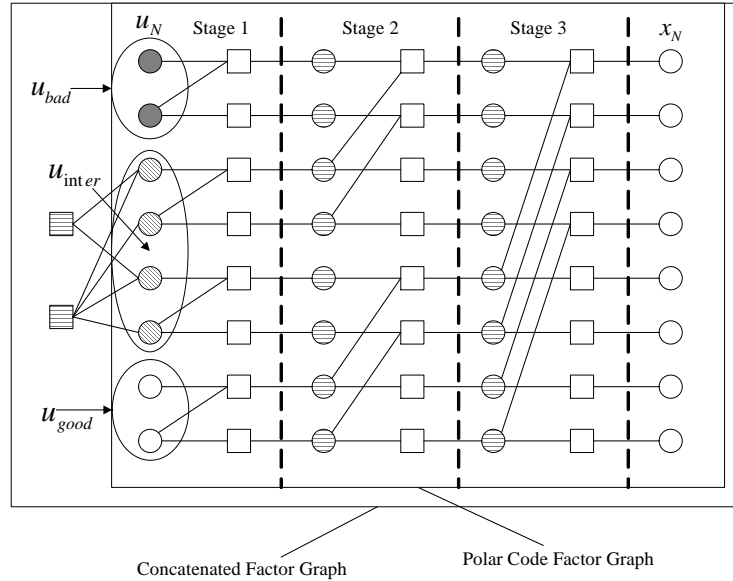
Figure 3.5 Factor graph for improved BP decoding algorithm

EXIT charts are used to design short LDPC codes using scattered extrinsic information transfer. An LDPC code designed using EXIT chart is used in a polar-LDPC cascaded design in [45]. Simulation results showed that the proposed design achieves 0.4 dB performance gain over conventional BP while 0.2 dB improvement is observed over polar-LDPC code of [44] at 10^{-5} BER.

As mentioned before, LDPC code is utilized for intermediate capacity bit channels of the inner polar code [44] to improve concatenated code performance. Furthermore, intermediate channels are sorted with respect to their capacities and then LDPC coding is applied [46]. A bit mapper is used to sort bit channels. Bit mapper makes sorting by comparing intermediate channels' leaf set sizes. It is shown in [39] that leaf set size is directly proportional with the protection rate of a bit channel. Since Bhattacharyya parameter based selection does not show its optimal performance on BP polar decoder, this kind of bit mapper that takes advantage of leaf set size shows better performance. As a result, the suggested structure of [46] has 0.3 dB gain over the study with EXIT charts in [44] and 0.5 dB gain over conventional BP scheme at 10^{-5} BER.

Another method that includes the use of LPDC codes for intermediate channels is proposed in [47]. Bit channels having the same leaf set size are sorted in descending order

based on Bhattacharyya parameter. After sorting, the number of bit channels are selected to be used for the outer LDPC code. When compared to the similar studies of [44], [46], and [47], it is obvious that leaf set size is an important parameter on BP decoder's performance.

As Arıkan stated in [2] polar codes are constructed in a recursive manner by using a kernel unit. Moving from this fact, larger polar codes as a concatenation of several smaller polar codes can also be constructed [48]. [48, Fig.2] demonstrates the code generation via concatenated code structures. A joint structure involving a short auxiliary polar code, an interleaver and an inner polar code is introduced in [48] where the aim of the interleaver is to force the likelihood values that flows from one polar encoder/decoder to another to be statistically independent of each other. As expected, auxiliary polar code is used to protect long polar code's semi-polarized information bits. Factor graph representation of the method is given in Fig. 3.6 where **y** stands for likelihood ratio of received data and **û** represents estimated data of decoder. Besides, polar code structures are connected to each other using interleaver/deinterleaver (π/π^{-1}) blocks.
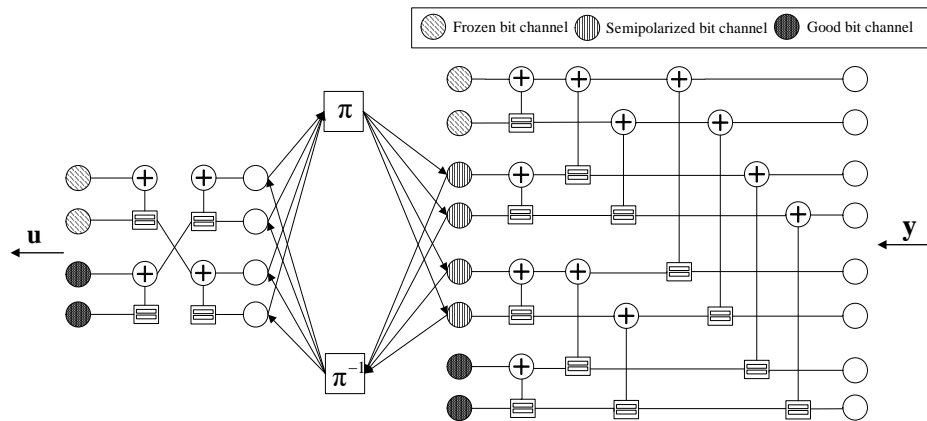
Figure 3.6 Shorter length polar codes are used as auxiliary codes for larger length polar code

A different version of [48] involving one auxiliary and two inner polar codes is presented in [49]. Proposed scheme achieves a gain of 0.3 dB at 10^{-5} BER when compared to the conventional BP polar decoder.

Two fundamental schemes that use belief propagation algorithm, sparse code multiple access (SCMA) and polar code decoder are combined in [50]. Proposed method combines the factor graphs of two schemes in a way such that the probability information between them can be circulated. Overall, a decoder with higher precision and faster convergence is achieved.

3.1.5 Hybrid Decoders

Hybrid decoding approach is applied with different types of polar decoders. Two different hybrid decoding approaches are studied in order to make BP-based polar decoders preferable in 5G frameworks. In this scope, first type of hybrid decoder follows a flexible process such that only one type decoder works and the other one works if and only if the first one fails. A hybrid decoding approach that utilizes both BP decoding and SC decoding is presented in [51, 52]. Whenever BP decoder fails, SC decoding algorithm is performed. As a result, 0.2 dB improvement over basic BP-based polar decoder is achieved.

Following the introduction of BP-SC hybrid decoding approach, hybrid decoding of BP-SCL is introduced in [53, 54]. When throughput of the decoder is paramount, it is obvious that a BP-SCL decoder is more advantageous than SCL algorithm. To avoid unnecessary iterations of BP decoder when it is already converged, CRC control is unified with BP-SCL hybrid decoder. If CRC is not satisfied for maximum number iteration M on BP decoder, then CRC aided SCL is performed. Proposed BP-SCL scheme is utilized [36] on $P(4096,2048)$, and same BER performance is achieved when compared to SCL decoding with list size $L = 32$. Despite its handicap on decoding speed, it is also demonstrated that the presented hybrid structure has lower latency after certain SNR [36, Fig.5a].

Second type of hybrid approach is presented for 5G NR eMBB where LDPC codes are used to correct errors on data channel while polar codes are utilized for the protection and correction of the control channel information. Since BP-based algorithms are used to decode LDPC codes if we use BP-based decoding for polar codes, then a combined, hardware friendly structure can be achieved. Instead of utilizing two separate decoder for data and control channels, employing one decoder for two separate channels will lower hardware consumption. Besides, time division duplexing (TDD) is going to be used in

eMBB communication services, control and data channel information will flow in separate time slots. If the polar codes are chosen to be decoded by BP-based algorithm, then a single hybrid decoder can handle whole operation [55, 56]. When the results of these studies are evaluated, it is seen that the combined polar code and LPDC decoders are promising for future studies.

3.1.6 Multi-trellis BP Decoding

There can be $n!$ different Tanner graph representation of BP polar decoder [4, 57] where N is code length and $n = \log_2 N$. By using different permutation of the n layers of the connection, multi-trellis representations can be achieved. In this scope, Fig. 3.7a represents original factor graph of the generator matrix of the polar code $\mathbf{G}_8$, and Fig. 3.7b and Fig. 3.7c depict its two differently permutated trellis. It is important to state that output of each trellis stays the same when the same information bits are used [58].
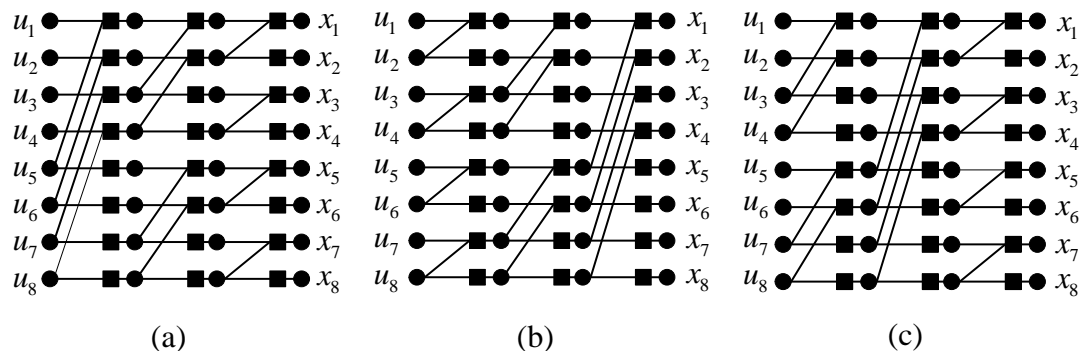
Figure 3.7 Permutations of trellis structure for $\mathbf{G}_8$

In [59], multi-trellis approach is used to eliminate error floor problem. In this study, whenever the decoder fails, a different permutation of the trellis structure is tried. Permuted factor graph is also utilized as an early detection and termination method to BP decoding of polar codes [60]. Overall, multi-trellis approach decreases the average number of iterations of BP polar decoder. It is also observed that 0.4 dB gain at BER 10^{-6} over SCL with list size $L = 32$ is achieved. However, CRC aided SCL still outperforms the proposed scheme [60, Fig.4].

Further improvement on multi-trellis approach of [60] is given in [61] by selecting permuted factor graphs more cleverly. A process to pick permutations that lead to better

error correction performance is introduced in [61]. It is shown that 0.2 dB improvement is achieved at 10^{-4} BLER for $t = 10$, where t is number of different permutations considering the case in which t number of alternative permutations are selected randomly. If a large number of alternative factor graphs are used, more improvement is observed.

As mentioned before, the main advantage of BP algorithm over SC/SCL algorithms is that its parallel decoding ability allows high throughput of decoding. Parallel decoding ability is defected when different permutations are used to get the correct codewords. As the number of utilized permutations is increased, throughput of the multi-trellis BP decoder decreases. In order to overcome this drawback, a BP-list (BPL) approach is introduced in [62]. BPL process saves the parallel decoding capability of the BP structure while adding complexity to the system. Proposed BPL algorithm has a good error correction performance and it is only 0.5 dB worse than the state-of-the-art SCL decoder at 10^{-5} BLER. Block diagram of the operation is given in Fig. 3.8 where L parallel BP decoders and for each of them **G**-matrix-based early detection and termination method are utilized. Each parallel branch runs at the same time, and BP decoders are all different from each other with respect to trellis structure. Different permutations are achieved by using k cyclic shifts of the original factor graph where $1 < k \leq L$. Estimated data and codewords, $\hat{\mathbf{u}}_i$ and $\hat{\mathbf{x}}_i$, are compared to channel output , $\mathbf{y}$, in terms of Euclidean distance. Among them the minimum one is chosen to be output of the BPL decoder. Moreover, different polar code construction method called as RM-Polar code is applied on BPL decoder in [62]. RM-Polar code construction leads to better error correction performance, and it is shown that Bhattacharyya parameter based polar code construction is optimum for SC decoder but not BP-based decoder. Detailed explanation for the RM-Polar codes can be found in [62]. One disadvantage of BPL decoding that is needed to be beaten is its complexity over SCL algorithm. Since both decoders use list decoding concept, their approach to list decoding is different. In BPL, list decoding capability is provided by using parallel decoders however, there is no parallel decoder that is utilized in SCL decoding. Nevertheless, BPL algorithm is a more promising technique than SCL with its high throughput capability. BPL is a soft decision based decoding algorithm that can be advantageous when joint forward error correction methods.
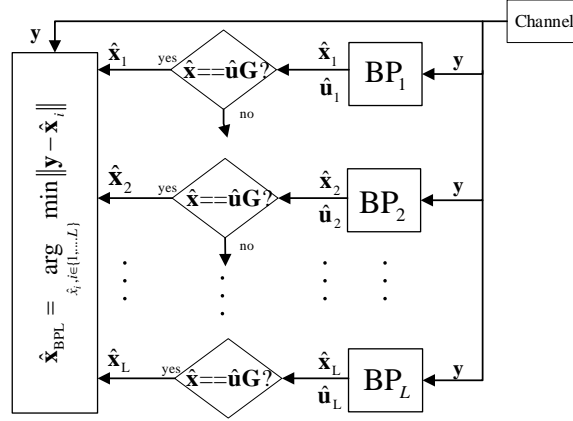
Figure 3.8 The block diagram of the multi-trellis belief propagation list decoding algorithm utilized for polar codes

As known, a factor graph consists of variable and check nodes. As expected, multi-trellis approach changes the number of frozen variable nodes for different rates of polar codes [58]. Frozen variable nodes are demonstrated in Fig. 3.10a with black color. As the number of frozen variable nodes increase, better BER performance is achieved. It is obvious that factor graphs with different trellis structures do not have same frozen variable nodes. As the code rate changes, frozen node number is calculated among permutations, and the largest one is selected as the best permuted trellis. It is shown in [63] that choosing the best permuted trellis helps use the list concept by giving this trellis a priority during decoding [63].

3.1.7 Deep Learning Based BP Decoding of Polar Codes

As a general decoding method for linear codes, it has been shown that a BP decoder may be the subject of deep learning. It has been shown that deep learning methods are useful for detecting and tracking objects based on image and video processing and achieve remarkable results with machine translation, which provides automatic translation from one language to another. In addition, significant results are observed in speech processing and recognition. Deep learning concept is also utilized in forward error correction to be able to reach Shannon's capacity. First adaptation of a BP-based decoder to deep learning concept is demonstrated in [64]. In this scope, BP-based decoding of BCH code is taken into consideration. A data set combination of 2^{45} codewords are used to train the system for BCH (63, 45) code. Huge amount of codewords makes system training difficult to

achieve. Less training data becomes sufficient by adding weights to the edges of BP decoder [64]. Adding weights means injecting multiplicative terms to BP decoder structure. By using this approach, an acceptable decoding performance is observed. To be clear, 0.4 dB gain is achieved at 10^{-2} BER for a BCH (63, 36) code over original BP-based decoder. Since multiplication terms increase complexity of the decoder, converting them into terms that are more applicable is offered in [65]. Hardware consumption is reduced by converting multiplicative weights to additive offset parameters. In addition to the hardware consumption reduction, 0.1 dB gain is achieved over the method presented in [64].

Presented studies of [64, 65] have not a complete full-scale deep learning approach such that a deep learning approach need to take care about a nonlinear activation function, hidden layers, loss function etc. [66] offers a complete deep learning process that is depicted in Fig. 3.9. Deep learning system consists of an encoder, a virtual communication channel that adds noise to the codeword, and a neural network decoder (NND). Input, hidden and output layer of NND are sorted from left to right, Fig. 3.9. For example, three hidden layers are introduced in the NND at Fig. 3.9. Unlike BP decoder, NND finds the estimates of the transmitted codeword and information bits without any iteration, and it is called as one shot decoding. Thanks to the one shot decoding feature, NND structure has superiority over BP polar decoder in terms of decoding latency and BER performance. However, large number of data set during training creates a serious deficiency.

Number of training data for a polar code $\mathcal{P}(16,8)$ ranges from 2^{10} to 2^{18} [66]. As the training data set size increases, NND's performance approaches to the Maximum a posteriori probability decoder's error correction performance. Although, training complexity and time increases exponentially, performance of a NND is marvelous with its one-shot decoding feature. In this scope, reducing training time of a NND is studied via partitioning in [67]. Instead of training long polar codewords, dividing it into smaller parts lowers the overall training time. Polar code with length $N = 128$ is divided into eight-bit NND structures. Same error correction performance is achieved when compared to SC and BP-based decoders. Since one shot decoding is utilized, throughput of NND is much higher than SC and BP-based polar decoders. Neural SC decoding utilizing smaller neural networks is studied in [68] where decoding latency is reduced up to 42.5% for $\mathcal{P}(128,64)$.
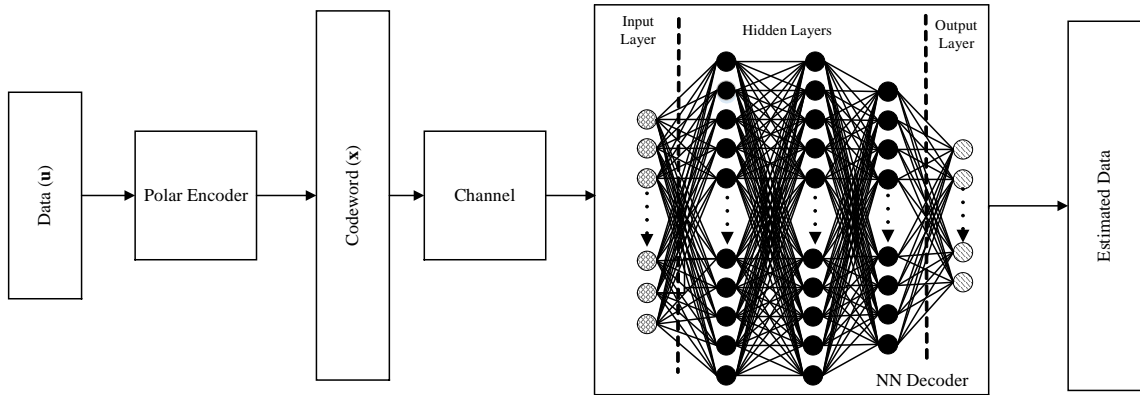
Figure 3.9 A complete deep learning setup

Previously mentioned SMS BP polar decoder [19] is proposed to compensate the performance loss on error correction caused by converting decoder process into logarithmic scale. In SMS BP, message propagation equations are modified by injection of a scaling parameter α. Inspired form SMS BP, applying different α values into different nodes of the decoder is proposed in [69]. NND structure, proposed in [69], is constructed using $2k+1$ hidden layers, where k is the iteration number that decoder is to be trained with. In other words, hidden layers are formed by tying the decoding stages to the other's tails. Training takes place to find α_i parameters where $1 < i \leq 2k+1$ for polar code $\mathcal{P}(64,32)$. It is also studied that BER performance of NND based decoder increases, as k is incremented. However, it is important to state that the structure with 6 iterations achieves the error correction performance of NND with 7 iterations [69, Fig. 6].

A concatenated code formed by CRC and polar code is utilized for NND based decoder in [70] where hidden layer of CRC is integrated into the hidden layers of polar decoder. A more detailed presentation of [70] is provided in [71, Fig. 5]. It is seen that 0.4 dB BER gain is achieved when $P(128,80)$ is used. Alternatively, a hybrid polar-LDPC decoder enhanced by deep learning is proposed for 5G eMBB framework in [72]. In 5G eMBB framework, LDPC codes are utilized in data channels whereas polar codes are utilized in control channels. In order to train a hybrid polar-LDPC decoder, an indicator section is added to choose the active one at the moment of the deep learning process. Since time division duplexing is going to be used in 5G eMBB framework, adding indication section and training both decoders together is a logical approach. Consequently, BER gain is observed for short length codes.

A smart post-processing method to increase the performance of the BP polar decoder is also introduced by the aid of deep learning in [73]. In this approach, bit flipping is applied as post-processing method when CRC on estimated data is not satisfied after maximum iterations performed. By using bit flipping enhanced by deep learning, BER/BLER performance is improved [73].

Original multi-trellis factor graph of BP polar decoder, depicted in Fig. 2.11, can be converted to an LDPC-like structure as presented in Fig. 3.1b. Unlike LDPC, polar factor graph is dense causing poor error correction performance. The study [74] offers a method to convert dense factor graph of polar decoder to its sparse version by applying pruning techniques on **G** to get **H**. Performance and throughput gain is achieved by using sparse Tanner graphs. Furthermore, sparse decoding structure is combined with neural networks using deep learning concept in [75]. Sparse neural network decoder is created for 10 iterations as shown in [75, Fig.3]. Sparse NND is designed for $P(256,128)$. Simulation results show that sparse NND outperforms MS and SMS BP polar decoders. As a result, we can state that although BP-based neural network offers one shot decoding, its long training time avoid its applicability for long code lengths, i.e., $N \geq 256$.

3.1.8 Noise-Aided BP List Polar Decoder

This type of decoder is the main subject of this thesis. Detailed explanations and performance results about noise-aided BP list polar decoder are going to take place in

chapter 4. It is important to state that noise-aided BP list polar decoder is the only BP-based decoder that approaches to the error correction performance of the state-of-art CRC aided SCL polar decoder.

3.2 Simplified BP Decoding Algorithm for Polar Codes

The complexity of the BP algorithm is a drawback for its utilization on polar codes in practical communication systems. For this reason, researchers focus on reducing the hardware consumption of the BP algorithm.

3.2.1 Node Classification and Unification Based BP Polar Decoding

Node classification seems possible when the BP factor graph of the polar decoder is inspected. There are three cases for classification. First node class is formed by frozen nodes when both of the inputs of a PE carry frozen likelihood. Second node class is formed when both of the processing element inputs carry information likelihoods. Third, a mixed node that has both frozen and information likelihood as inputs. Three different classes are depicted in Fig. 3.10a.

The first aim of the classification of the nodes is to avoid unnecessary calculations during message propagation on PEs [30]. Node labeling is applied, i.e., frozen nodes are labeled as N^0 while information nodes are labeled as N^1. Before the starting of the decoding operation, N^0 nodes are set to ∞ for initialization. PEs with two N^0 nodes have also two N^0 nodes at the output of the PE, Fig. 3.10a. In Fig. 3.10, frozen nodes are demonstrated with black color while white nodes are shown in white color. In this scope, the PE which has two N^1 nodes, has also N^1 nodes at the output. Messages on PEs consisting of the same kind of nodes are not updated during iterations to avoid unnecessary calculations. Moreover, further classification is done by introducing repetition nodes, N^{REP}, and single parity check nodes, N^{SPC}. Repetition nodes have single information bit and three frozen bits, Fig. 3.10b. Single parity check nodes have three information bits and single frozen bit, in other words, parity bit. Simplified versions of N^{REP} and N^{SPC} nodes are presented in Fig. 3.10c. By using node classification and simplification methods, authors aim to lower the complexity of the BP polar decoder [30]. Simulation results show that the average number of iterations is smaller than the average number of iterations performed for the MS and SMS BP decoders. Besides, it achieves similar error correction

performance as MS BP decoder with round trip (RT) scheduling [30]. The complexity of the BP decoding is reported to be reduced by 92.8% compared to SMS algorithm while BLER performance remains the same [30].

Node classification and simplification is presented in [76] where four different PE structures are considered. Fig. 3.11 depicts these PE structures where v_i and v_o represent input and output variable nodes respectively.
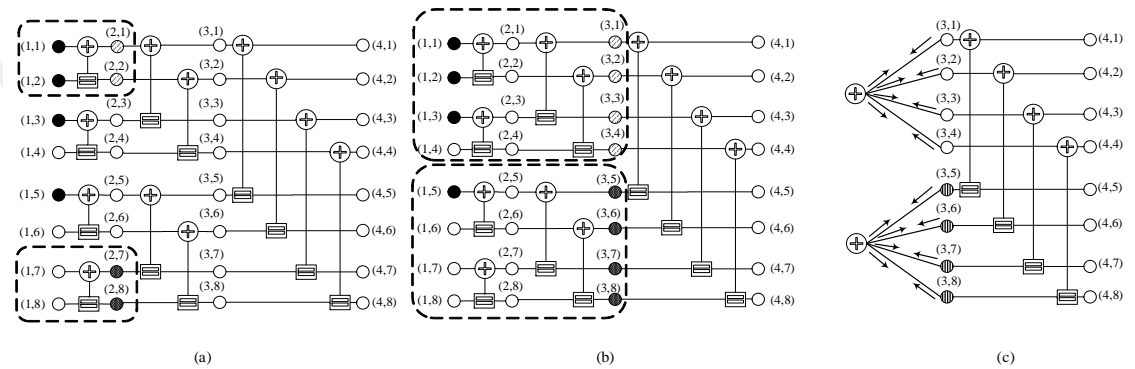
Figure 3.10 (a) Frozen nodes are in black color and information nodes are in white color (b) N^{REP} nodes are striped and N^{SPC} nodes are dashed (c) simplified N^{REP} and N^{SPC} nodes
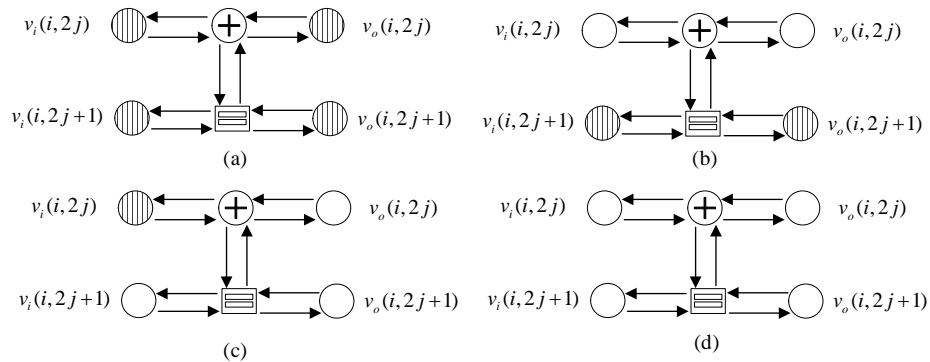
Figure 3.11 Four possible variable node permutations

Equations for left and right propagating messages are simplified. [76, Table I] shows that PE simplification provides a reduction in complexity of about 75% in terms of multiplications and summations performed. In addition, the performance of the classical BP decoding algorithm is achieved.

3.2.2 Stage-Combined BP Decoding Algorithm

As known, there are $\log_2 N$ stages in a BP polar decoder and each stage has $N/2$ processing elements. Despite its parallel decoding capability of the BP polar decoder at one clock cycle, only one stage is processed. By aiming to reduce latency of the decoder, idea of combining adjacent stages is proposed in [22]. Therefore, it is possible to reduce the latency, complexity and memory requirement of the decoder.

As depicted before, BP decoder consists of stages and each stage has a number of process elements. Parallel decoding property of the BP algorithm enables to process a stage at a time. It is clear that by combining adjacent stages, it may be possible to reduce the latency, complexity, and memory requirement for LLR values. Following this idea, a memory efficient BP decoding algorithm is proposed in [22] by merging four 2×2 PEs into a single 4×4 PE as shown in Fig. 3.12.
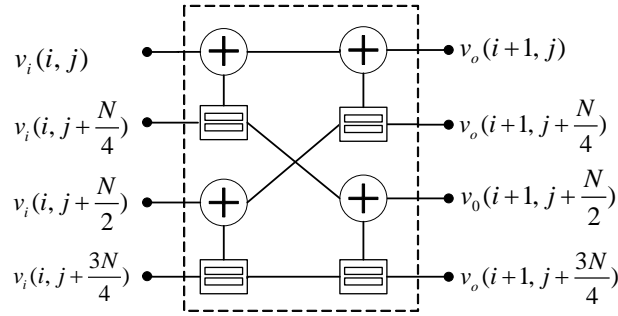
Figure 3.12 4×4 Basic processing element consisting of 2×2 PEs

Left and right propagating messages of the newly formed 4×4 PE are generated by modifying the message equations of 2×2 kernel [22, Eq.9-10]. A polar code with length 16 is constructed using 4×4 PEs as depicted in Fig. 3.13. As noticed, two less stages are used for implementation when compared to the original scheme. Besides, number of clock cycles needed to perform an iteration decreased from $2(\log_2 N)$ to $2(\log_2 N - 1)$.
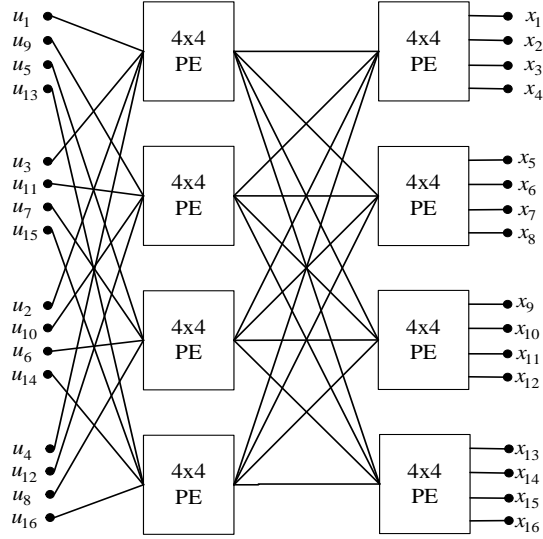
Figure 3.13 16×16 decoder consists of 4×4 PEs

Implementation of stage-combined decoder design presented in [22] is accomplished by using TSMC 45 nm Low Power CMOS technology. Overall, 18.4% area reduction is observed when compared to the implementation of original structure with code length 2^{12}. On the other hand, critical path number increases in stage-combined approach. It can be seen as a drawback when throughput of the design is the critical factor. Moreover, another stage-combined design is presented in [77], and approximately 0.2 dB gain is achieved at 10^{-4} BER for $\mathcal{P}(1024, 512)$, and storage requirement is halved.

3.2.3 Stochastic BP Decoding of Polar Codes

Another method to lower the complexity of the BP-based polar decoder is considered using stochastic computing in [78] where it is indicated the decreased complexity reduces the silicon area and power consumption of the decoder. In stochastic computing, magnitude of a probability is expressed by a number of 1's e.g. 0.6 can be represented by streams 0110110101, 1101001011 or 0111100101 and 0.5 can be represented by 0101100011 or 0111000011 [78]. In this scope, study [79] offers a stochastic BP decoding algorithm. Improvement on stochastic process is done by increasing bit streams' length, re-randomization of bit streams. As a result, BP decoder utilizing stochastic structure can be used to decrease the amount of computational complexity.

Throughput increment on stochastic BP polar decoder is studied by the proposed method of bit-wise iterative stochastic decoding architecture in [80]. Additionally, an optimized version of proposed scheme of [80] is introduced in [81] in order to reduce the hardware consumption and to obtain faster convergence.

3.2.4 Improved BP Decoding Algorithm with Modified Kernel Matrix

As mentioned before, two-input two-output (2×2) kernel is the base of a polar code. In a polar code, there are $\log_2 N$ stages of PEs with $N/2$ PEs at each stage where N is the code length. When the message propagation equations of a BP-based iterative decoder are inspected, it is obvious that iterative decoding operation needs memory units. If memory requirement of the decoder can be reduced, then BP polar decoder becomes more suitable for real time applications. To decrease the memory requirement, a decoder with 3×3 kernel is proposed in [24, 82]. Fig. 3.14 demonstrates the code structures with 2×2 PEs and 3×3 PEs.
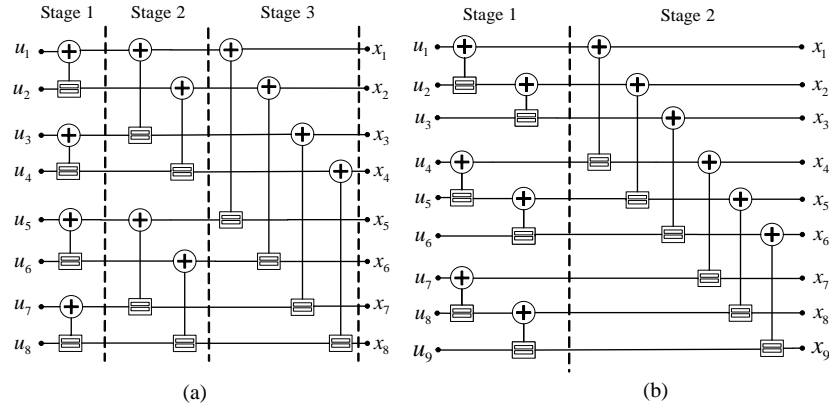
Figure 3.14 (a) 2x2 PE based encoder/decoder representation (b) 3x3 PE based encoder/decoder representation

PE with 3×3 kernel is reported to reduce the memory requirement and decoding delay by 37% [24]. Moreover, error correction performance remains the same when compared to the original structure with 2×2 kernel. Because of this part, it will be concluded that studies that try to lower the complexity of the BP polar decoder are not sufficient. However, node classification and unification based BP polar decoding seems suitable for practical applications considering the hardware implementation.

3.3 Increasing Decoding Speed

Despite its advantage on keeping decoding latency low with its parallel processing capability, BP-based decoders suffer from its iterative processing nature. Thus, decreasing the number of iterations to be performed by avoiding unnecessary iterations on already-converged nodes has been studied extensively. Originated form the idea such that well-known early detection and termination methods are valid for LDPC and turbo decoders, they are also applicable for BP polar decoder. In this section, different types of methods are presented.

Way of the message propagation, i.e. scheduling, is also vital for faster convergence of the decoder. Moreover, selected scheduling method guides the decoder into true convergence or not.

3.3.1 Early Detection and Termination Methods

An iterative BP-based decoder, either for polar codes or for LDPC codes, performs the iterations until a pre-defined number M. Most of the time, related of the channel signal-to-noise ratio, BP decoder converges before M is reached. Thus, this convergence has to be detected using some methods. Detecting convergence of the decoder and stopping its process improves the throughput of the BP decoder. In this scope, average number of iterations is an important indicator of the decoder's success. A number of techniques tries to detect the correct convergence of the BP polar decoder. A significant aspect of the techniques are their complexities. Complexity of the technique is a decisive factor on whether an early detection and termination method is applicable or not. In this section, proposed techniques in the literature are explained and a performance table is depicted for better guidance to the readers.

In general, encoding of data, $\mathbf{u}$, is accomplished multiplying it with the generator matrix, i.e., $\mathbf{x} = \mathbf{u}\mathbf{G}$. In the decoder, estimation of data $\hat{\mathbf{u}}$ and codeword $\hat{\mathbf{x}}$ can be used to check whether convergence is achieved or not. If $\hat{\mathbf{x}} = \hat{\mathbf{u}}\mathbf{G}$ is satisfied, then the decoder should be stopped. This type of test is proposed in [20] and it is named as $\mathbf{G}$-matrix-based detection. In factor graph of the decoder, leftmost nodes stand for $\hat{\mathbf{x}}$ while $\hat{\mathbf{u}}$ is valid on the rightmost part of the decoder. At each iteration, $\hat{\mathbf{x}} = \hat{\mathbf{u}}\mathbf{G}$ is checked. If condition is

satisfied, then the decoding process is stopped. Despite its complexity, it's reported that the average number of iterations is reduced by 42.5% when SNR is 3.5 dB [20].

Estimation results for the variable nodes are obtained using their likelihood values. Likelihood of variable node a is calculated by using

$$LR(a) = \frac{p(a=0)}{p(a=1)}.$$

Upon calculation of $LR(a)$, a decision or an estimation can be made by using

$$\hat{a} = \begin{cases} 1 & LR(a) < 1 \\ 0 & otherwise \end{cases}. \tag{3.5}$$

According to (3.5), $\hat{a}$ equals 1 if $LR(a)$ is smaller than 1 otherwise it equals 0. As mentioned before, decoder's operations can be performed in logarithmic domain. When likelihood calculation and message propagation equations are converted into log domain, estimation of information bits are done according to

$$\hat{a} = \begin{cases} 1 & \mathrm{LLR}(a) < 0 \\ 0 & otherwise \end{cases}. \tag{3.6}$$

Logarithmic domain conversion of BP decoder also affects the accuracy of the estimation of variable nodes. When equation (3.6) is inspected, it is seen that a hard decision can be made by only looking to the sign of the $\mathrm{LLR}(a)$. Estimation of data bits, $\hat{u}_i$, can be calculated using (3.6), and the estimated data vector $\hat{\mathbf{u}}$ is used on **G**-matrix-based early detection and termination methods. Apart from this methodology, the addition of the magnitude part of the $\mathrm{LLR}(a)$ for the detection of the converged nodes for the BP polar decoder is offered in [20]. Absolute value of $\mathrm{LLR}(a)$, $|\mathrm{LLR}(a)|$, can be utilized for the early detection and termination process such that when magnitude of $\mathrm{LLR}(a)$ is greater than a predefined threshold, β, the result of (3.6) is accepted as estimated value. Method in [20] is named as min*LLR*. It is also important to state that the choice of threshold value is very crucial. [20, Fig. 7, 8] state that different β values have considerable effects on both BLER performance and average number of iterations. Besides, an adaptive method is proposed to get an optimum threshold called as min*LLR* for early detection. Channel estimation is employed for this purpose. It is shown in [20, Fig. 7] that an adaptive min*LLR*

algorithm with $\beta = 2.5$ when $SNR \leq 3$ dB and with $\beta = 9.5$, when $SNR = 3.5$ dB gives better results, and the average number of iterations is reduced by 32.5%. To sum it up, despite to its simplicity over **G**-matrix-based detection method, the method in [20] is still worse in terms of average number of iterations performed and error correction performance.

Following the presentation of **G**-matrix-based early termination using the equality of $\hat{\mathbf{x}} = \hat{\mathbf{u}}\mathbf{G}$, a quantity of studies that utilizes the **G**-matrix method on BP polar decoder are proposed in [21, 25, 26, 27, 44, 83, 84]. One of the reasons to use the **G**-matrix method is to compare it with the newly discovered detection method, and the other reason is simply to provide better error correction performance.

Apart from the methods of **G**-matrix and min*LL*R, there are other proposed methods to avoid unnecessary iterations of the BP polar decoder. In this scope, a method that observes the decisions of the three consecutive iterations (we will call it as Observation of Consecutive Iteration results-OCI) to check whether a change on bit decisions is observed or not is proposed in [83]. If no change is detected, then it means that the decoder is converged and decoding can be terminated.

Similar to **G**-matrix-based detection, an **H**-matrix-based early detection and termination method to avoid unnecessary iterations is proposed in [31]. Parity check matrix **H** can be derived from the generator matrix **G**. Consequently, early termination can be applied using the criteria $\hat{\mathbf{x}}\mathbf{H} = \mathbf{0}$, similar to $\hat{\mathbf{x}} = \hat{\mathbf{u}}\mathbf{G}$ [31].

As presented in [31], min*LLR*-based detection method can be used as an early detection and termination method in a BP decoder. Inspiring from the min*LLR*-based detection, LLR-Magnitude aided (LMA) is proposed in [23]. In this method, LLR values of the last nodes of the BP decoder are compared with their previous values, and decoding operation is terminated if they are same. For the maximum iteration number $M_{\max} = 30$, it is observed that, 69.4% decrement in iteration number is achieved at $E_b / N_0 = 3.5$ dB.

It is also important to state that, LMA and **G**-matrix-based early detection and termination methods can make wrong decisions due to falsely converged LLRs.

CRC is a powerful technique to detect whether a received word is erroneous or not. CRC is widely used in digital networks like LAN and WAN. Usually, when CRC is not satisfied then re-transmission of the data is required. Moreover, a well-designed CRC can check perfectly whether decoder is successful or not. In this area, a 32 bit wide CRC utilized to make robust decision about whether BP polar decoder is converged or not is considered in [23, 85] i.e. early detection and termination. The use of CRC-32 decreases the average number of iterations by 82.8% at $E_b / N_0 = 3.5\,\text{dB}$ [23]. Although performance remains the same, the use of CRC reduces the decoding latency compared to LMA. Further usage of CRC as an early detection method can be found in [60,124,125].

G-matrix-based detection requires a matrix multiplication; however, matrix multiplication after each iteration increase the latency. Thus, a simplified version of **G**-matrix-based detection is proposed in [84]. Convergence check, in other words, success in decoding is checked observing a cluster of information bits that is polarized to the highest error probabilities. Method in [84] is called as worst of information bits (WIB) and WIB method can be configured using two parameters; n_{WIB}, number of WIB information bits, and M, the number of last iterations where the sign of the WIB remains the same. Performance of the WIB method depends on these two parameters. It is stated that choosing $n_{\text{WIB}} = N/8$ where $N = 2048$, successful decoding is achieved using WIB [84]. Selection of the information bits that are checked by WIB method is done using [84, Eq.(5)]. As mentioned before, aim of the introduction of WIB method is its simplicity over **G**-matrix-based method. Despite to its simplicity, the performance of the WIB method is worse than **G**-matrix-based detection. In order to overcome this defect, a channel adaptive approach for WIB is presented in [84].

A practical early detection of decoder nodes' convergence is proposed in [83, 84, 86]. In this method hard decision made on *m* consecutive iterations are observed to see whether likelihood value stays stable or not. WIB method also follows a similar approach, however, all information and frozen bits are observed in [83, 86] instead of observing estimation of a number of information bits. A more accurate but still simple method that is about observing consecutive iteration (OCI) results is presented in [87]. In this study a

threshold, (ϵ), is defined and the relation between three consecutive iterations are given as

$$0.5 \times \left[\left(L_{i,1}^{t} - L_{i,1}^{t-1} \right) + \left(L_{i,1}^{t+1} - L_{i,1}^{t} \right) \right] < \epsilon \qquad (3.7)$$

where $L_{i,1}^{t}$ stands for the current node message. Equation (3.7) utilizes three consecutive iteration probabilities for the decision. When all estimated bit nodes satisfy the equation (3.7), then decoding stops. With this method, average number of iterations is reduced by 80% at 3 dB for $\mathcal{P}(2048,1024)$ [87].

Another early detection and termination method is proposed in [21, 25] with the concept of subfactor-graph freezing on BP polar decoder. Subfactor-graph freezing leads to faster convergence by freezing already converged nodes to avoid unnecessary oscillations on variable nodes of the decoder. Decision for freezing is made by observing node messages in every iteration, and frozen nodes are not updated anymore. Same error correction performance is achieved when compared with conventional BP polar decoder with **G**-matrix-based early detection. Even more, average number of iterations are lowered.

A technique similar to the WIB method [84] is mentioned in [27]. As known, a number of information bits are observed to see whether they are converged or not in WIB method. Similarly, a number of frozen bits that have largest capacity (after polarization) among all frozen bits are selected to be observed [27] where if a number of frozen bit places are converged, then decoding is assumed to be successful and decoding operation is terminated. It is shown that the average number of iterations needed is less than when WIB is utilized [27, Fig. 4]. However, **G**-matrix-based method has still the best performance. Another early detection and termination method that uses convergence of the frozen nodes is presented in [123] with name Best Frozen Bits (BFB). If a number of best frozen bits i.e., $N_{\text{BFB}} = 128$ at $\mathcal{P}(1024,512)$ are converged correctly then all information bits are assumed to be converged successfully.

In Table 3.1 the early detection and termination methods are compared in terms of accuracy, complexity and whether a rate loss is valid or not. It can be concluded that despite to its complexity, **G**-matrix-based detection has the best performance. Its complexity, which increases with *N*, is mostly due to the matrix multiplication.

Table 3.1 Performance of early detection and termination methods

Early Detection and Termination Method	Complexity	Accuracy	Rate Loss
G-matrix-based [16]	Highest	High	No
minLLR based [16]	Intermediate	Intermediate	No
LMA [19]	High	Inter.	No
CRC [23,60,85,124,125]	Low	Highest (depends on CRC length)	Yes
WIB [66]	Intermediate	Low	No
OCI [65]	Lowest	Intermediate	No

Perfect knowledge based (PKB) early detection and termination method is used to show a BP decoder's real potential to correct errors [60, 124]. In this method, BP polar decoder stops when estimated data **û** and user data **u** are equal to each other. Similar comparison can be made between estimated codeword **x̂** and **x**. Although, it is not possible to utilize PKB method in any decoder, it is used to demonstrate lower bound of decoder error correction performance [60, 124].

3.3.2 Scheduling

Scheduling is a type of roadmap that follows by BP-based decoder during iterations. Proper scheduling can lead to improved performance of the iterative decoder. Improvement on performance can be measured in terms of BER/BLER performance, complexity of the decoder and throughput. Applying different scheduling techniques has immediate effect on the convergence speed of the BP-based decoder. Scheduling methods are extensively studied for BP-based LDPC decoders. Two-way scheduling (conventional) and flooding scheduling are introduced for the first time in [88] to be applied to the LDPC decoder. There are six scheduling methods defined to be used in

polar decoder, and these methods are two-way, flooding, round-trip, half-way, quarter-way and SCAN scheduling. In this sub-section, some of these scheduling methods are going to be explained for further understanding.

In two-way(conventional) scheduling, message propagation from left to right and right to left occurs at the same time. As known there are $n = \log_2 N$ stages for a polar code with length N. Each stage consists of Z shaped subgraphs. In each subgraph, the messages, i.e., probabilities or likelihoods are updated starting from lower horizontal edge and continued with diagonal edge, and finally upper horizontal edge is updated. A modified scheduling is proposed in [4, 28] that uses a multi-level update process to lead faster convergence. Overall, average iteration number is decreased from 31.7 % to 36.5 % in the range of 2-3.5 dB when compared with conventional BP algorithm.

Unlike conventional parallel BP decoders, SCAN decoders [89] offer sequential decoding like SC, but unlike SC; it uses soft messages for decoding. Thus, SCAN can be regarded as a serial version of BP polar decoder, and can also be considered as another scheduling method [44]. It is shown that, SCAN scheduling can improve the performance and reduce complexity of the design [46, 90, 91]. As with the soft version of SC, SCAN scheduling takes the precedence over flood BP when Bhattacharyya parameter are used to construct the polar codes [2]. However, SCAN timing causes a much larger decoding delay when compared to other scheduling methods.

RT scheduling is recommended in [83] and is studied in [30] to reduce the average number of iterations performed, and to enhance the performance of the BP decoder. In conventional scheduling of BP decoders, messages are calculated from left to right and from right to left at the same time as shown in Fig. 3.15a. In RT scheduling, first messages $L_{i,1}$ to $L_{i,n}$ are calculated, then the messages $R_{i,2}$ to $R_{i,n+1}$ are calculated as illustrated in Fig. 3.15b.

As it is seen from Fig. 3.15, the number of steps needed to perform RT scheduling is two times greater than the conventional scheduling. By utilizing RT scheduling on BP polar decoder, the average number of iterations is reduced when compared to the conventional BP decoder [83]. The same error correction performance is observed with respect to SMS

BP [30]. There are other studies that uses RT scheduling [27, 90, 92] which show that it is a proper method to use.

Steps	1	2	...	n	n+1	n+2	...	2n	...
Message Update	$R_{i,2}$	$R_{i,3}$	...	$R_{i,n+1}$	$R_{i,2}$	$R_{i,3}$	...	$R_{i,n+1}$	...
	$L_{i,1}$	$L_{i,2}$	...	$L_{i,n}$	$L_{i,1}$	$L_{i,2}$	...	$L_{i,n}$	...

← 1st iteration → ← 2nd iteration → ← · — · ·

(a)

Steps	1	2	...	n	n+1	n+2	...	2n	...
Message Update	$L_{i,1}$	$L_{i,2}$	...	$L_{i,n}$	$R_{i,2}$	$R_{i,3}$	...	$R_{i,n+1}$	...

← 1st iteration → ← · — · ·

(b)
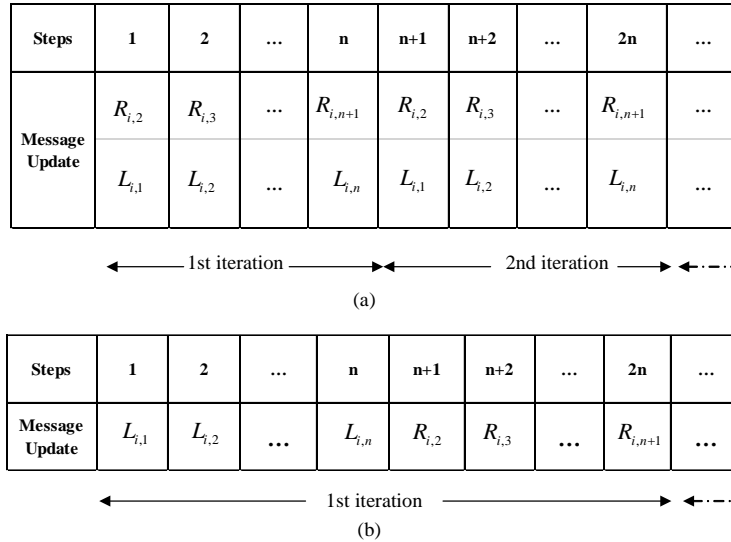
Figure 3.15 (a) Two-way (conventional) scheduling (b) round-trip scheduling.

Other two well-known scheduling methods are the half-way and quarter-way scheduling methods. In half-way scheduling, information flow starts at the same time from leftmost and rightmost part of the factor graph. Propagating messages are exchanged in the middle of the BP factor graph, and flow direction is reversed. Information flow for both of the half-way and quarter-way scheduling is depicted in Fig-3.16. Quarter-way scheduling is nothing but divided version of half-way scheduling. Both methods can be considered as a modified version of RT scheduling. The aim of both methods is to decrease the latency of BP decoder and to lead to faster convergence.
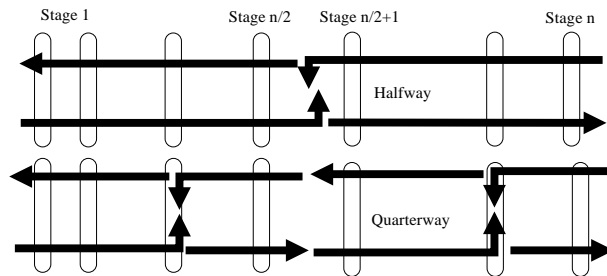
Figure 3.16 Half-way and quarter-way scheduling

In most of the scheduling algorithms, one or two stages of the decoder are utilized at the same time while PEs on other stages stay idle. In other words, conventional BP decoder activates PEs stage-by-stage from left to right and right to left in each iteration. In order to achieve 100% utilization, folding approach is presented in [93]. n-level folded decoding scheme is presented in Fig. 3.17 where $K_{1,(1...n)}^{1,...,n}$ stands for decoding stage and variable node.

Clock Cycle	1	2	3	4	5	6	7	8	9	10	11	12
Stage 1	$K_{1,1}^1$	$K_{1,2}^1$	$K_{1,3}^1$	$K_{1,4}^1$	$K_{1,1}^2$	$K_{1,2}^2$	$K_{1,3}^2$	$K_{1,4}^2$	$K_{1,1}^3$	$K_{1,2}^3$	$K_{1,3}^3$	$K_{1,4}^3$

Figure 3.17 4-level folded decoding scheme for N=16.

When all four stages of polar code with length N=16 are folded, decoding can be completed in 12 clock cycles using one stage of PEs. 100% utilization of PEs is achieved, but latency is increased. Folding scheduling can be adjusted by adding some stages to the decoder in order to lower latency. For an example, Fig. 3.18 demonstrates 2-level folded scheme for N=16.

Clock Cycle	1	2	3	4	5	6	7	8
Stage 1	$K_{1,1}^1$	$K_{1,2}^1$	$K_{1,1}^2$	$K_{1,2}^2$	$K_{1,1}^3$	$K_{1,2}^3$		
Stage 2			$K_{1,3}^1$	$K_{1,4}^1$	$K_{1,3}^2$	$K_{1,4}^2$	$K_{1,3}^3$	$K_{1,4}^3$

Figure 3.18 2-level folded decoding scheme for N=16.

By using folding approach, total logic gate number to implement BP decoder and latency of the decoder can be decreased significantly [93, Table I].

SCAN scheduling is proposed in [89] as a soft version of SC algorithm [2]. As known, bit by bit decoding of the information bits takes place in a SC based decoder. In other words, to decode m^{th} bit, all the bits from 1 to $m^{th}-1$ must be decoded. Despite to the increment on latency, SCAN scheduling achieves lower BER. $N\log_2 N$ cycles are needed for SC/SCAN algorithms do decode the polar code $P(N,K)$. However, $M\log_2 N$ cycles are

needed to decode $P(N,K)$ with BP-based polar decoder where M is the iteration number which is usually 10-20 times smaller than N. However, SCAN algorithm is still a strong candidate among soft decision based polar decoders to achieve high BER/BLER performance when throughput is the second importance criteria.

When all the decoders in this sub-section are inspected, it is seen that scheduling method has significant effect on decoders' BER/BLER and throughput performances. RT scheduling shows the best results while folded decoding scheme offers a trade margin between throughput and complexity to the developers.

3.4 Polar Code Construction Methods

The capacity of the split channels presented in [2] are calculated using the Bhattacharyya parameter for SC decoding. As for the BP decoder, there is no proposal in the literature to calculate the capacities of split channels. The need to use different methods for the calculation of channel capacities arises from the fact that recursive prediction approaches based on the Bhattacharyya boundaries, Gaussian approach (GA) and density evolution (DE) are appropriate for the SC decoding scheme. There is still no definite method for the selection of frozen bits for continuous channels like AWGN, Rayleigh fading channels etc. It is also important to note that computational-based selection algorithms such as GA and DE show similar performance compared to Arıkan's recursive prediction approach based on Bhattacharyya parameter. In the remainder of this section, Monte Carlo simulation (MC) based polar code construction methods that provide better performance for BP-based decoding will be discussed.

As mentioned earlier, a stopping set consists of a set of variable nodes, and each neighboring check node is associated with at least two variable nodes. A stopping tree can be considered as a subset of stopping set. A sample stopping tree for variable node $v(1,6)$ is shown in bold lines in Fig. 3.19.

Leaf size is related to the stopping tree concept, and it is a significant parameter for the decoder. To find leaf size of a variable node, tree structure of the decoder can be followed. For example, leaf size of variable node $v(1,6)$ presented in Fig. 3.19 is four. Leaf set size based frozen bit selection is proposed in [39] where information bit channels whose leaf set sizes are smaller than 2^8 are taken as frozen bit channels for $N = 2^{13}$. Also, previously

selected frozen bit channels with leaf sizes greater than 2^8 are converted to the information bit channels. By using leaf set size based code construction method [39], BP polar decoder outperforms the BP polar decoder constructed using Bhattacharyya parameter. Leaf set size based construction is used by Reed-Muller codes, which can be seen as an inspiration for the idea.
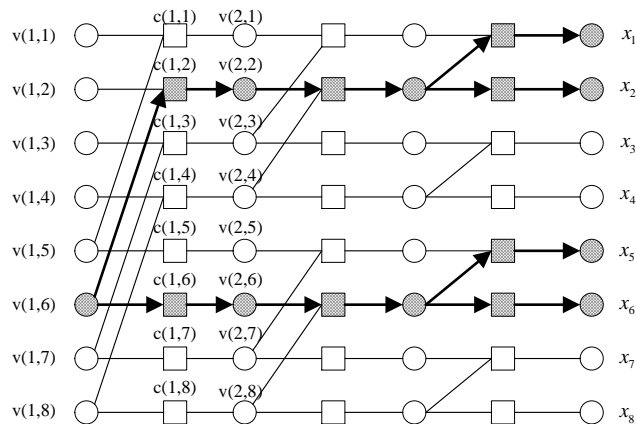
Figure 3.19 Stopping tree for the variable node $v(1,6)$

To improve the error correction capabilities of BP decoding schemes, a method for selecting frozen bits via MC tests is proposed in [90, 94, 95, 96]. In these studies, better performance is obtained for decoders that use BP decoding. In [95], the selection of frozen bits according to MC trials is utilized for data lengths 128 and 256. Due to its complexity, the structure based on MCs trials is not offered for large block lengths.

Alternatively, a scheduling-adapted method for generating polar code is presented in [90]. RT scheduling in the factor graph of a BP decoder has its own characteristics. The authors use two different approaches to generate the polar code. In the first one, MC trials are performed using BP decoding with an RT schedule. Since the MC approach takes time, an alternative algorithm is proposed to generate the polar code. In the second algorithm, a jump-start MC based polar code construction is proposed. In this method, an initial index setting is set using RM(3,8) which provides an index set corresponding to the indices of the rows of **G** of the largest Hamming weight. After initial index is chosen, MC trials start. Overall, better error correction performance for the polar decoder with RT scheduling at $E_b / N_0 = 3$ dB is achieved over GA based polar code construction. Besides, authors show

that there might be some systematic methods to optimize iterative polar decoder design under a specific scheduling method [90].

It is shown in [90, 95] that MC trial based polar code construction requires huge amount of time for block lengths greater than $N = 256$. FPGA is used to lower time consumption by accelerating MC trials in [96] where it is reported that three month length simulation time is reduced to 6.3 hours for $N = 1024$ and three different MC trial based approach is presented, and these approaches can be briefly explained as:

- In One-Time Rank and Freeze Method:
    1. Set all bits as information bits.
    2. Run MC on BP decoding simulations and measure the error rate of each bit.
    3. Rank the bits based on error rate and freeze the $N - K$ least reliable bits to obtain the bit selection for a $\mathcal{P}(N,K)$ polar code.
- In Iterative Rank and Freeze Method:
    1. Set all bits as information bits.
    2. For $i = 1$ to N_{it} where N_{it} is maximum iteration number of BP decoder:
        a. Calculate error rate for each non-frozen bit after MC simulations.
        b. Sort non-frozen bits according to step 2.**a** calculations, and freeze M_i least reliable one.
- In-order Bit Selection Algorithm:
    1. For $i = 1$ to $N - 1$
        a. When $i = 1$, set all bits as information.
        b. When $i > 1$, freeze u_0^{i-1} and set bits u_i^{N-1} as non-frozen.
        c. Calculate error rate of u_i after MC simulations.
    2. Sort the bits according to error rate and freeze $N - K$ least reliable bits.

Polar code constructed with "In-order bit selection algorithm" is shown to outperform the polar code that is constructed with DE method when BP-based decoding is applied.

Another MC based frozen bit selection method is presented in [24] which is explained as follows:

1. Set code rate to $R = 1/N$, i.e., one information bit for N different codes.

2. Perform simulations and observe the error correction performance of N different codes.

3. Record and sort error rates.

4. Choose the best one as information bit.

5. Increase the number of information bits and return to step 2.

6. Finish simulation when $N - K$ information bits are selected.

MC based frozen selection method presented in [24] is performed for the code $\mathcal{P}(243,121)$. As mentioned previously, MC-based methods are applicable to short blocks if the delay reduction algorithm is not utilized. Therefore, to achieve the best performance during BP-based decoding, a more efficient polar code generation method need to be developed.

A new polar code construction method, LLR-based bit-swapping is presented in [91] as an alternative to MC trial based code construction. In this approach, final log-likelihood ratio (LLR) of each channel after each iteration is observed, and twelve best-converged ones are chosen to be used as information bits. Similarly, twelve worst-converged ones are chosen to be used as frozen bits. The remaining frozen and information bits are determined using Bhattacharyya parameter based selection method. With this approach it is seen that error correction performance of the code improves when compared with conventional BP, BP with SCAN scheduling and SCL decoding [91]. To sum up, MC trial based frozen bit selection for different decoder types is still an attractive topic that draws researcher's interest.

RM and polar code construction relies on the matrix $\mathbf{F}^{\otimes n}$ where construction of $\mathbf{F}^{\otimes n}$ is mentioned in the chapter 2. The difference between them lies in the selection of channels carrying information bits. The construction of the polar code is based on the calculation of the Bhattacharyya parameter to minimize the error probabilities for SC decoding. This is not an appropriate method for the selection of information bits for BP decoding. Bearing this in mind and knowing that the structure of RM is based on the greatest minimum Hamming distance, it is shown in [97] that for finite length codes, the Hamming distance is more important than the polarizing effect. Thus, a method for generating a hybrid RM-polar code is provided. This method can be applied in three stages:

a) The index of the frozen bits is decided considering the Hamming weights of the rows of **G** matrix having Hamming weight less than a definite threshold d.

b) $N - K$ number of information bits are chosen according to smallest Bhattacharyya parameter from the remaining set.

c) Finally, remaining set of bits are chosen as frozen.

[62, Fig. 4] shows that a hybrid RM-polar code provides slightly better performance than the Bhattacharyya parameter based code under BPL and SCL decoding. Hybrid RM-polar code yields better error correction performance due to increased minimum distance. However, they do not show the best result in the literature, since the performance of the linear code in the iterative decoding process is determined by the stop sets on the Tanner graph of the code [98], not just by the minimum distance of the code.

Deep learning approach is also used to construct polar codes for any $\mathcal{P}(N,K)$ [71]. It is shown that the polar codes constructed using the deep learning method show similar performance to the polar codes employing Bhattacharyya parameter based polar code construction for $\mathcal{P}(1024,512)$ in AWGN channel. Nevertheless, deep learning based estimation is a promising candidate.

One of the best methods for the construction of a polar code for the BP polar decoder is proposed in [99], where a genetic algorithm is used to for the code construction. The polar code developed using the genetic algorithm considers the structure of the stopping sets in the Tanner graph of the BP decoder. BP polar decoder whose frozen bit locations are selected using genetic algorithm achieves SCL decoder's error correction performance on AWGN channel for $\mathcal{P}(2048,1024)$. As we are going to mention in the next chapter of this thesis, polar codes constructed using the genetic algorithm have the best performance results when compared with the other polar.

3.5 Errors Types of BP Polar Decoders

It is important to understand why bit errors occur in a BP polar decoder. The structure of the encoding, decoding algorithm/scheduling, and SNR of the communication channel are important factors that affect the BER performance of any forward error correction code. In the field of BP-based polar decoder, literature focuses on changing, upgrading the

decoder structure, and the decoding schedule to avoid errors. Error sources, error types, and methods for resolving various error types are described in this section.

Stopping sets are introduced with the Tanner graph representation of LDPC code for first the first time. A stopping set, **S**, is a subset of **V**, the set of variable nodes such that all neighbors of **S** are connected to **S** at least twice [100]. Fig. 3.20 depicts a sample of stopping set in a polar code. It is shown in [75] that smaller stopping sets play an important role in the decoding process and cause error floor. In other words, stopping sets are strongly related with the probability of error such that when nodes in a stopping set are erased or decoded wrongly, then P_e increases. It can be concluded that larger stopping sets yield lower BER values in Tanner graph based decoders.
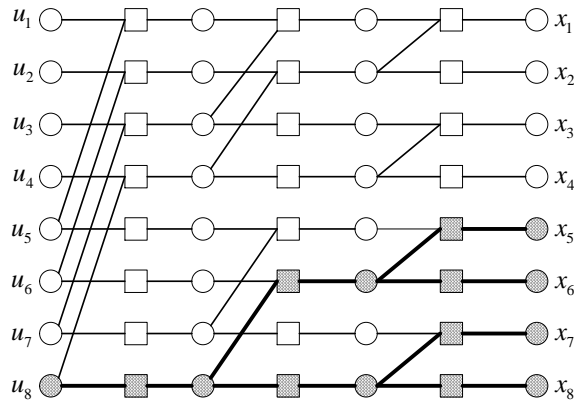
Figure 3.20 A sample stopping set on a polar code Tanner graph

Smaller stop sets usually cause error floor problem especially in LDPC and turbo decoders. When the SNR increases, the BER decreases exponentially, but if the decoder encounters the error floor problem [38, 59, 101, 102], then the BER remains unchanged or decreases at a low speed. Thanks to the size of the stop set, it is shown in [101] that the polar codes show excellent minimum error floor performance such that error floor is not observed even at a BER value of 10^{-9}. In [39, 59] it has been shown that the fixed bit selection and the size of the stop set are closely related. A modified algorithm for selecting frozen bits is developed by increasing the stopping distance of the polar code in [39].

Girth of a factor graph is another source of error floor [38, 39]. Girth is defined as the shortest cycles on the factor graph. Shortest cycles prevent BP decoders to reach

convergence state. Thus, shortest cycles, i.e. paths are considered as deficiency of the BP-based decoders.

The aim of using logarithmic versions of the propagating message equations in BP-based decoders is to lower the complexity of the decoders. In logarithmic domain, multiplication operations are converted to additions and divisions are converted to the subtractions. Word lengths (bit length) of the LLR values are also decisive on the accuracy of the decision made by decoder. In some cases, bit length of the messages need to be clipped to keep the overall complexity low. However, clipping of log-likelihood values can cause error floor problem [59]. Clipping into shorter lengths is shown to lead to error floor in [59, Fig. 3]. Therefore, four different algorithms are presented to avoid error floor caused by clipping operation, and these algorithms can be outlined as:

a. Guessing algorithm is based on finding oscillating LLR values. After finding an indication of oscillation, a threshold is assigned to follow and lead to correct convergence.

b. Adding virtually generated noise to the input received from the channel can lead to correct convergence.

c. A scaling constant (α i.e., $0 < \alpha < 1$) is injected to the PE of the BP polar decoder. Changing this constant can avoid saturation that is known to be the reason of error floor.

d. Multi-trellis BP decoder is proposed for the first time in [4] where it is stated that when conventional factor graph fails to converge, then a differently oriented trellis can be utilized. Multi-trellis BP decoder is a promising candidate to improve the BER performance significantly.

Although mentioned methods avoid error floor caused by low clipping value [59], complexity increment should be taken into consideration when these methods are to be used in practical systems.

On the other hand, the BP decoding scheme is a promising method with a parallel structure that provides higher throughput and lower latency. It is also important to understand how the BP decoder fails. An error classification is made in [102] such that BP decoder can handle them intuitively, and three error types are introduced. These error types are:

a. Unconverged errors: If the decoder fails to resolve the error and other iterations do not work, this error is called an unconverged error. This type does not have a specific error pattern.

b. Converged errors: Converged errors occur if convergence results to the wrong codeword or BP decoder stays in local minima such that hard decisions coincide with repeated repetitions. Such errors cannot be corrected using further repetitions [102].

c. Oscillation errors: The BP decoding scheme suffers from the propagation of wrong messages due to the loop structure. If the solutions of these incorrect messages change periodically, then this error type is defined as oscillation error.

By following hard decisions in sequential iterations, unconverged and oscillation errors can be detected, while errors that are falsely converged cannot be detected. At different SNR levels, error distributions show difference. According to [102], unconverged errors are dominant at low SNR, converged and oscillation errors become dominant as SNR increases. When CRC is used in the BP decoding process, it appears that some of the unconverged and oscillation errors have been successfully resolved, but converged errors are still not discussed.

In the study [90], the authors use [102]'s error type definitions and investigate the error distributions under different scheduling (RT and SCAN) and polar code construction methods such as GA and MC. Simulation results show that similar results are obtained under RT and SCAN scheduling. It is also seen that most decoding errors are caused by unconverged errors in low SNRs, and as the SNR increases, converged errors become dominant. However, unconverged errors are less effective when RT scheduling is used.

Alternatively, three separate post-processing (PP) methods are provided to avoid any type of errors in [103] where it is shown that the BP decoder employing PP methods achieves better BER performance than the SC decoder [103, Fig. 12].

3.6. Adaptive Strategies in Polar Decoders

The need for adaptive strategies in any coding scheme results from the desire to achieve better error correction performance. Therefore, a wide variety of adaptive strategies has been developed. In any decoder, the SNR of any channel is an important factor in the BER

/ BLER performance and bandwidth of the decoder. Adaptive SNRs based BP polar decoders are being investigated to further performance improvement.

An adaptive early detection and termination method has been proposed for the minimum LLR-based detection approach in [20]. Different thresholds are applied to possible SNRs to determine the optimal convergence condition. Another early detection method adapted to alter SNR is WIB [84]. With adaptive WIB, the average iteration is reduced compared to a fixed WIB.

In [31], a method of channel estimation based on the creation of a motivated parity check matrix is presented with the fact that adaptive LLR values are closely related to channel SNR. In general, parity check matrix-based decoding with adaptive approach shows better BER performance than SCL decoder.

Adaptive quantization is studied in [86] to reduce the complexity of the polar BP decoder. Simulation results show that accurate precision is required in low SNRs, while coarse precision is sufficient in high SNRs.

In the meantime, the adaptive design of the polar code has been developed for the "In-order bit selection method" [96] as presented in sub-section 3.4. Frozen bits are selected for five SNR values ranging from 1 dB to 5 dB in the AWGN channel. Simulation results show that in high SNRs, polar codes generated with 4 dB and 5 dB give better BLER performance, but there is no significant difference in BER characteristics. Channel SNR adapted strategies for a BP-based polar decoder have been classified according to performance improvement and complexity reduction. In addition, the applied scheduling technique is the subject of adaptive research.

In SNR channel estimation process adaptive methods are vital to use. In [20], a novel algorithm is proposed that is based on Hamming distance between $\mathbf{\hat{u}G}$ and $\mathbf{\hat{x}}$. A new parameter λ is introduced and the new parameter λ equals to 0 when $\mathbf{\hat{x}} = \mathbf{\hat{u}G}$. As expected, λ takes large values at low SNR values, and it takes small values at high SNR region. As a result, the relationship between λ and channel SNR is utilized for the presented adaptive method.

3.7. Hardware Implementation

Efficient hardware implementation of BP-based polar decoders is an important issue to prove its superiority to SC and SCL structures. Implementations are accomplished on three different hardware platforms: FPGA, GPU and Application Specific Integrated Circuit (ASIC) applications. BP-based algorithms can be processed in a parallel manner in these platforms.

The first hardware implementation of the BP polar decoder is proposed in [104] using an FPGA device. Typically, an FPGA consists of a matrix of configurable logic blocks connected by programmable interconnects and is mainly used to process parallel operations. This is one of the best candidates for hardware decoding to perform BP decoding. The performance of a BP decoder is compared to the convolution turbo code (CTC) used in the IEEE 802.16 wireless broadband standard. It can be seen that the BP decoder is better in complexity and bandwidth, but not better than CTC in BER performance. Additionally, the designed BP decoder achieves 27.83 Mbps for polar code $\mathcal{P}(1024,512)$ [104].

Based on the fact that there are similarities between the polar BP decoder and the fast Fourier transform, a fully parallel BP decoder is provided, including a pipeline BP decoder and feedback and pipeline BP decoder in [105]. Most pipeline architectures are implemented in FPGAs and a desired trade-off between performance, efficiency, latency, and coverage of the decoder is achieved.

A modified implementation of the BP decoder yielding a throughput of 9.45 Gbit/s is performed on the FPGA platform in [26] where an early stopping criteria is used using a high-speed Ling adder with a parallel prefix, and a simplified processing element.

Another hardware platform for implementation of BP decoders is the graphics processor especially designed for fast mathematical calculations for rendering. It is capable of performing parallel calculations that are very important for BP decoder. A GPU consists of a series of processors that can perform calculations in parallel. The first example of a GPU-based polar BP decoder is presented in [66] for $\mathcal{P}(1024,512)$ and a throughput of 3.55 Mbit/s is reported. A higher 34 Mbit/s throughput is observed in [53] where the

hybrid structure of BP-SCL was applied to the GPU using NVIDIA CUDA C programming.

In [122], it is shown that 1 Gbit/s throughput for 5 dB with polar code of length $N \leq 1024$ in AWGN channels can be achieved using a device (GPU) with a graphics processor. In [59], the authors examine the effect of clipping on error floor when decoding BP polar codes on the GPU platform. ASIC-based applications provide better performance. ASICs are configured to meet the requirements of a parallel BP decoder structure and are presented in various scheduling methods [20, 22, 25, 83, 93]. Table 3.2 shows the outcomes of the ASIC applications in terms of frequency, area, average number of iterations, latency, energy per-bit and average efficiency. Further comparisons with other decoding schemes (SC, SCL) and other hardware platforms are also examined in [107]. *finFet* technology has emerged as an alternative to CMOS devices. *finFet*-based designs can lead to low power consumption and decoding delays. Therefore, *finFet* technologies and near-threshold calculations are used to obtain high-speed, low-power BP polar decoders [108]. With *finFet* technology, the critical path delay can be reduced to 110 ps. For comparison, the critical path delay at 45 nm TMSC is 1050 ps.

Table 3.2 ASIC implementation results of BP decoder for $P(1024,512)$

Reference Design	[16]	[65]	[18]	[21]
Architecture	Overlapped	Double Column	Stage Combined	Subfactor-Graph Freezing
Schedule	5-stage folded	Round-trip	Round-trip	Quarter-way
Technology	45nm	65 nm	45nm	65 nm
Frequency (MHz)	500	300	197	334
Area	N/A	1,476	0,747	1,6
Avg. number of iterations	23	6,57	N/A	6,34
Latency	56	65,7	N/A	31,7
Energy per bit	220	102	N/A	40
Avg. throughput (Gbps)	4,5	4,67	1,683	10,7

CHAPTER 4

NOISE AIDED BELIEF PROPAGATION
LIST DECODING OF POLAR CODES

Noise in communication systems is generally considered as a negative phenomenon and countermeasures are taken to eliminate it. On the other hand, the concept of stochastic perturbation opens a new perspective that would benefit from the artificially generated noise into a nonlinear system. The addition of white noise to a nonlinear system to obtain a stable model was firstly studied in [110]. Generally, a stochastic perturbation system has three elements, which are a weak input signal, a noise signal, and a bistable nonlinear system. Since its introduction, noise is used as a useful phenomenon in detecting a weak electrical signal using stochastic distortion, i.e. stochastic resonance, in EMG, EEG based measurements, in signal detection on nano-scale devices like nano-wire FETs and CNT-FETs, in object detection in images, and in even for the explanation of neural transmission of human brain. Thus, stochastic perturbation is not a new idea as a useful phenomenon. However, it is new for forward error correction.

Polar codes, which are the first mathematically proven error correcting codes achieving Shannon's capacity, are introduced by Arıkan in [2]. After its introduction, polar codes are adopted to be used in uplink/downlink control channels of 5G framework of eMBB introduced by 3GPP group [3]. There are different decoding schemes for polar codes. Arıkan in [2] introduces the first decoding algorithm, SC decoding algorithm, for general channels. In addition to SC decoding algorithm, other decoding methods such as CRC-aided SC List [10] decoding, belief propagation (BP) based decoding [7], SC stack (SCS) decoding [111], and linear programming [8] based decoding of polar codes are proposed. It is shown that polar codes utilizing CRC aided SCL decoding outperforms maximum likelihood bound of polar code with a large list size and is able to compete LDPC codes

[10]. However, SC and SCL decoders suffer from their serial decoding nature. In other words, to decode m^{th} bit, all bits indexed from 1 to $m-1$ must be decoded to successfully accomplish the decoding of current bit d_m where $1 \leq m \leq N$ bit. To overcome this drawback and giving another perspective to researchers, BP algorithm based polar decoder is proposed by Arıkan [7]. The basic feature of the BP polar decoder is its ability to decode bits in parallel. Although BP decoder have higher complexity than SC based decoders (SCL, SCS), its throughput achievement is remarkable when compared with the state-of-the-art SC based decoders.

Another advantage of BP polar decoder is its soft information flow. Soft information flow at each step of the BP decoding allows us to use it jointly with other well-known code schemes like LPDC and Reed-Solomon codes. List decoding concept is also used in BP-based decoding of polar codes [62] as in SC decoder [10], but in a different manner. In SC list decoding, the list size is doubled at each decoding stage until the maximum list size. On the other hand, L number of BP polar decoders for L parallel branches are utilized by a BPL decoder [62]. In this thesis, we propose a BPL decoder having independent BP decoders enhanced by the virtually generated noise intensities at each parallel branch, such that it is named as **Noise-aided Belief Propagation List polar decoder**. Up to know, we tried to enlighten the path and motivation for the Na-BPL polar decoder. In section 4.1 the parameters we prefer to use for the BP polar decoder are explained by making some analyzes. In the section, 4.2, two different polar code construction methodologies are presented to enhance the BP decoder's performance. In section 4.3, our Na-BPL decoder is introduced, and its maximum error correction capacity is demonstrated. Finally, in section 4.4, Na-BPL decoder with practical and realistic parameters are studied and a comparison with state of art studies is made.

4.1 Constructing a BP Polar Decoder

Suggestions on error correction improvement for BP polar decoders are briefly explained in chapter 3. Besides, to improve its performance, countermeasures are taken into consideration in terms of adding extra check nodes, chosen scheduling type, chosen early detection and termination method, and chosen polar code construction method etc. We have used some of the methods presented in the literature survey, i.e., chapter 3. Among

these methods, we have chosen the most feasible and accurate methods to improve the performance of BP polar decoder. In this sub-section, we are going to explain the methods selected and explain the reason why we chose them.

4.1.1 Adding Frozen Check Nodes

BP polar decoders as presented in the previous chapters, have iterative processing nature. At each iteration, it is expected that the decoder's likelihood values converge to reliable values for better estimation. Thus, the nodes with known likelihoods should be bypassed for faster convergence. In polar codes, some of the nodes are frozen nodes, and the outputs of these nodes are known and they are usually zeros. If the likelihood values of the frozen nodes are calculated during the iterations, they can be badly affected due to the wrongly converged nodes, and the likelihood values of the frozen nodes may oscillate. Despite its oscillation, frozen nodes can be correctly decoded at the end of the decoding. However, if we do not evaluate the likelihood values of the frozen nodes, and accept them as constant values during iterations, decoder's convergence is achieved with less iterations. To show this, we performed the simulations using BPSK modulation and AWGN channel, and the results are given in Fig. 4.1, Fig. 4.2 and Fig. 4.3.

Figure 4.1 BER/BLER comparison between original BP polar decoder and BP polar decoder with frozen check nodes for $\mathcal{P}(512,256)$
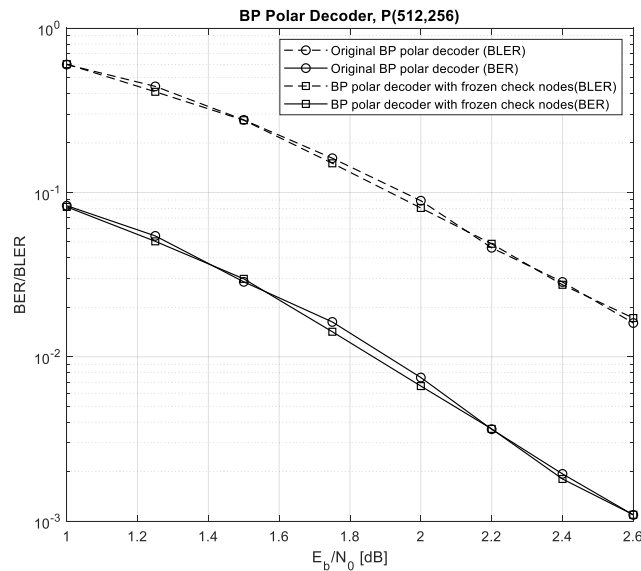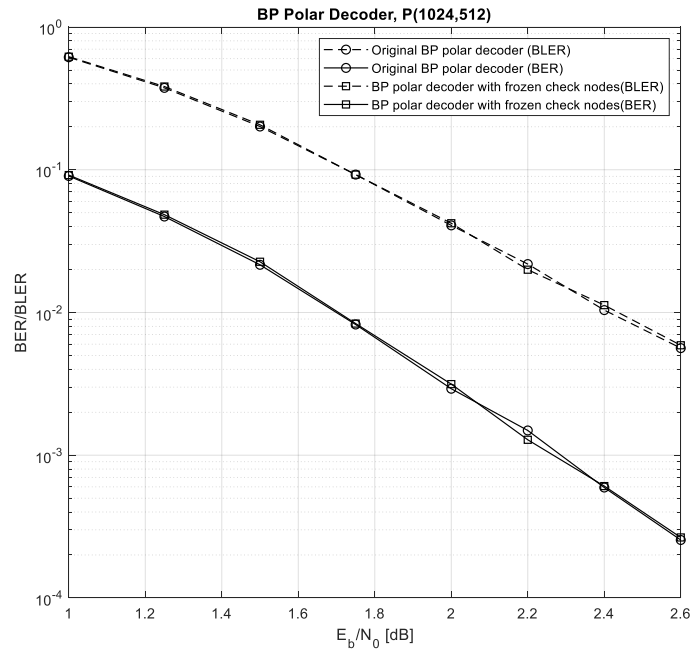
Figure 4.2 BER/BLER comparison between original BP polar decoder and BP polar decoder with frozen check nodes for $\mathcal{P}(1024,512)$
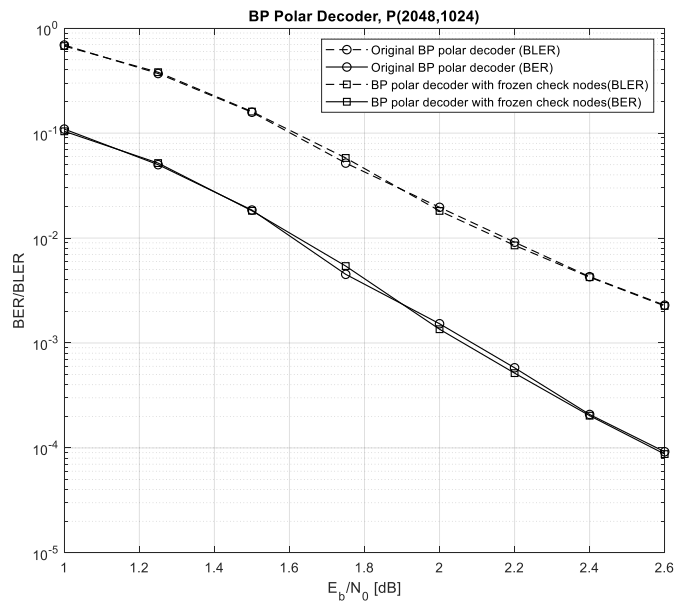

Figure 4.3 BER/BLER comparison between original BP polar decoder and BP polar decoder with frozen check nodes for $\mathcal{P}(2048,1024)$

As it is seen from the Fig. 4.1, Fig. 4.2 and Fig. 4.3 that the use of frozen check nodes with fixed likelihoods does not affect BER or BLER performance of the decoder. In Table 4.1, the average number of iterations performed for different code lengths are given. It is obvious from Table 4.1 that, the use of frozen check nodes with fixed likelihoods leads to faster convergence. After this point, all BP polar decoders utilized in the thesis will have frozen check nodes.

Table 4.1 Average number of iterations comparison for different code lengths

Polar Code/ SNR(dB)	1	1.25	1.5	1.75	2	2.2	2.4	2.6
Original $\mathcal{P}$ (512,256)	40.3	35.0	30.3	24.4	21.08	19.0	16.8	15.4
$\mathcal{P}$ (512,256) with frozen check nodes	35.8	28.7	21.8	16.0	11.8	9.3	7.7	6.6
Original $\mathcal{P}$ (1024,512)	42.7	37.0	30.8	25.6	21.7	19.5	17.8	16.0
$\mathcal{P}$ (1024,512) with frozen check nodes	38.5	29.6	21.6	15.6	11.4	9.3	7.9	6.98
Original $\mathcal{P}$ (2048,1024)	45.1	38.4	31.1	25.7	22.1	20.1	18.4	17.1
$\mathcal{P}$ (2048,1024) with frozen check nodes	42.7	31.6	21.8	14.9	11.3	9.75	8.5	7.6

4.1.2 Scheduling

Updating strategy of variable and check nodes on an iterative BP decoder affects the decoding speed. Thus, choosing the best scheduling method for our decoder is a prominent issue. In the previous chapter, different scheduling techniques are explained briefly as conventional scheduling, half-way scheduling, quarter-way scheduling, SCAN scheduling and round-trip scheduling. In this sub-section, after briefly mentioning scheduling techniques, comparison in terms of error correction performance and issued average number of iterations will be done. Meanwhile, Fig. 4.4 shows the message propagation strategies of different scheduling methods.

We made a comparison of the scheduling methods in terms of error correction performance i.e. BER and BLER. Fig. 4.5, Fig. 4.6 and Fig. 4.7 depict the simulation

results obtained using BPSK modulated signal and AWGN channel for $N = 512, N = 1024$, and $N = 2048$.

(a) conventional scheduling

(b) half-way scheduling

(c) round-trip scheduling
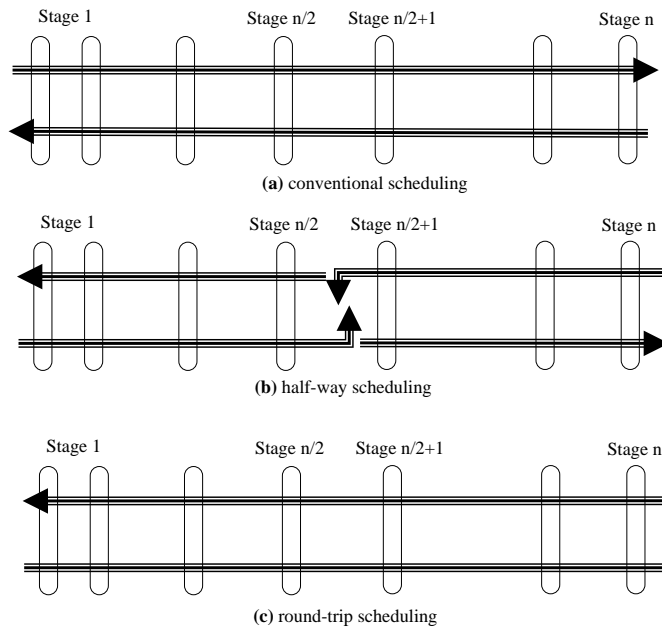
Figure 4.4 Scheduling strategies for BP polar decoder; (a) conventional scheduling, (b) half-way scheduling and (c) round-trip scheduling
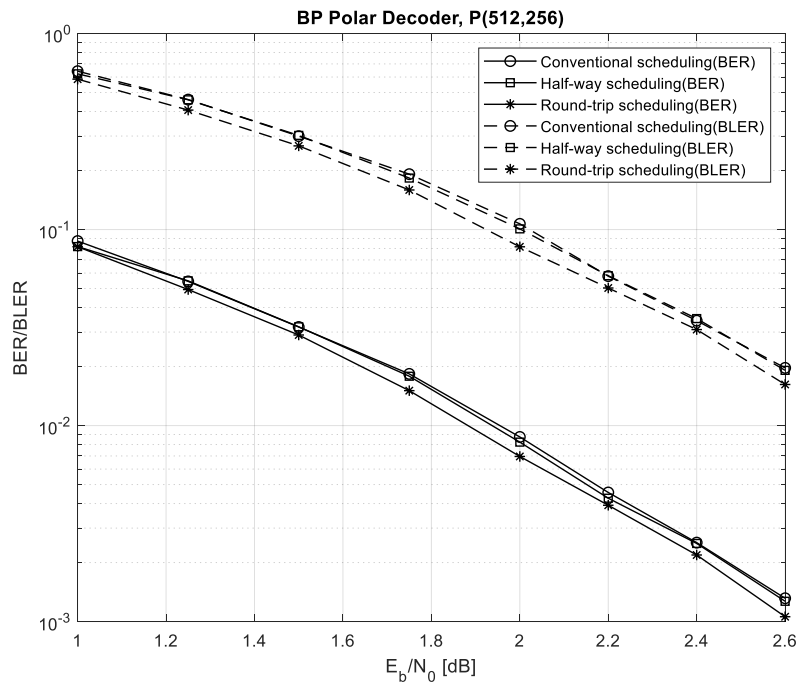
Figure 4.5 Scheduling strategy comparison for $\mathcal{P}(512,256)$ in terms of BER and BLER
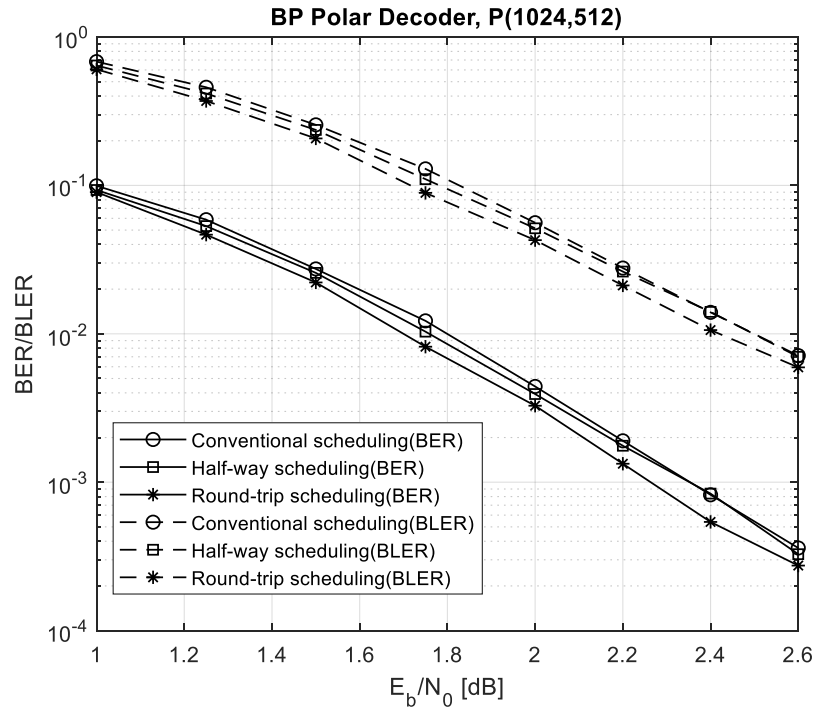
Figure 4.6 Scheduling strategy comparison for $P(1024,512)$ in terms of BER and BLER
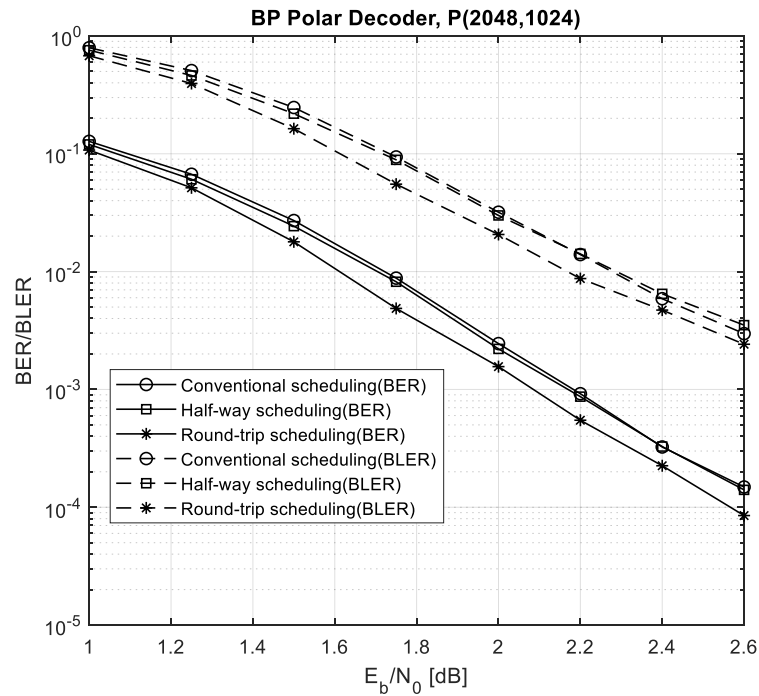
Figure 4.7 Scheduling strategy comparison for $P(2048,1024)$ in terms of BER and BLER

Simulation results show that round-trip scheduling has the best error correction performance. On the other hand, convergence speed of the decoder is an important property that is inversely proportional with the latency of the decoder. We also observed the average iteration number for each scheduling strategy during simulations. Besides, we employed **G**-matrix-based early detection and termination method for the BP polar decoders. Fig. 4.8, Fig. 4.9 and Fig. 4.10 show the average number of iterations performed for each scheduling method for different code lengths when maximum number of iteration, *M* is set as 50.
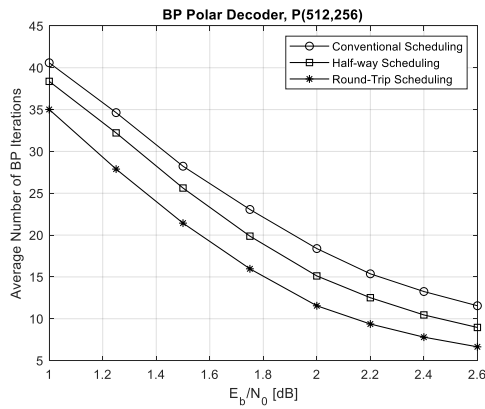
Figure 4.8 Scheduling strategy comparison in terms of average number of iterations for $\mathcal{P}(512,256)$
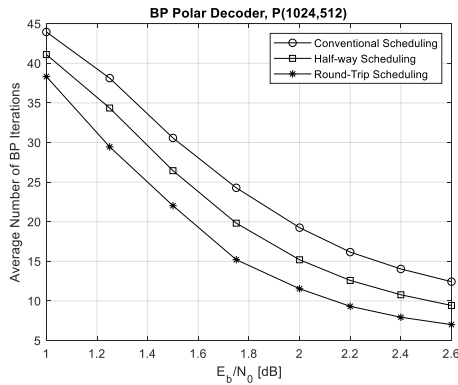
Figure 4.9 Scheduling strategy comparison in terms of average number of iterations for $\mathcal{P}(1024,512)$
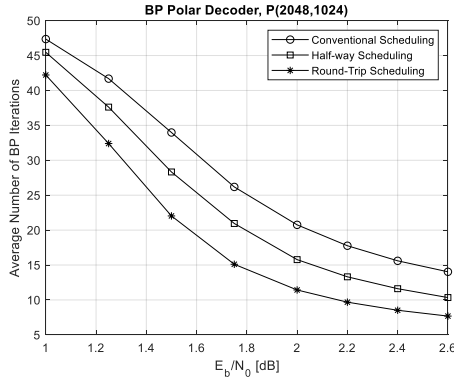
Figure 4.10 Scheduling strategy comparison in terms of average number of iterations for $\mathcal{P}(2048,1024)$

It can be inspected from the Fig. 4.4 that RT scheduling needs $2n$ cycles for an iteration while conventional and half-way scheduling need n cycles. Thus, comparing RT scheduling with others having the same maximum iteration number is not fair. Therefore, for a meaningful comparison on $\mathcal{P}(2048,1024)$; we set $M = 50$ for RT scheduling (equivalent of $M = 100$) and set $M = 100$ for other two scheduling types. As a result, Fig. 4.11 shows that BER performance of all three scheduling types are the same, and half-way scheduling has the lowest average iteration number. It can be concluded that using half-way scheduling can be advantageous if throughput of the decoder is a critical factor for communication systems. On the other hand, the use of RT scheduling can be the preferred choice if the computational complexity of the decoder is the critical issue. In half-way scheduling, two stages must be performed at the same time, while in RT scheduling, only one stage is active at each step of the decoding. In this thesis study, all three scheduling strategies are followed.
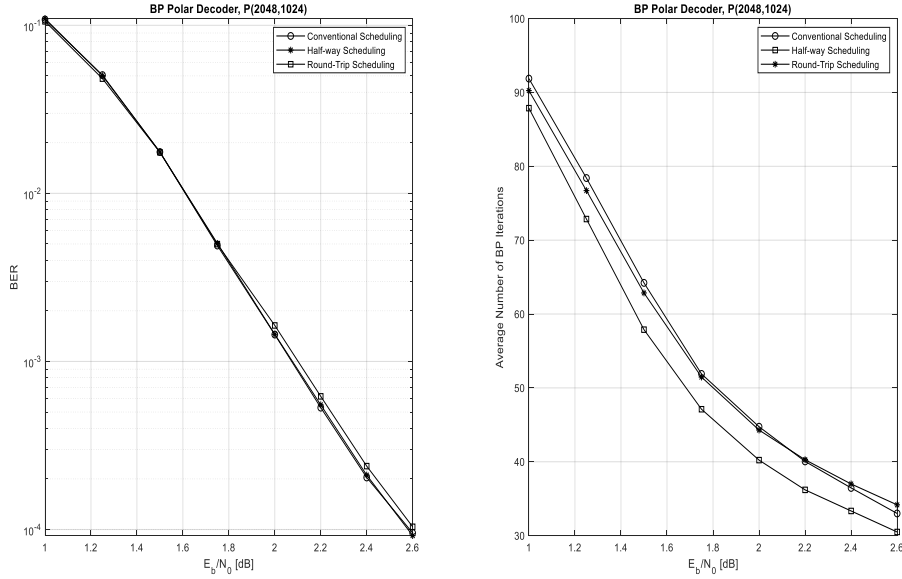
Figure 4.11 Scheduling strategy comparison for $\mathcal{P}(2048,1024)$ the when number of maximum iterations are kept same for all methods

4.1.3 Early Detection and Termination Method

BP-based polar decoding operation is assumed to be successful when all the nodes are converged. Hence, the convergence of the nodes should be checked at every iteration to avoid unnecessary iterations. Early detection and termination methods can be used to decrease the total iteration number. Early detection and termination methods utilized for the BP polar decoder should be accurate and easy to implement. As a result of the lessons learned from the literature survey, we decided to use two methods, **G**-matrix-based detection and CRC in our Na-BPL polar decoder.

As it is explained in [20] that **G**-matrix-based early detection checks the estimated data and estimated codeword using the equality $\hat{\mathbf{x}} = \hat{\mathbf{u}}\mathbf{G}$. Since BP polar decoder estimates both dataword and codeword, this equality can be used as an early detection method. On the other hand, since this method involves a matrix multiplication, it increases the hardware implementation complexity.

The cyclic redundancy check (CRC), which is easier to implement when compared with **G**-matrix-based estimation for early detection, is employed in [23,85,60,124,125]. Two different approaches can be followed in order to use CRC for early detection.

Approach I (CRC₁): In the first approach, for a polar code $\mathcal{P}(N, K)$ where K is the length of the information bits, and $N-K$ is the number of frozen bits, the number of information bits decreases from K to $K-R$ where R is the number of CRC parity bits and the number of frozen bits is kept same. Thus, the code rate ranges from K/N to $K-R/N$ causing rate penalty.

Approach II (CRC₂): In this approach, code rate is be kept the same i.e., K/N. Number of frozen bits ranges from $N-K$ to $N-K-R$.

Both approaches are used in the literature, in [10, 60, 124] the first approach (CRC₁) is employed, while in [125] the second approach CRC₂ is considered. In our work, we utilized both approaches to present fair comparison with literature. Structure of CRC, i.e., its polynomial representation, and its length are important parameters that affect the performance of the decoder. We choose CRC-6 for $N = 128$, CRC-8 for $N = 512$, CRC-16 for $N = 1024$ and $N = 2048$. Polynomials are selected referring to papers and 5G standards documents [10, 60, 124] as:

- CRC-6 $p(x) = x^6 + x^5 + 1$
- CRC-8 $p(x) = x^8 + x^2 + 1$
- CRC-16 $p(x) = x^{16} + x^{15} + x^2 + 1$

Despite the code rate penalty, CRC usage is practical when compared with **G**-matrix-based early detection method. Besides, better error correction performance is achieved as it is to be shown in the incoming figures.

Simulations are performed to compare the performances of early detection methods in terms of BER, BLER and average iteration number. BER and BLER results are presented in Fig. 4.12. Two different code lengths $N = 1024$ and $N = 2048$ are used in simulations. It is observed that BER performance of BP polar decoder with **G**-matrix-based early detection is better than CRC based one for a single BP polar decoder. During simulations, CRC₁ approach is employed. The other performance comparison factor, average number of iterations for both early detection methods are given in Table 4.2 where it is seen that **G**-matrix-based detection leads to faster decoding process with maximum iteration number limited to 50.

Table 4.2 Comparison of average iteration number for different early detection and termination methods

Polar Code/ SNR(dB)	1	1.25	1.5	1.75	2	2.2	2.4	2.6	2.8	3
$\mathcal{P}$ (1024,512) with G-matrix-based early detection	40.4	32.5	24.2	17.4	13.6	11.2	9.4	8.3	7.47	6.76
$\mathcal{P}$ (1024,512) with CRC based early detection	42.4	34.8	25.6	17.6	12.8	10.1	8.3	7.1	6.31	5.76
$\mathcal{P}$ (2048,1024) with G-matrix-based earl detection	43.3	34.0	25.1	17.9	13.5	11.4	10.0	9.1	8.22	7.61
$\mathcal{P}$ (2048,1024) with CRC based early detection	45.4	36.0	25.6	17.3	12.4	10.1	8.7	7.69	6.98	6.42
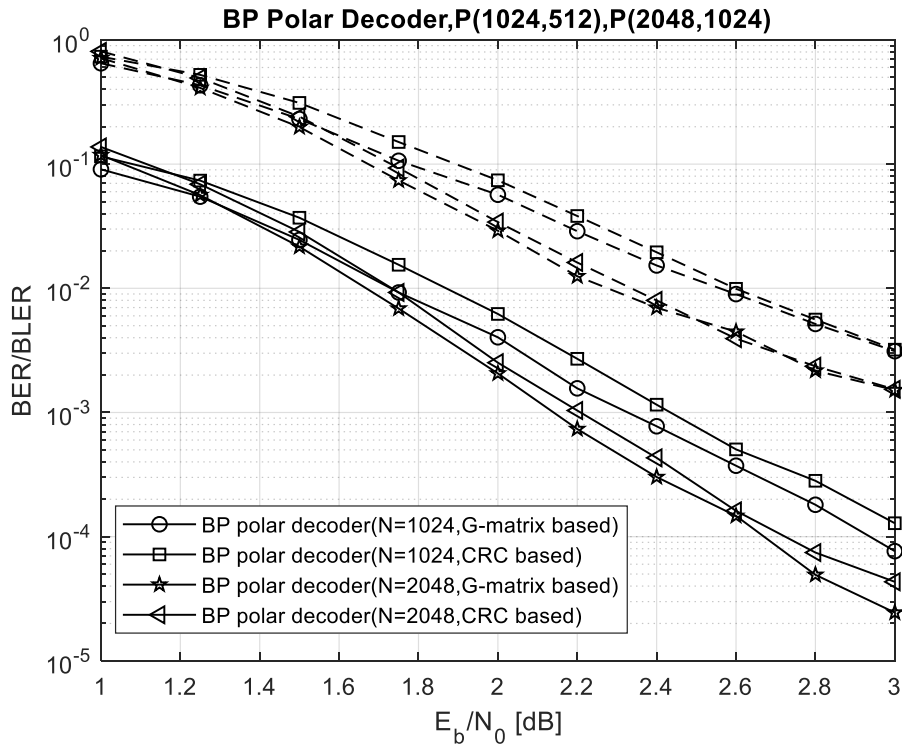
Figure 4.12 BER/BLER comparison of BP-based polar codes under different early detection and termination methods for $\mathcal{P}$(1024,512) and $\mathcal{P}$(2048,1024)

BER performance of **G**-matrix-based early detection method is better than CRC based detection, however, they achieve the same BLER performance when E_b/N_0 is larger than 2.4 dB. Complexity is another factor to be taken into account while choosing the appropriate early detection method. Complexity of the methods is directly related to hardware implementation cost. In Table 4.3 we provide the hardware complexity of the early detection methods. Hence, considering the performance graphs and complexity tables, we conclude that both methods have the potential to be utilized for practical systems. Besides, CRC is proved to be useful in list decoding.

Table 4.3 Hardware blocks that are needed to implement early detection methods

Early Detection and Termination Method	Add	Compare	$N \times N$ matrix multiplication
G-Matrix-based detection	$2N$	N	1
CRC-L based detection *(L is the length of the CRC parity bits)*	N	N	-

4.1.4 Systematic Coding of Polar Code

In systematic codewords of any error correction code, the locations of the information bits are known a-priori. This feature of systematic codes can be advantageous over non-systematic codes if receiver quickly determines whether error free transmission occurred or not. Although original version of polar codes is offered in non-systematic form [2], Arıkan also offered a systematic form of the polar code, and showed that BER/BLER improvement can be achieved via systematic polar codes [112]. When systematic and non-systematic polar codes are compared in terms of BER performance, it is observed that 0.3 dB gain is achieved for $P(256,128)$ at BER 10^{-5}. However, the same BLER performance is achieved for both type of codes.

Systematic polar encoding can be achieved considering the following steps [113].

- Let **u** be the data vector to be used by $P(N,K)$.

- Multiply this **u** with the generator matrix, $\mathbf{w} = \mathbf{uG}$.

- Force frozen bits, chosen according to your polar code construction method, to zero on **w**.

- Perform a second encoding on **w**, i.e. $\mathbf{x} = \mathbf{wG}$.

For the decoding of systematic polar code, a method, easy to implement, is proposed in [113] where a non-systematic polar decoder along with a non-systematic encoder are used as systematic polar decoder. Fig. 4.13 illustrates the systematic encoding and decoding operations.

In Fig. 4.13, systematic decoding includes a non-systematic decoding and a non-systematic encoding. As it is mentioned before, BP polar decoder estimates both the dataword and codeword. Inspired by this fact of the BP polar decoder, we propose that if the systematic decoding is performed using BP polar decoder, we don't need to use non-systematic encoding along with our non-systematic decoder, and a single BP polar decoder is sufficient. This results in no additional complexity to the decoder. Fig. 4.14 shows the achievement of a systematic decoder if a BP-based polar decoder is utilized.
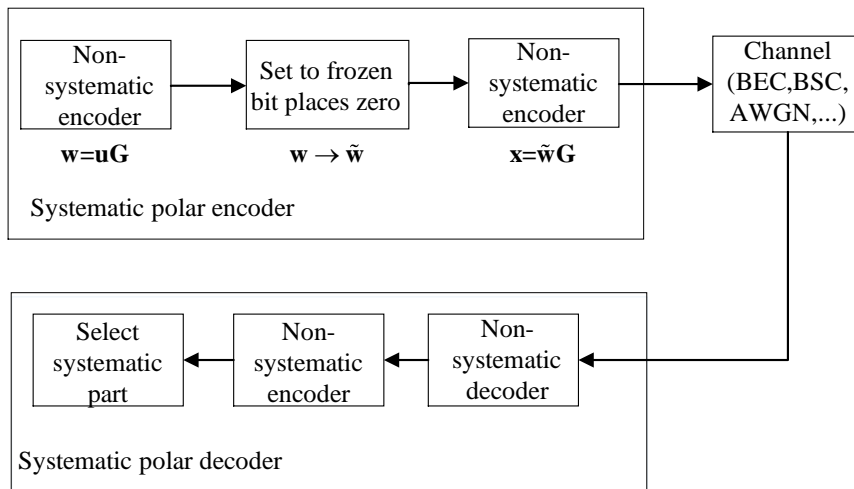
Figure 4.13 Systematic encoding/decoding of polar codes
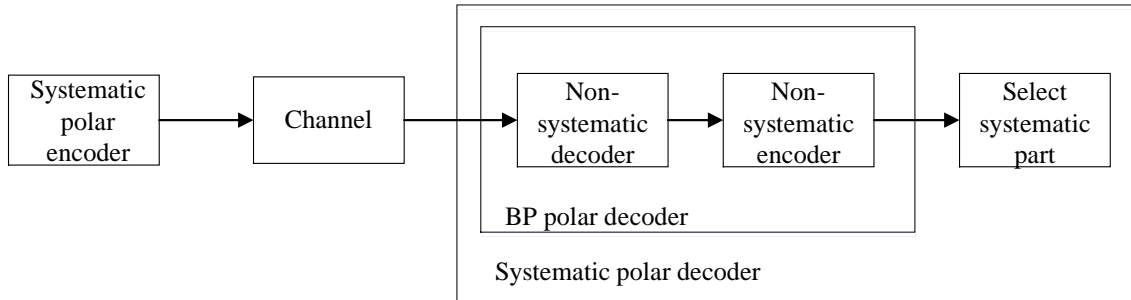
Figure 4.14 Systematic decoding of polar codes

In this scope, simulations are performed to show the effect of systematic approach in terms of error correction. In our simulations, BP polar decoder employs frozen check, uses **G**-matrix-based early detection and RT scheduling with maximum iteration number $M = 50$.
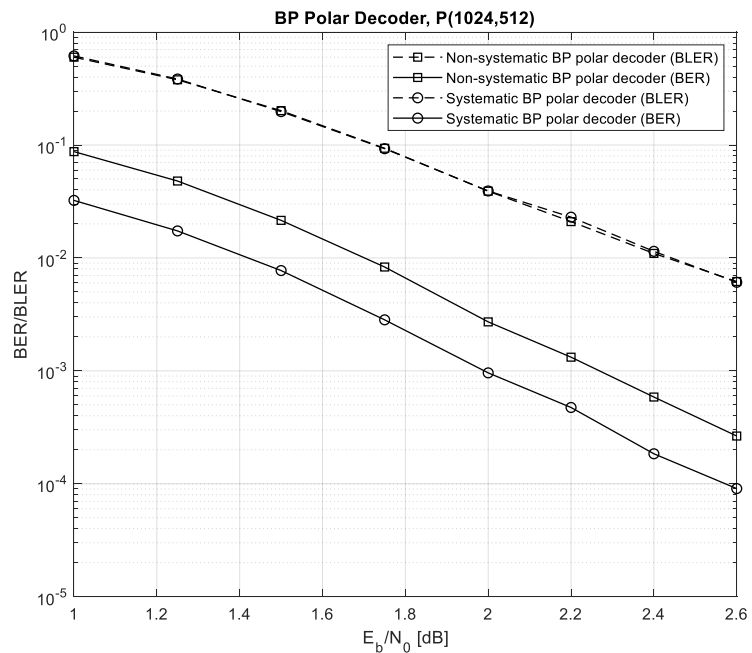
Figure 4.15 BER/BLER comparison of systematic and non-systematic BP polar decoder for $\mathcal{P}(1024,512)$
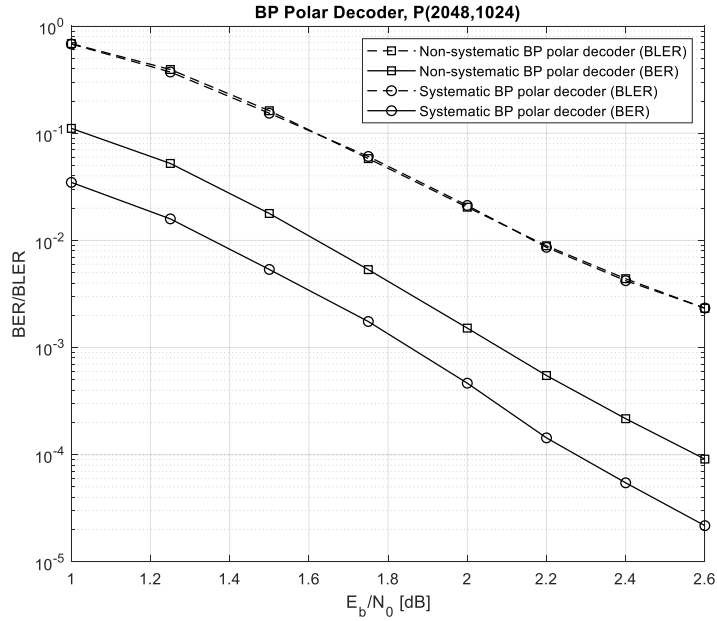
Figure 4.16 BER/BLER comparison of systematic and non-systematic BP polar decoder for $P(2048,1024)$

Fig. 4.16 and Fig. 4.17 show that BER performance gain achieved by systematic approach while BLER performance stays same when compared with non-systematic BP polar decoder. Average iteration number comparison is done in Table 4.4 between non-systematic and systematic polar codes when BP decoder with $M = 50$ is employed. Overall, average iteration number for both cases are similar.

Table 4.4 Average number of iterations comparison between systematic and non-systematic BP polar decoders

Polar Code/ SNR(dB)	1	1.25	1.5	1.75	2	2.2	2.4	2.6
Non-Systematic P (1024,512)	38.2	29.9	21.7	15.6	11.4	9.3	7.9	7.02
Systematic P (1024,512)	38.9	30.6	21.6	15.7	11.3	9.4	7.9	6.9
Non-Systematic P (2048,1024)	42.2	32.5	21.9	15.1	11.5	9.6	8.5	7.6
Systematic P (2048,1024)	42.2	31.7	21.8	15.3	11.48	9.6	8.5	7.7

4.2 Polar Code Construction

Two different kind of methodologies can be followed to construct polar codes; first type follows analytical approach and the second one is Monte-Carlo (MC) simulation based approach can be used. As an analytical approach, Bhattacharyya parameter [2] is a well-defined for the SC-based polar decoder for the binary erasure channel (BEC). Different decoders with different channel types cannot achieve optimal BER/BLER performance due to the absence of well-defined polar code construction methods [121]. To cover this deficit, some methods that relies on both analytic methods like density evolution (DE) [114], and Gaussian approximation for DE [115], and MC based designs [16, 90, 95, 96] are introduced.

Among the polar code construction methods, two of them are utilized in our studies. One of them is Bhattacharyya parameter based construction [2], and the other one is a kind of MC based construction that is enhanced by the genetic algorithm [99].

4.2.1 Bhattacharyya Parameter Based Polar Code Design

Bhattacharyya parameter based estimation is designed for binary erasure channel, and it is not the optimum construction method for AWGN, Raleigh fading channels etc. Bhattacharyya parameter of a binary-discrete memoryless channel denoted by W is defined as

$$Z(W) \triangleq \sum_{y \in Y} \sqrt{w(y\,|\,0)w(y\,|\,1)} \tag{4.1}$$

where $y \in Y$ is an output alphabet. The value of $Z(W)$ is closely related with the capacity of a channel, $C(W)$, such that $C(W) + Z(W) = 1$. Recursive calculations are performed to compute the channel capacities Bhattacharyya parameter can be recursively calculated as

$$Z(W_{2N}^{(i-1)}) = 2Z(W_N^{(i)}) - Z(W_N^{(i)})^2 \tag{4.2}$$

$$Z(W_{2N}^{(2i)}) = Z(W_N^{(i)})^2 \tag{4.3}$$

An example of combining channels is depicted in Fig. 4.17.
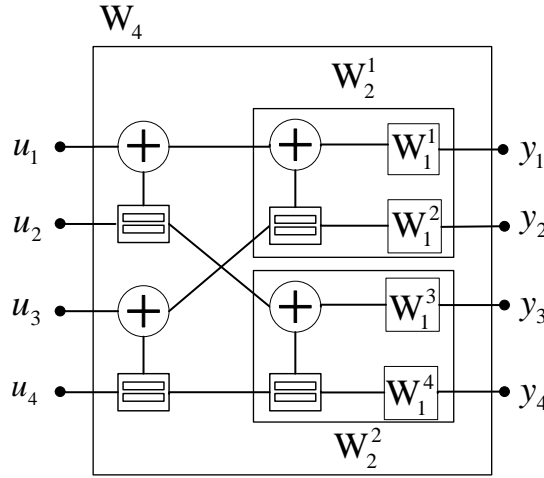
Figure 4.17 Combining channels to get polarized channels

For the rightmost side, we have $W_1^i{=}W$ where $1{\leq}i{\leq}N$. Erasure probability of the BEC can be used directly for W_1^i. However, initial value of Bhattacharyya parameter is difficult to find for Gaussian and Rayleigh channels since both channels are continuous. We prefer to use a practical construction method of polar codes in AWGN channels [116]. In [116], initial value of W_1^i is determined by bit error probability P_e of AWGN channel defined as $P_e\left(\frac{R_s R_s E_b}{N_0}\right)$ where R_s is symbol rate, R_c is code rate and E_b / N_0 is the energy per-bit. We use BPSK and AWGN channel, that is R_s is 1 and the code rate is generally set to 0.5. It is important to state that the value of E_b / N_0 in dB scale is always set to 0.5 dB, the design SNR, for the simulations of this thesis, unless otherwise indicated.

As an example, frozen bit distribution for P(2048,1024) and P(1024,512) are calculated for both BEC and AWGN channels and frozen bit locations are illustrated in Fig. 4.18, Fig. 4.19, Fig. 4.20 and Fig. 4.21.
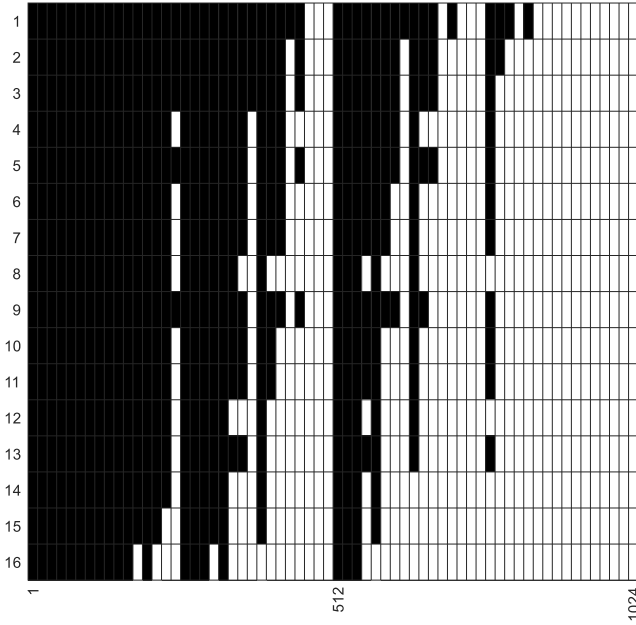
Figure 4.18 Frozen bit locations of $\mathcal{P}(1024,512)$ for BEC with erasure probability of 0.5
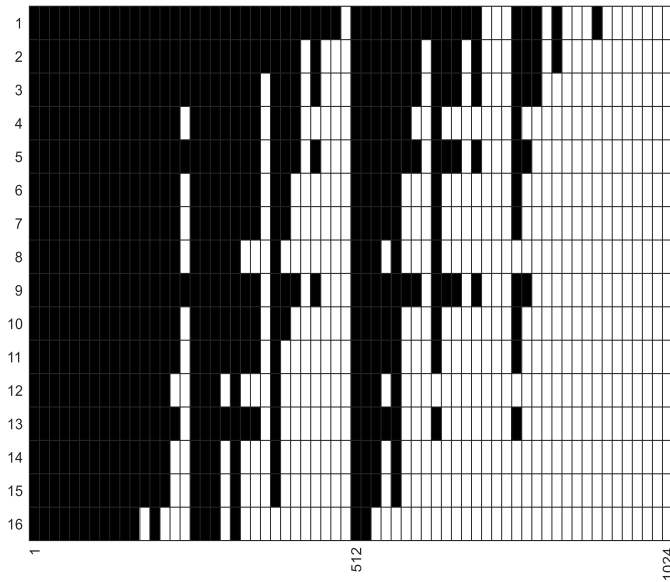
Figure 4.19 Frozen bit locations of $\mathcal{P}(1024,512)$ for AWGN with design SNR of 0.5 dB
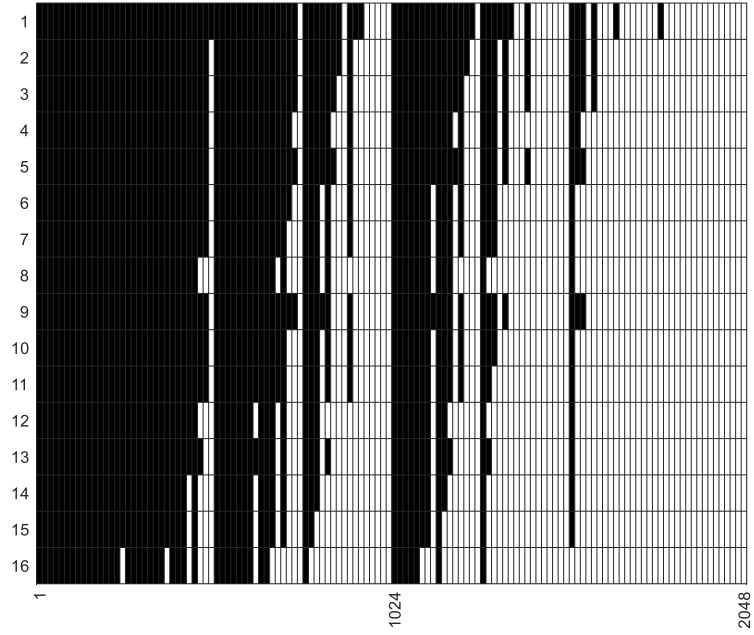
Figure 4.20 Frozen bit locations of $\mathcal{P}(2048,1024)$ for BEC with erasure probability of 0.5
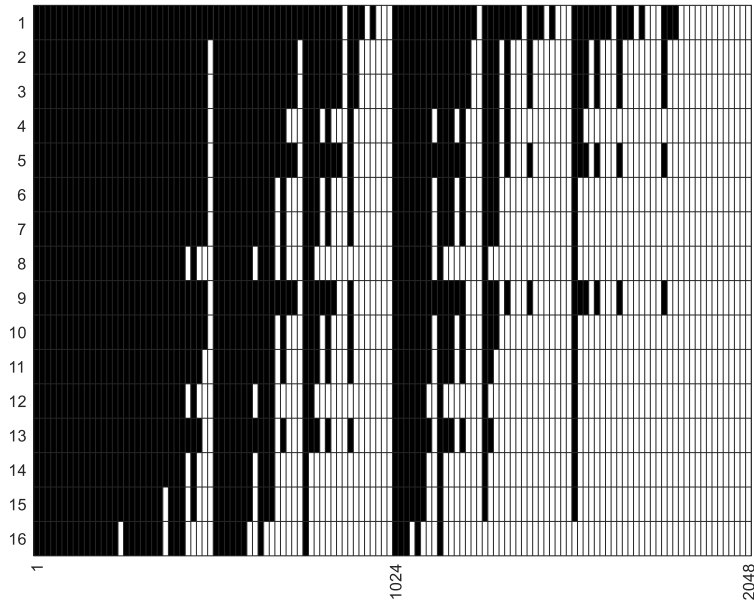
Figure 4.21 Frozen bit locations of $\mathcal{P}(2048,1024)$ for AWGN with design SNR of 0.5 dB

4.2.2 Decoder-tailored Polar Code Design using Genetic Algorithm

The polar code design method using the Genetic Algorithm (GenAlg) is proposed in [99] where remarkable results are obtained. If the polar code is designed with GenAlg, then the error correction performance of the BP polar decoder achieves the performance of SCL polar decoder, and the structure of BP decoder stays the same. The only difference is on the location of the frozen bits.

Genetic algorithm is proposed by Holland in 1975 [117] that is based on the concept of Darwin's theory of evolution. Genetic algorithms are widely used to generate high quality solutions for optimization and search problems, based on biological terms like mutation, crossover and selection. In order to explain genetic algorithm's usage in polar code construction accurately, we will explain these biological terms and the equivalents of these terms.

Population, a collection of individuals, is the candidate solution of the optimization problem. In our case, population is the number of possible information sets that lead to the best optimization i.e., low BER, low BLER. There are two factors to pay attention to when a GenAlg is employed. The first one is diversity of the population. If the diversity of the population is not high, then optimized results cannot be achieved. Secondly, population size is also an important parameter to maintain a good mating pool. Small population sizes could cause to deceleration on the GenAlg process.

Another factor in GenAlg is crossover. Crossover, the most important process of the algorithm, exchanges the genes of the families in the population. After crossover, two parents generate new offspring, in other words, child/children. Different types of crossovers are offered in the literature like one-point crossover, two-point crossover and uniform crossover. It is shown in Fig. 4.22 that in one-point crossover, genes of the parents are separated from one point, and the genes are transferred to the children. In two-point crossover, two different breaking points on parent's genes are presented. Lastly, uniform crossover creates many (randomly selected) breaking points.
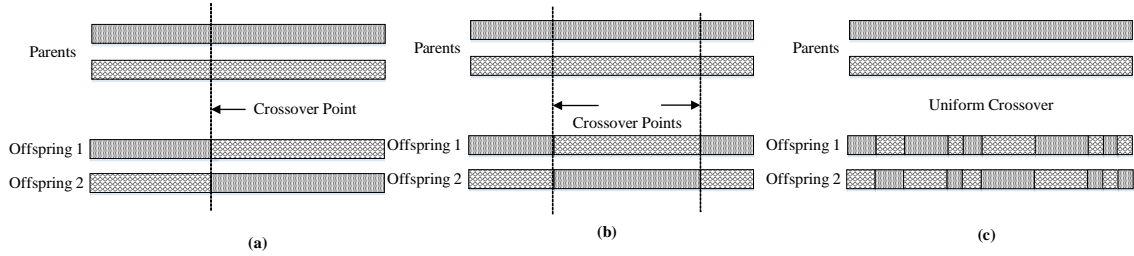
Figure 4.22 Crossover types; (a) one-point crossover, (b) two-point crossover and (c) uniform crossover

A simple crossover example presented in [99] is illustrated in Fig. 4.23 where one-point crossover is applied, and zeros represent frozen bit places while ones represent information bit places. It is seen that after crossover five frozen bit places appeared. Thus, a rate adjustment is necessary to keep coding rate at its constant value, i.e., 0.5.

$$\begin{matrix} \mathbf{0000}1111 \\ 0001\mathbf{0111} \end{matrix} \xrightarrow{\textit{Crossover}} \mathbf{00000111} \xrightarrow{\textit{AdjustRate}} 00100111$$

Figure 4.23 A simple crossover example for $\mathcal{P}(8,4)$

The outcomes of the crossovers often yield better solutions. If a generated offspring is not good enough, then it will be removed from the population through selection process. In decoder tailored polar code construction approach [99], offspring represent the location of frozen bits, and population competes with each other in terms of selected fitness function, i.e., BER/BLER performance.

Mutation is an operation used in genetic algorithm; it is used to create genetic diversity between individuals of the population. In mutation, results usually change entirely in a bad or good way. Mutated individuals are eliminated in the selection part of the GenAlg. It is stated in [99] that mutation can be created flipping a bit of an information vector in a random position. Bit flipping converts frozen bit locations to data bit locations. However, to keep code rate the same, another bit flipping at a random data location is necessary. A sample mutation example is given in Fig. 4.24.

$$00\dot{0}10111 \xrightarrow{\textit{Mutation}} 00\dot{1}10111 \xrightarrow{\textit{AdjustRate}} 00100\dot{0}11$$

Figure 4.24 A mutation example on $\mathcal{P}(8,4)$

A genetic algorithm should have a proper fitness function. Fitness function is used as a decision mechanism on survival of individuals. The aim of the channel coding is to get the optimized error correction performance, such that BER or BLER performance can be selected as fitness function [99]. Fitness function deletes information sets that has high BER/BLER values after MC simulations with a specific decoder, specific channel type and a certain SNR. In our case, we use BP polar decoder, on AWGN channel with a design SNR of 1.25 dB.

After population individuals are fed into the fitness function, a selection and population update operation should be performed. Since population size is a paramount factor to reach to accurate convergence on the optimization problem, we choose the population size as 20 individuals, i.e., a vector that contains frozen and information bit places.

Initial population content should be chosen with precision to lead GenAlg into a faster convergence. In decoder-tailored polar code design using GenAlg, the initial population is formed from splitted channel capacities constructed for AWGN channels for different design SNR using Bhattacharyya parameter based estimation. Our initial vectors that are constructed with five different design SNR as 0, 0.5, 1, 1.5 and 2 dB.

Flowchart of the GenAlg [99] is visualized in Fig. 4.25. GenAlg of [99] is developed for different type of decoders like SCL and BP, and different types of communication channels like AWGN and Rayleigh fading channel. However, since we are dealing with BP-based polar decoders on AWGN channel, we configured the process according to our parameters. In order to achieve the optimum results, process repetition number, K, is determined to be a number between 20 and 40 where $1 \leq i \leq K$. Alternatively, instead of using a constant number for K, GenAlg can work until the same information set with the best error correction performance is selected for three consecutive processes. We prefer to use BLER results of the individuals at 1.25 dB as fitness function under BP polar decoding for BPSK modulated AWGN channel.
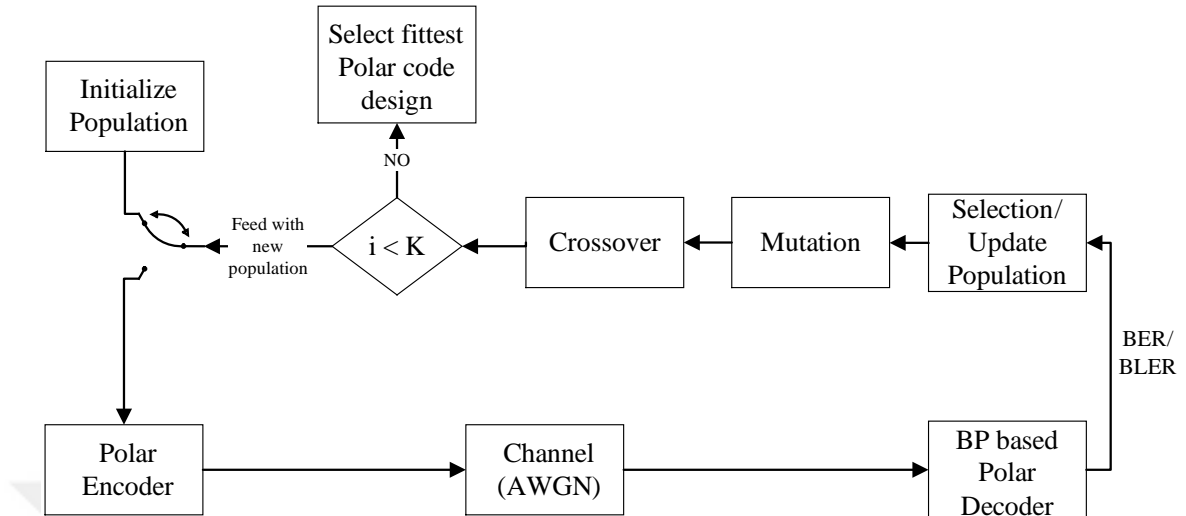
Figure 4.25 Flowchart of GenAlg based polar code construction

Frozen bit locations achieved utilizing GenAlg for polar code construction are depicted for the codes $\mathcal{P}(1024,512)$ and $\mathcal{P}(2048,1024)$ in Fig. 4.26 and Fig. 4.27, respectively.
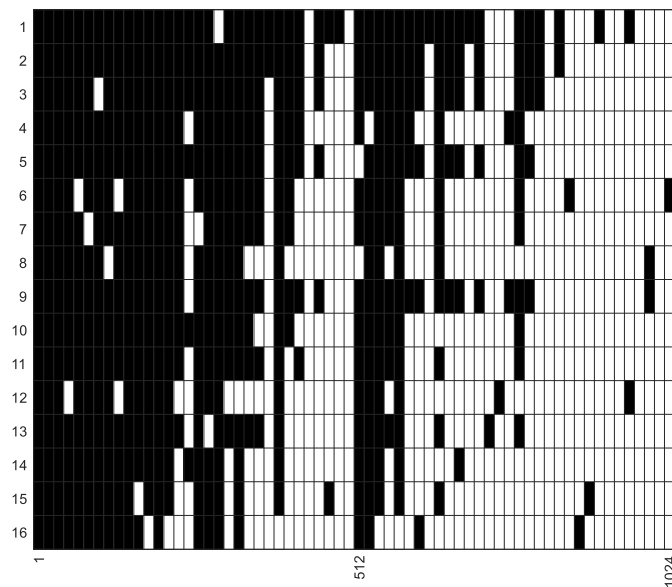
Figure 4.26 Frozen bit locations of $\mathcal{P}(1024,512)$ with design SNR of 1.25 dB for BPSK modulated AWGN channel
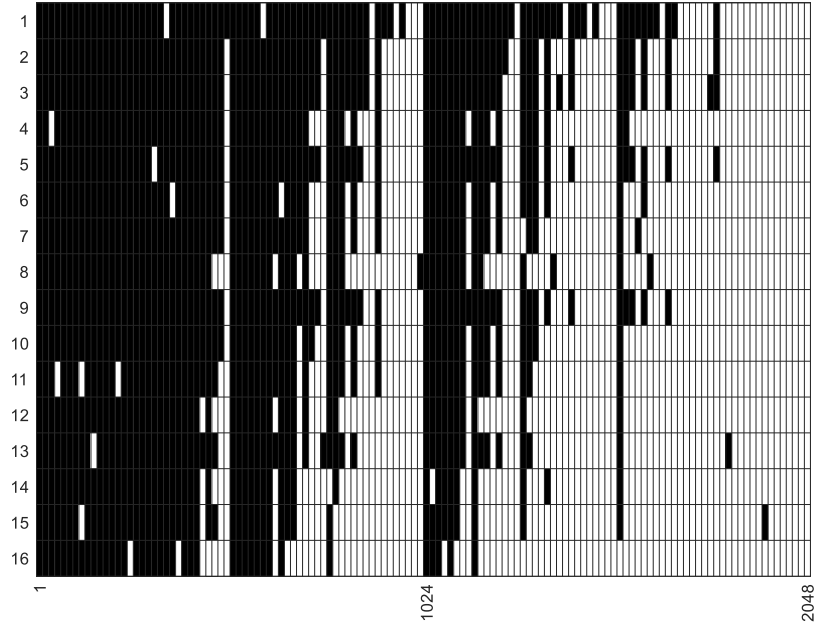
Figure 4.27 Frozen bit locations of $\mathcal{P}(2048,1024)$ with design SNR of 1.25 dB for BPSK modulated AWGN channel

We can provide a comparison between performances of polar decoders when different polar code construction methods are utilized. Fig. 4.28 and Fig. 4.29 present error correction performances of polar codes constructed with Bhattacharyya parameter based estimation and GenAlg based approach. G-matrix-based detection, RT scheduling and $M = 50$ are applied during simulations. It is seen from these figures that GenAlg based polar construction at 1.25 dB SNR achieves better performance for different code lengths and channel SNRs.
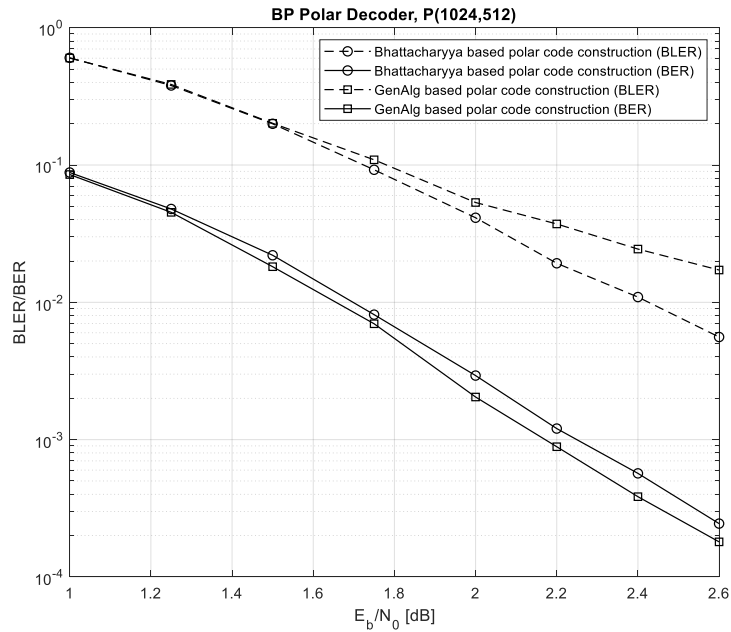
Figure 4.28 BER/BLER comparison between Bhattacharyya parameter and GenAlg based polar code construction when BP polar decoder is utilized for $\mathcal{P}(1024,512)$ under BPSK modulated AWGN channel
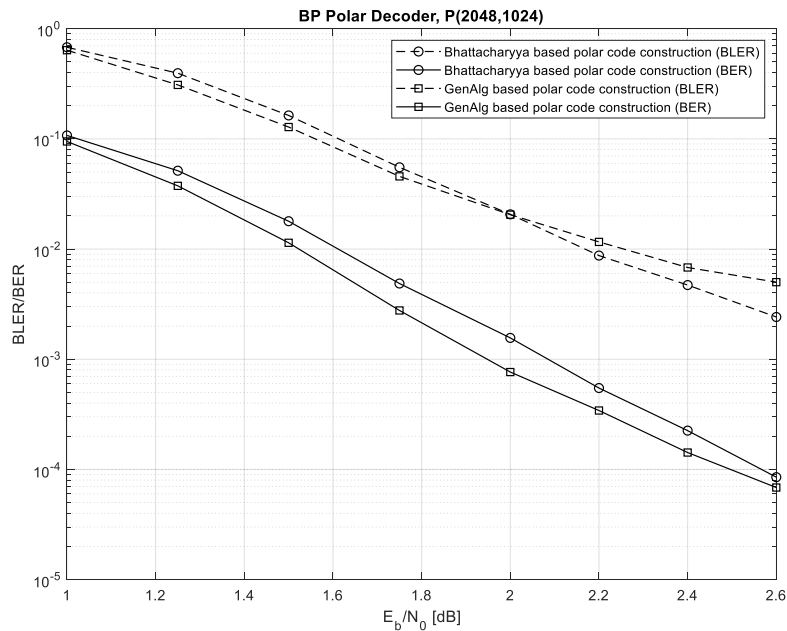
Figure 4.29 BER/BLER comparison between Bhattacharyya parameter and GenAlg based polar code construction when BP polar decoder is utilized for $\mathcal{P}(2048,1024)$ under BPSK modulated AWGN channel

4.3 Na-BPL Polar Decoder

Shannon's theorem [1] states that it is possible to achieve error-free communication for a given SNR and bandwidth. During the past many decades, numerous error detection and error correction schemes are developed. Error correcting codes are divided into two main categories; block codes and convolutional codes. Block codes include Hamming codes, repetition codes, BCH nodes, Reed-Solomon codes, Reed-Muller codes, LDPC codes, and polar codes. In convolutional coding, the codeword is formed using the convolution of dataword and impulse response. Convolutional codes are concatenated using interleavers to obtain more powerful like turbo codes and serially concatenated convolutional codes. In this thesis, we consider the use of weak noise for the enhancement of the error correction capability of the channel codes mainly targeting the polar codes.

The decoders use the artificially generated noise. Different noise intensities are applied to L different decoders, and at least one decoder achieves correct decoding. It is also important to state that artificial noise should be small in power when compared with the communication channel noise. Otherwise, decoding failure occurs for all of the L decoders. Zero mean additive white Gaussian noise (AWGN) is chosen as the artificial noise. Although we use the artificially generated noise for BP-based polar decoders, the other type of code decoders can use our approach.

BP polar decoder has three types of errors, unconverged, falsely converged, and oscillating errors. There are some proposed methods in the literature to avoid these type of errors. In order to avoid oscillating errors, noise injection into the propagating messages at the intermediate stages (have not to be a code system) is proposed in [118]. Noise injection in belief propagation based decoding is considered in [59] to overcome the error floor problem caused by small log-likelihood ratio clipping values. Clipping threshold on LLR values is decisive on the complexity of the decoder. The addition of noise as a post processing method for BP polar codes is proposed in [19]. Post processing process is performed when CRC check fails, and an estimation is done to determine whether a falsely converged, an unconverged or an oscillating error occurred or not. The post processing operation is performed in accordance with the estimation. For each case of post processing operation, random sign changes are used to achieve accurate of the convergence. Different

from the studies of [19, 59, 118], we are going to propose a novel artificial noise injection method that boosts the performance of polar decoder significantly.

Another aspect of our proposed method of noise-aided belief propagation based list (Na-BPL) polar decoder is its list decoding property. List decoding is proposed by Elias in [119]. In list decoding, a number of candidate messages compete for the decoder's estimate. List decoding is shown to be effective especially in Reed-Solomon codes [120] and in successive cancellation decoding of polar codes [10].

BP-based list decoder (BPL) decoder is proposed in [62] with the multi-trellis approach. BPL polar decoder in [62] includes permutations of original factor graph. In [62], different permuted factor graphs are chosen randomly, or, cyclic shifts of the factor graph of the polar code shown in Fig. 3.7, are used. Fig. 4.30 presents the block diagram of the BPL decoding operation. List size changes with the number of differently constructed trellis structures.

Our proposed approach called noise aided belief propagation based list (Na-BPL) polar decoder consists of three stages as shown in Fig. 4.30. The first stage includes L parallel (branch) BP polar decoders where L is the list size. The structure of the decoders $BP_1, BP_2, \ldots, BP_N$ are the same, and it is presented in Fig.13a. The proposed decoder structure is illustrated in Fig. 4.30 where $\mathbf{y}$ is channel output, $\mathbf{n}_{1, 2, \ldots, L\text{-}1}$ is artificial noise, and $\tilde{\mathbf{y}}_{1, 2, \ldots, L\text{-}1}$ represent the artificial noise added version of channel output $\mathbf{y}$. It is important to state that no artificial noise is used for the first branch. In the second stage of the decoder, early detection and termination methods to estimate decoder's convergence are used. Different early detection and termination methods can be applied for our decoder i.e., perfect knowledge based (PKB), $\mathbf{G}$-matrix-based, and CRC based. In the third stage of decoder, a post decision is applied to the output of each branch. A variety of post decision mechanisms can be utilized to lead Na-BPL to achieve more accurate results. In this manner, four different post decision mechanisms are utilized such as: argmin based, leader of converged decoders (LCD), average based assumption (ABA), and correlation based decision (Corr). Since we are going to utilize PKB method for our decoder in this section, there is no need to use any post decision mechanism. In this method, BP polar decoder stops when estimated data $\hat{\mathbf{u}}$ and user data $\mathbf{u}$ are equal to each other. Similar

comparison can be made between estimated codeword $\hat{\mathbf{x}}$ and $\mathbf{x}$. Thus, details of post decision mechanisms are posted to Section 4.4.

We adapt a convention to indicate the type of decoder together with its parameters as in:

Decoder_type(List size, Early Detection Mechanism, Post Decision Mechanism)

Some examples of the convention are given as:

- Na_BPL(L=10, **G**-matrix, Argmin): Noise aided belief propagation list decoder with list size 10. **G**-matrix based early detection is used with argmin based post decision mechanism.
- Na_BPL(L=32, PKB): Noise aided belief propagation list decoder with list size 32. Since PKB is used as early detection there is no need for a post decision mechanism.

The key idea in Na-BPL decoder is to use weak artificial noise for perturbation. For artificial noise, zero mean additive Gaussian noise is used. Artificial noise is generated using

$$n_i(x) = \mu + \sigma_i \times wgn(x) \tag{4.4}$$

where $1 \leq i \leq L - 1$ and $1 \leq x \leq N$.

Artificial noise, in other words, zero mean additive white Gaussian noise with standard deviation starting from 0 with 0.0125 incremental steps is added to the received signal, $\mathbf{y}$, at each branch of the list decoder. Equation (4.4) is used for the generation of artificial noise. In (4.4), $wgn(\mathrm{x})$ is the noise generator function for the zero mean and unity variance Gaussian noise samples, mean μ is set to zero to avoid any bias to the channel output. Standard deviation is denoted by σ_i.
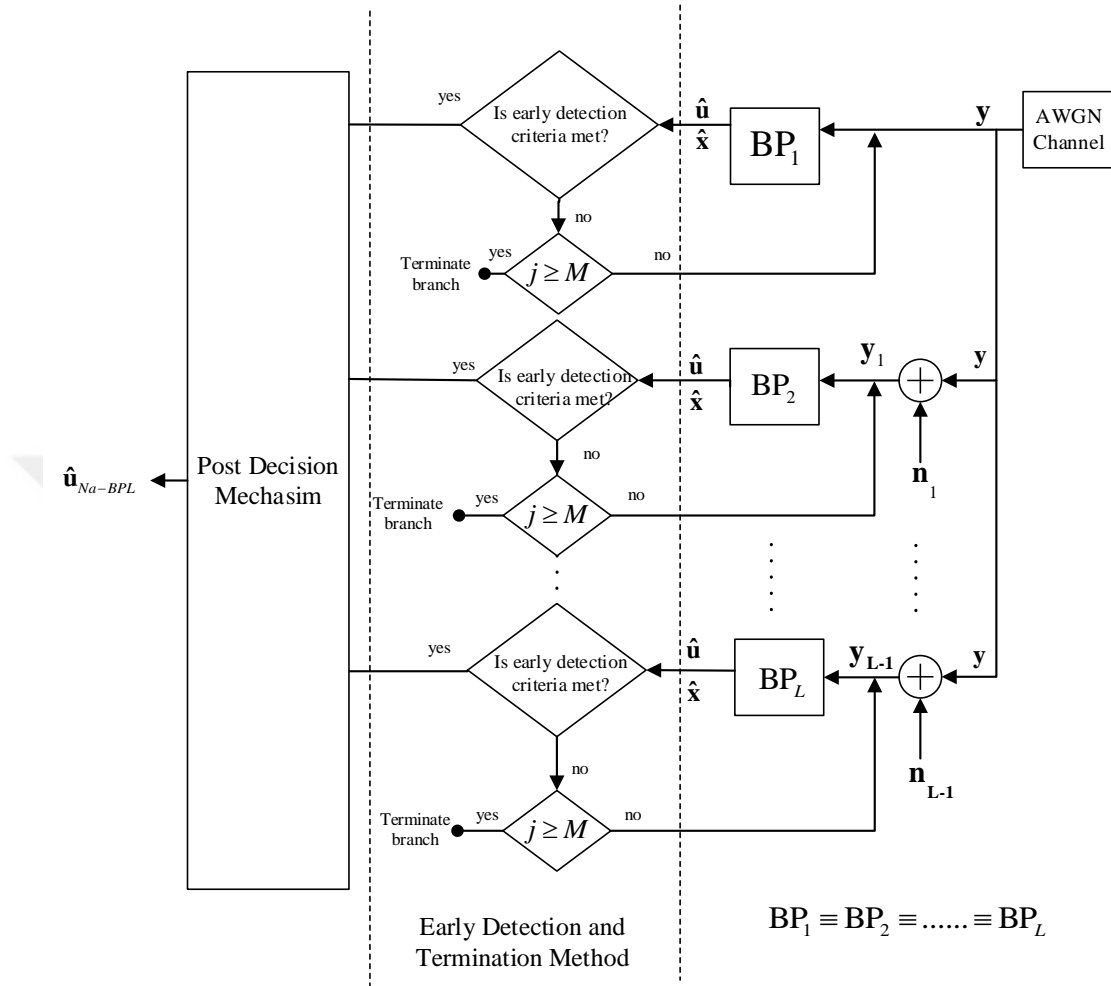
Figure 4.30 Na-BPL polar decoder design

Artificial noise added to the channel output should be too small compared to channel SNR. For instance, in our trials, standard deviation of the noise ranges from 0 to 0.3875, and the increment rate can change according to list size i.e. 0.0125, 0.00625.

After each iteration of the decoding operation, an early detection and termination criteria is checked, if the criteria are satisfied in any branch, then decoding is assumed to be successful. Estimated data and codeword are obtained as the output of the system can be used for further processing if needed. On the other hand, if none of the parallel branches meets the early detection and termination criteria, then decoding is assumed unsuccessful. When an unsuccessful decoding occurs i.e. none of the branches is successful then, the output of the first branch on the Na-BPL decoder is accepted as estimated data.

Up to this point, we discussed the Na-BPL polar decoder. From now on, we will work on different aspects of the proposed decoder. First of all, BER/BLER performance under different list sizes will be demonstrated. In Fig. 4.31 and Fig. 4.32, error correction performance in terms of BER of the Na-BPL polar decoder is demonstrated for $\mathcal{P}(1024,512)$ and $\mathcal{P}(2048,1024)$ respectively, and for these simulations the list sizes $L = 1, 2, 4, 8, 16, 32$ are used. Simulations are performed using BPSK modulated signal on AWGN channel and iteration number M equals to 50, PKB early detection method is selected, and RT scheduling is employed. Besides, frozen nodes are kept constant to provide faster convergence of the Na-BPL decoders. For the simulation results presented in Fig. 4.31 and Fig. 4.32, artificial noise ranging from 0 to 0.3875 is used. Standard deviations of the artificial noise are denoted by $\sigma_1, \sigma_2, ...,\sigma_{32}$ and noise signals are indicated by $n_{1, 2,..., L-1}$ for the list size of $L = 32$. For instance, $\sigma_1, \sigma_2, ...,\sigma_{32}$ can take the values 0, 0.0125, 0.025,…., 0.3875, respectively. For $L = 16$, $\sigma_1, \sigma_2, ...,\sigma_{16}$ can be chosen as 0, 0.0125, 0.025,…., 0.1875, respectively.

Block error rate performance also improves as the list size increases. Simulations are held done for $\mathcal{P}(1024,512)$ and $\mathcal{P}(2048,1024)$, and the results are depicted in Fig. 4.33 and Fig. 4.34 where the list sizes are chosen as $L=1, 2, 4, 8, 16, 32$.

Na-BPL Polar Decoder, P(1024,512)
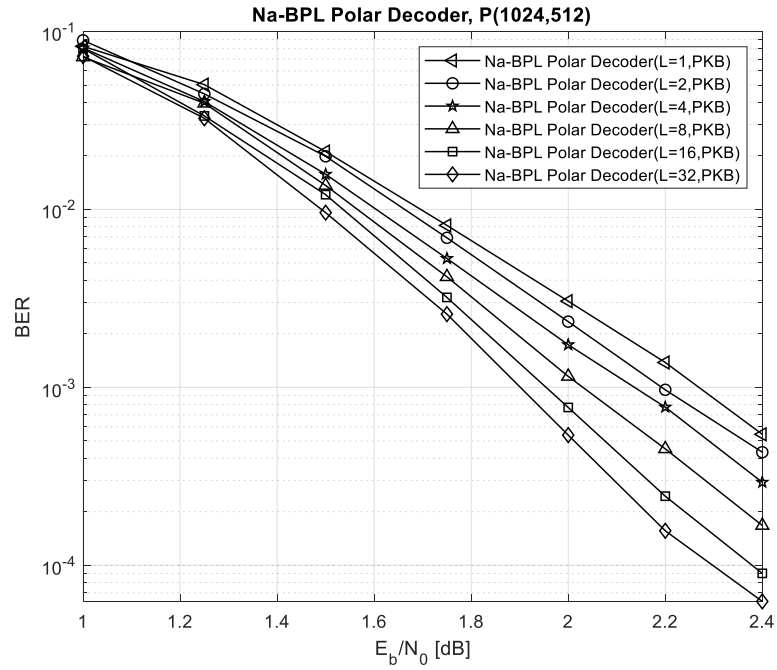
Figure 4.31 BER performance of Na-BPL decoder with different list sizes $\mathcal{P}(1024,512)$ under BPSK modulated AWGN channel

Na-BPL Polar Decoder, P(2048,1024)
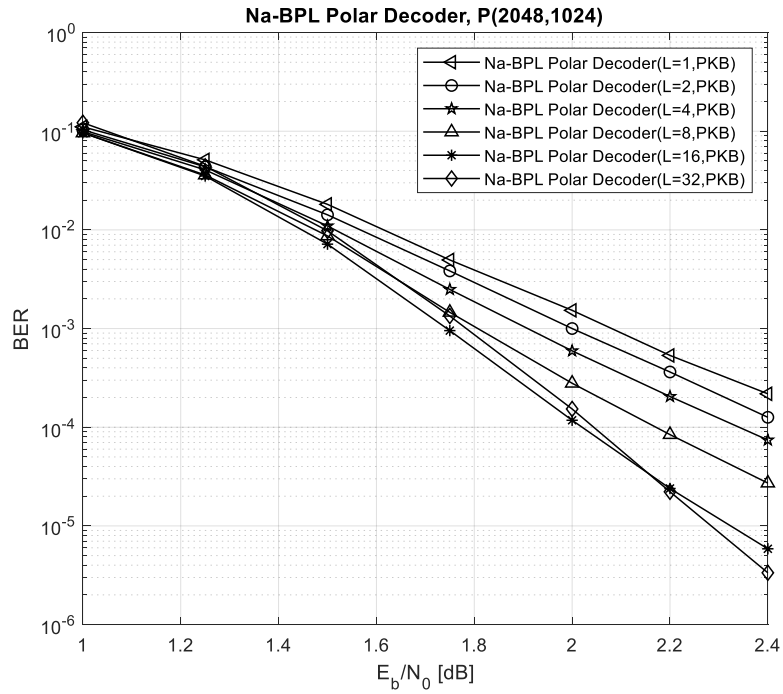
Figure 4.32 BER performance of Na-BPL decoder with different list sizes $\mathcal{P}(2048,1024)$ under BPSK modulated AWGN channel
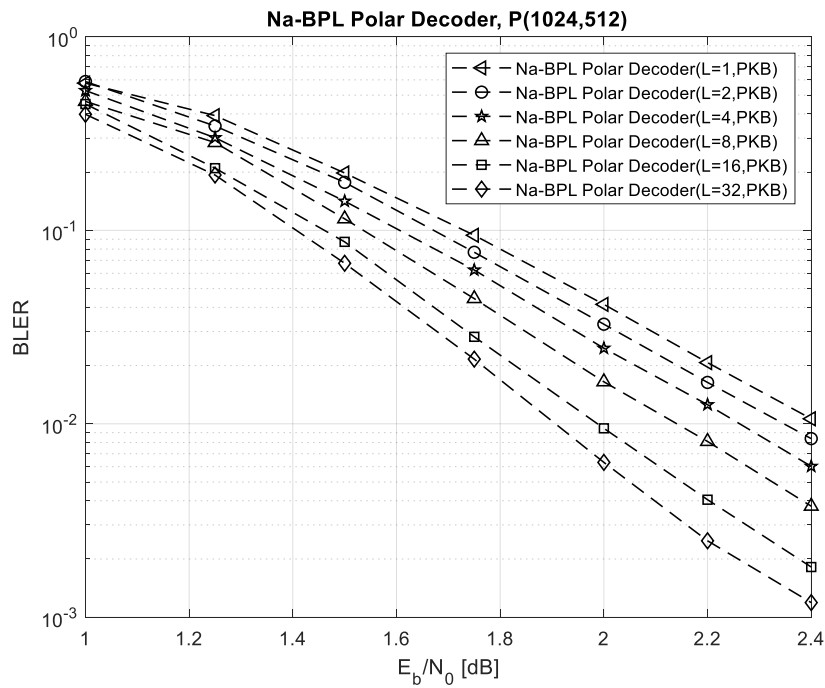
Figure 4.33 BLER performance of Na-BPL decoder with different list sizes for
$\mathcal{P}(1024,512)$ on BPSK modulated AWGN channel
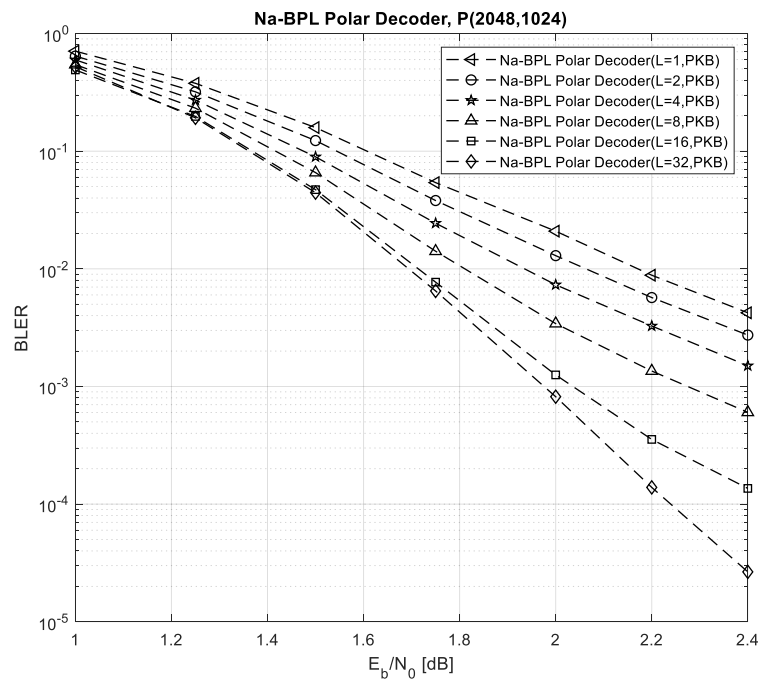
Figure 4.34 BLER performance of Na-BPL decoder with different list sizes for
$\mathcal{P}(2048,1024)$ on BPSK modulated AWGN channel

High noise level can also be used when low-level artificial noise does not improve the error correction performance of the decoder. However, high noise levels may harm decoder's performance. For this reason, we provide an example to show the effect of artificial noise intensities with different standard deviations. Three noise intensity ranges are utilized for Na-BPL decoder on $\mathcal{P}(1024,512)$ with list size 16 and $M = 50$. In this way, Na-BPL I, Na-BPL II and Na-BPL III decoders are constructed with standard deviation ranges of

$(0, 0.00625, 0.025,\ldots, 0.0938), (0, 0.0125, 0.025,\ldots, 0.1875), (0, 0.125, 0.25,\ldots, 1.875)$

respectively. The effects of selected noise intensities used to construct Na-BPL decoder are depicted in Fig. 4.35 from which it is seen that second design achieves the best error correction performance. It is seen from Fig. 4.35 that the magnitude of artificial noise intensity is a crucial factor on Na-BPL performance.
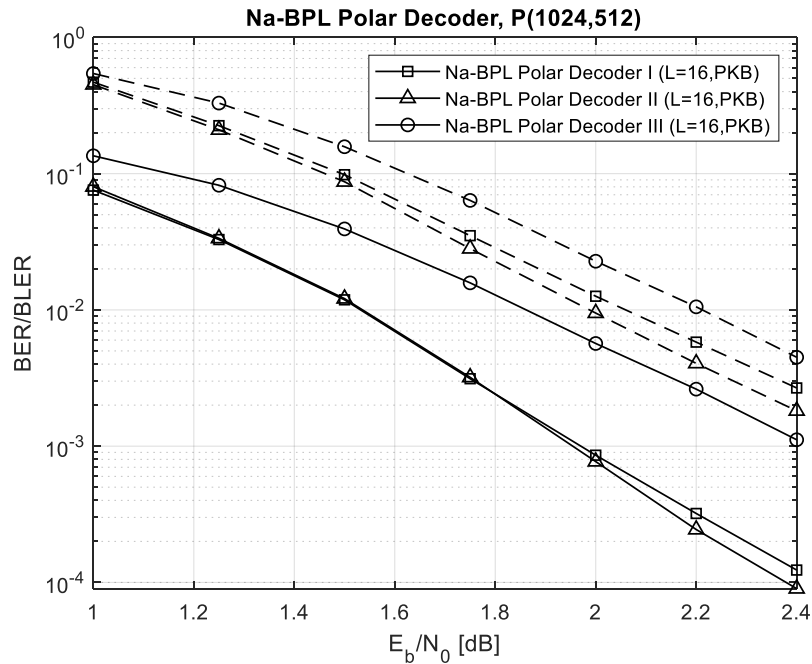
Figure. 4.35 BER/BLER performance of Na-BPL decoder with different noise intensities with fixed list size 16 for $\mathcal{P}(1024,512)$ under BPSK modulated AWGN channel

The maximum number of iterations performed by a single BP decoder of Na-BPL unit has effect on the polar decoder's performance. Different iteration numbers 50, 100, 200, 500,

1000 are used for Na-BPL decoder with $L = 16$ for $\mathcal{P}(1024,512)$. It is shown in Fig. 4.36 that as the maximum iteration number increases, BER and BLER performance of the Na-BPL decoder increase as well. We aim to replace CRC-aided SCL decoder via BP-based decoder considering its parallel decoding capability; however, the increase in iteration number eliminates the advantage of parallel decoding. Keeping maximum iteration number M between 50 and 200 seems to be reasonable to get optimized error correction performance without increasing latency significantly.
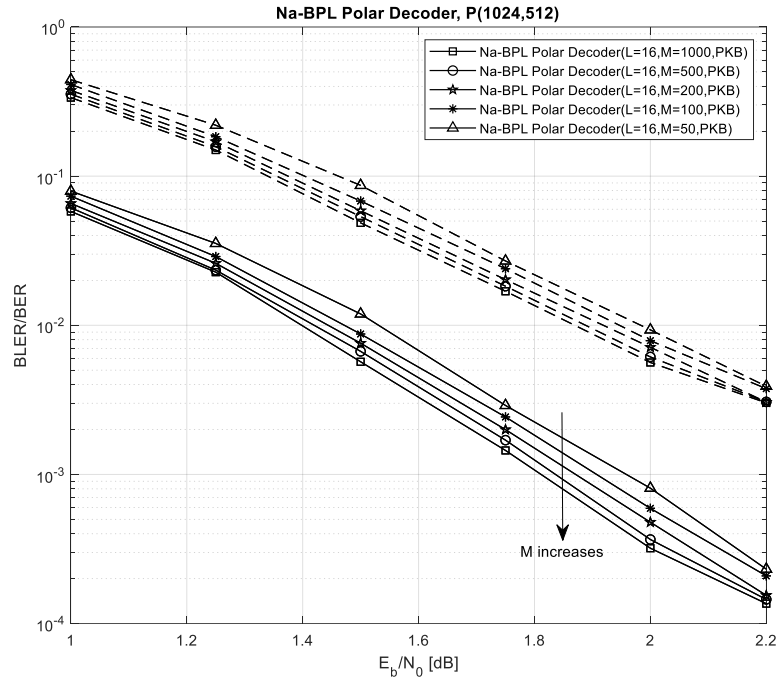
Figure 4.36 BER performance of Na-BPL decoder with different number of iterations for fixed list size 16 on $\mathcal{P}(1024,512)$ under BPSK modulated AWGN channel

4.3.1 Folded Na-BPL Polar Decoder

Folding approach can also be used by Na-BPL polar decoder benefiting from the fact that Na-BPL decoder consists of L identical BP decoders. Folded Na-BPL structure presented in Fig. 4.37 can be employed when low complexity is the issue under concern where $1 \leq j \leq L$ and $1 \leq i \leq M$. However, in this case the throughput of the decoder is decreased, since different artificial noise intensities are added to the decoder one by one until a correct convergence of the decoder is provided.

The working principle of folded Na-BPL is similar to the studies containing multi-trellis factor graphs obtained permuting the original factor graph [58, 60, 61, 63, 124]. In this type of decoder, each decoder runs until early detection criteria is satisfied. If criteria can't be fulfilled for M iteration number, then next decoder with different trellis structure starts to decoding operation. The proposed decoder of this thesis always use the same trellis structure with the aid of different weak artificial noise intensities.
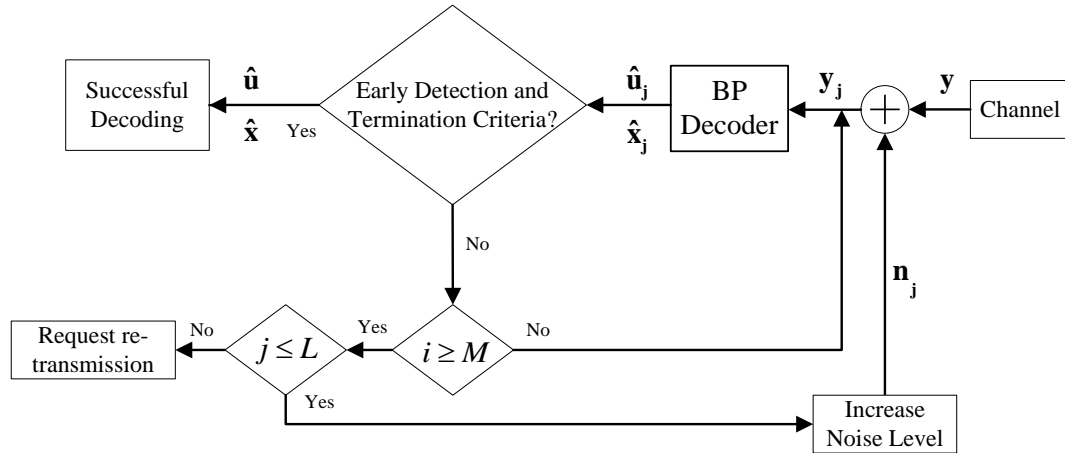
Figure 4.37 Folded Na-BPL structure

BER/BLER performance of the folded structure is similar with parallel structure, Fig. 4.30. However, the use of folded Na-BPL increases average iteration number dramatically. Table 4.5 shows that huge difference on average number of iterations between parallel and folded Na-BPL is observed for low signal-to-noise ratio, i.e., SNR < 2 dB. Beyond 2 dB, decoding speed of the decoders approaches to each other.

Table 4.5 Average number of iterations comparison between Na-BPL and folded Na-BPL polar decoders ($M = 50, L = 16$)

Polar Code/ SNR(dB)	1	1.25	1.5	1.75	2	2.2	2.4
Na-BPL P(1024,512)	33.7	22.8	15.3	10.7	8.2	7.11	6.33
Folded Na-BPL P(1024,512)	765.5	416.7	172.5	68.19	28.63	17.73	12.23

Stochastic perturbation provided by artificially generated noise intensities cause stochastic resonance to occur. A weak input signal and a nonlinear system are needed to achieve

stochastic resonance. Normally, weak signal cannot cause an output signal on nonlinear system. On the other hand, by the aid of artificially generated noise signals, weak signal becomes detectable. Amount of noise that enhances the input signal is vital in this process. In order to observe SR effect, noise signal with intensities (decided by standard deviation) from 0 to some value are mounted on the weak input signal before passing through the threshold part. At each trial, output performance is calculated. When noise intensity versus output performance graph is plotted, characteristic SR curve is observed. A valid SR curve has concave shape. Consequently, benefit of noise in signal detection becomes a fact for specific set of noise values. Notice that if there is no artificial noise signal introduced to nonlinear system, output performance will be zero since weak input signal has no effect on nonlinear system.

Stochastic resonance curves can also be observed in our decoder. In this perspective, our Na-BPL decoder represents nonlinear system and received signal **y**, is weak input signal. As mentioned before, unsuccessful decoding occurs when decoder converges inaccurately. In order to beat inaccurate convergence, we add different levels of artificial noise into received signal. In Na-BPL decoder, there are L parallel identical BP polar decoders. Received signal **y** is fed to all decoders with only difference of artificial noise level. As a result, amplitude level of noise increases as the number of parallel decoders increases. In this scope, we try to observe stochastic resonance curves by using the accurate error correction results of the Na-BPL decoder. Fig. 4.38 shows the results of folded Na-BPL decoder with list size 32. The standard deviation of artificial noise ranges from 0 to 0.3875 with increment step size of 0.0125. Folded Na-BPL employs PKB method to detect the convergence of the decoder. For simulations $\mathcal{P}(256,128)$ is utilized for BPSK modulated AWGN channel, and seven different channel SNRs are used. Success rate is calculated by dividing successfully decoded frame number for a given noise level to the total decoded frame number.

$$\text{success rate}_i = \frac{\text{successful frames in a certain arfticial noise}_i}{\text{total frame sent}} \qquad (4.5)$$

To observe stochastic resonance curve, we omitted the successful decoding results of artificial noise free branch of Na-BPL decoder. Ninety percent of the frames are decoded successfully in the first stage (artificial noise free) of the decoder. Fig. 4.38 contains resonance curves for the decoders that include artificial noise. Fig. 4.38 shows that stochastic resonance can be observed in a Na-BPL polar decoders. It can be concluded that as the AWGN channel SNR increases, a small level of artificial noise is sufficient to achieve accurate convergence of the decoder.
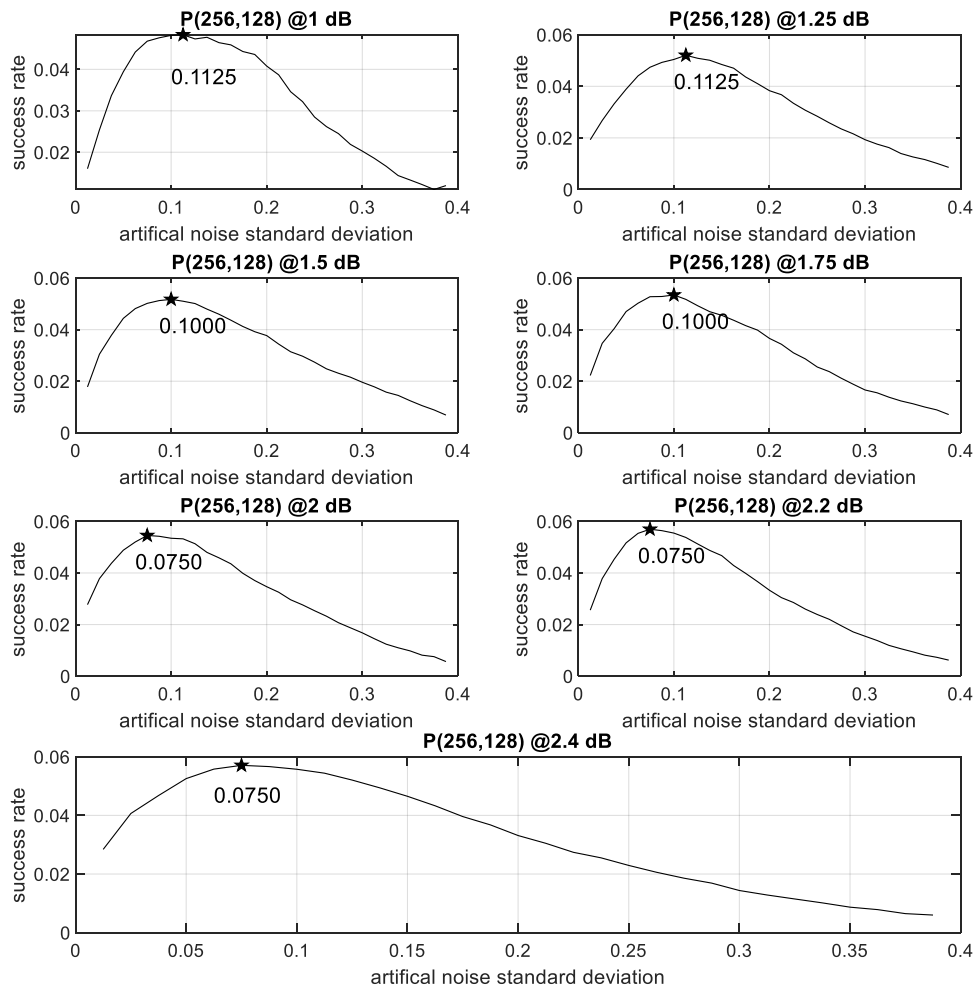
Figure 4.38 Stochastic resonance curves achieved by folded Na-BPL polar decoder for $\mathcal{P}(256,128)$ with list size 32 under BPSK modulated AWGN channel

4.3.2 Systematic Na-BPL Polar Decoder

Systematic coding is also applied for Na-BPL decoder. As it is illustrated in Fig. 4.14, the systematic decoder of [113] is well suited for BP-based polar decoder. Error correction performance is enhanced using systematic decoding for Na-BPL when compared with non-systematic one. BER and BLER performances of systematic and non-systematic codes involving $\mathcal{P}(1024,512)$ and $\mathcal{P}(2048,1024)$ for different list sizes are compared to each other. During simulations, BPSK modulated AWGN channel, Bhattacharyya parameter based polar code construction, RT scheduling, perfect knowledge based early detection and constant frozen node approach are employed. Fig. 4.39 and Fig. 4.40 show the BER/BLER achievements for $\mathcal{P}(1024,512)$ and $\mathcal{P}(2048,1024)$, respectively.
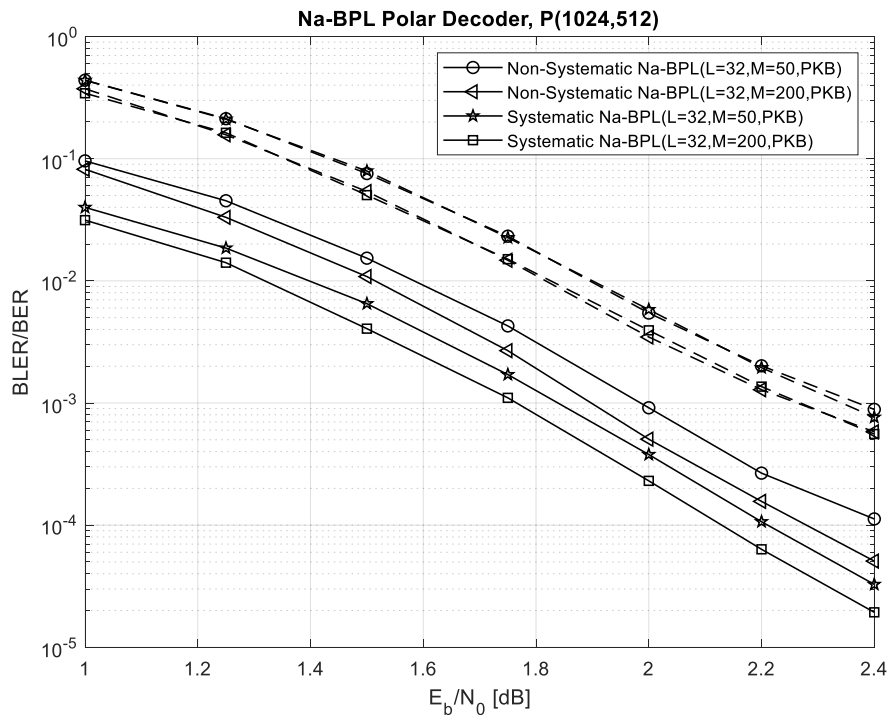
Figure 4.39 BER/BLER performance comparison of non-systematic Na-BPL and systematic Na-BPL decoders for $\mathcal{P}(1024,512)$
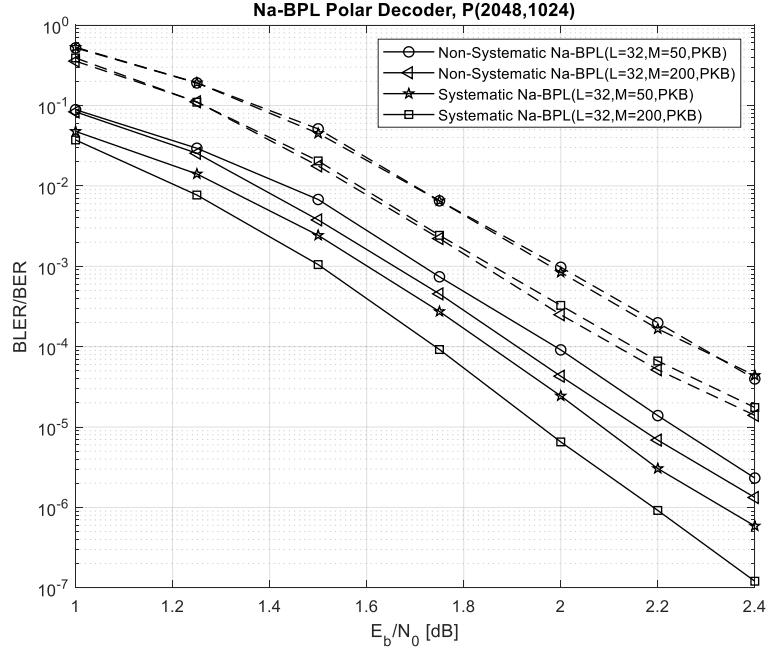
Figure 4.40 BER/BLER performance comparison of non-systematic Na-BPL and systematic Na-BPL decoders for $\mathcal{P}(2048,1024)$

Another comparison is made with CRC aided SCL decoder, Fig. 4.41. Fig. 4.41 shows BER comparison of non-systematic Na-BPL, systematic Na-BPL, non-systematic CRC-aided SCL decoder, and systematic CRC-aided SCL decoder for $\mathcal{P}(2048,1024)$ for which the list size is set to 32. Maximum iteration number, M, of Na-BPL decoders is set to 200. BER performance of the systematic Na-BPL decoder is 0.1 dB away from the state-of-the-art non-systematic CRC-aided SCL decoder, but, it cannot compete with systematic CRC-aided SCL decoder yet. However, comparison between systematic Na-BPL and non-systematic CRC-aided SCL decoder is not fair. Because, Na-BPL decoder uses perfect knowledge based early detection and termination method. On the other hand, no complexity is added to Na-BPL decoder by the use of systematic approach.
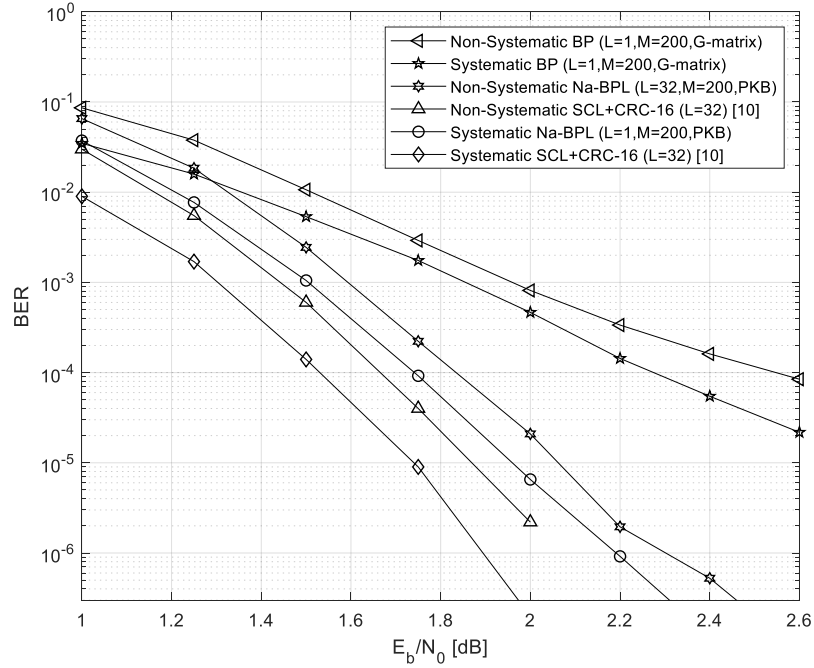
Figure 4.41 BER performance comparison of BP, Na-BPL and CRC-aided SCL decoders for $\mathcal{P}(2048,1024)$

4.3.3 Na-BPL Polar Decoder with Genetic Algorithm based Code Construction

GenAlg based polar code construction, in other words frozen bit selection, is presented in section 4.2.2, and it is shown to be effective on BP polar decoding when BPSK modulated AWGN channel is used. We will use the same polar code that is constructed using a single BP polar decoder for Na-BPL decoder simulation. Simulation results show that Na-BPL decoder with GenAlg based polar code outperforms Na-BPL decoder with Bhattacharyya parameter based polar code. Fig. 4.42 and Fig. 4.43 depict the BER/BLER graphs to show the effect of GenAlg for $\mathcal{P}(1024,512)$ and $\mathcal{P}(2048,1024)$.

GenAlg based polar code with Na-BPL($L = 32$,PKB) decoder approaches to the performance of the CRC-aided SCL decoder employing $\mathcal{P}(2048,1024)$. It is shown in Fig. 4.44 that there is 0.1dB difference for BER of 10^{-6}. Same amount of BER gain is achieved with systematic decoding of polar codes. Thus, it is straightforward to arrive in the thought that decoder-tailored GenAlg based polar code construction can be applied to the systematic Na-BPL decoder. In the original paper of GenAlg based construction [10], it is

103

stated that the proposed approach can also be applied to the systematic polar code when it is decoded with BP decoder on BPSK modulated AWGN channel. However, we observed that initial population has the best performance when systematic coding is applied thus, crossover and mutation could not achieve any further error correction performance improvement.
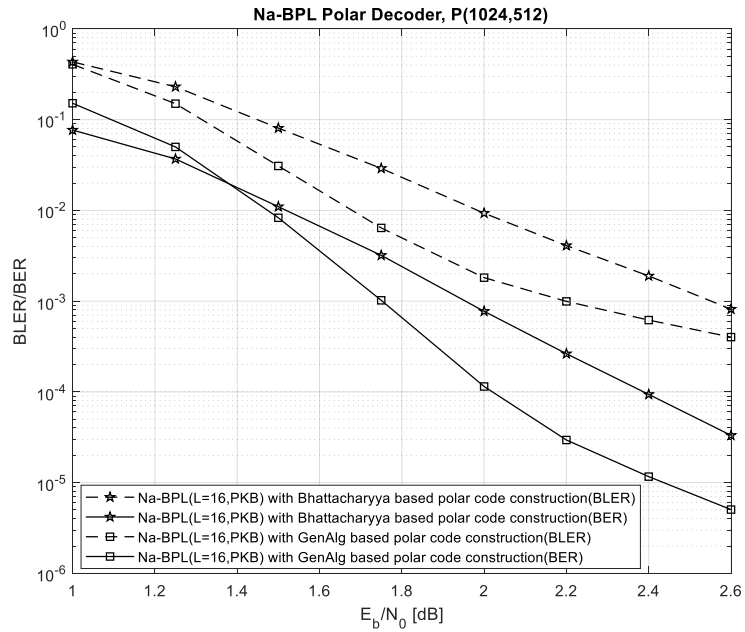
Figure 4.42 BER/BLER performance comparison under different polar code construction methods for $\mathcal{P}(1024,512)$ at BPSK modulated AWGN channel
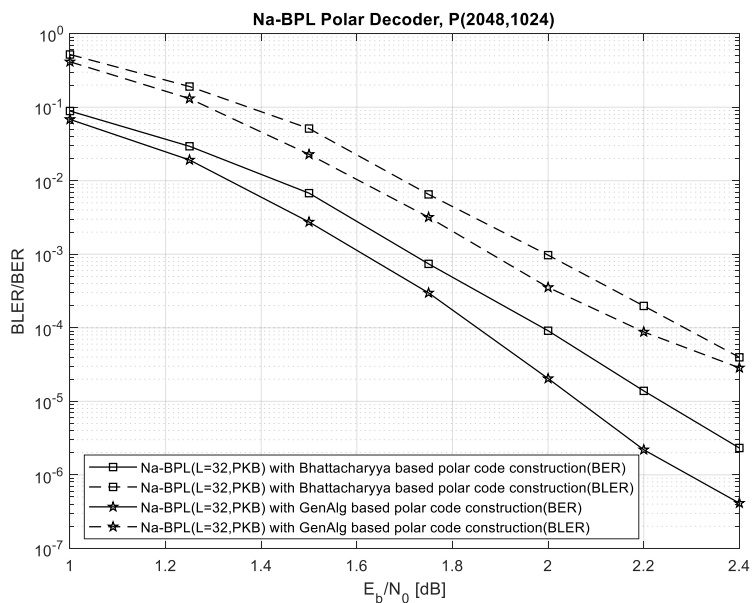
Figure 4.43 BER/BLER performance comparison under different polar code construction methods of $\mathcal{P}(2048,1024)$ for BPSK modulated AWGN channel
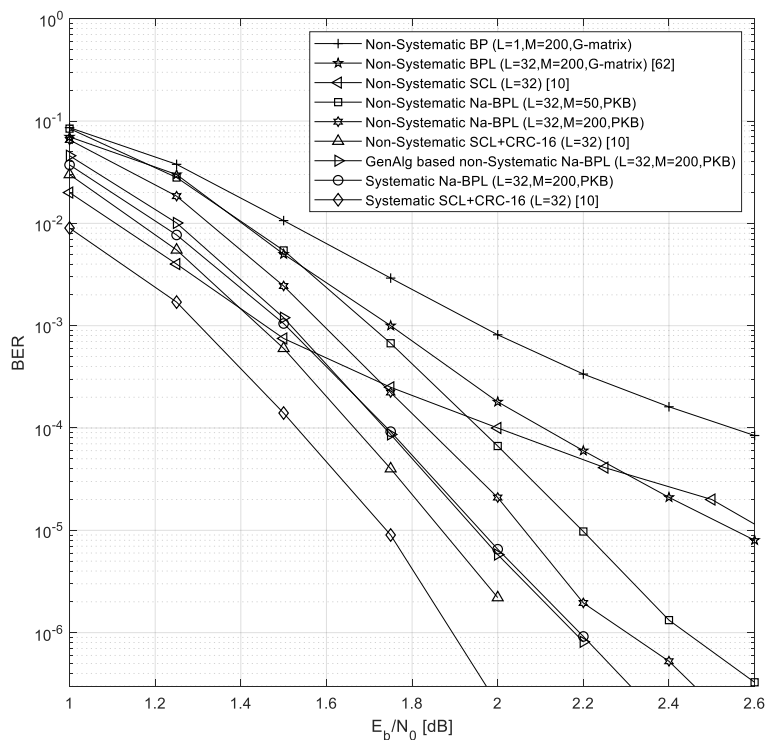
Figure 4.44 BER performance comparison of different types of polar decoders for $\mathcal{P}(2048,1024)$

4.4 Na-BPL Polar Decoder for Practical Applications

For practical application of the Na-BPL decoder, it is critical to implement it in log-domain for lower complexity and for ease of implementation. For these reason, the proposed Na-BPL decoder is implemented in log-domain and is simulated using log-domain message propagation equations, early detection method and post decision mechanism. Simulation results that are depicted till this point are achieved using non-logarithmic equations that includes multiplication and division operations. These operations are difficult to implement in hardware as the code length increases. For this reason, logarithmic versions of the propagation messages are considered in this sub-section. Complexity of any decoder is an important factor that affects its utilization on practical communication devices. The complexity reduction technique, min-sum (MS) approximation of the BP polar decoder, is proposed in [104] to reduce the complexity of the decoder. Logarithmic versions of the message propagation equations are approximated for complexity reduction. Due to approximations, performance of the decoder decreases. In order to compensate the performance loss, a scaling factor is used in [19], and it is named as scaled MS (SMS) BP and is shown with equations from (3.1) to (3.4). Since, our Na-BPL decoder also suffers from the high complexity due to its parallel structure, we utilized SMS BP for our decoder, i.e., for Na-BPL.

Furthermore, PKB method is utilized to detect early detection of convergence of Na-BPL decoder in the previous sub-section. Since PKB isn't a realistic case for any decoder, other methods should be used for a fair comparison with the studies of the literature. Since polar codes will be used in communication systems (for now), it is important to present block error rate performances rather than bit error rate performances. In this context, BLER results are going to be shared in this section.

4.4.1 Post Decision Mechanisms

In this thesis, we used a set of post decision mechanism that boosts our Na-BPL decoders' error correction performance. Since early detection criteria does not contain sufficient information for an accurate decision, the use of a post decision mechanism is vital for better judgement. In our study, we are going to use four different post decision mechanisms, and these post decision mechanisms are:

106

a. Argmin based post decision: Argmin based decision relies on Euclidian distance between received codeword and estimated codewords as formulated in

$$\hat{\mathbf{x}}_{BPL} = \arg\min_{\mathbf{x}_i, i \in \{1,\dots,L\}} \min \|\mathbf{y} - \hat{\mathbf{x}}_i\| \tag{4.6}$$

Since list decoder has a number of estimated codewords, the Euclidian distances between the estimated codewords and the received codeword **y** give us a measure of success. The branch with minimum Euclidian distance is selected as the winner decoder. Argmin based post decision is utilized in BPL polar decoder of [62].

b. Leader of Converged Decoders: In this approach, when an early detection condition is met on at least two branches, the branch with the less intensity of artificial noise is accepted as successfull. This method can be perceived as wasting the potential of the list-based decoder, but its error correction performance is not low such that we measured the error correction performance of each branch separately in proposed Na-BPL decoder with a list length of 16, and BLER is calculated for 50000 frames. It can be seen from the results of Fig. 4.45 that the success of the first branch is normally higher in the Na-BPL structure. Thus, it can be concluded that LCD method seems to be a promising approach and it can be considered for practical applications.
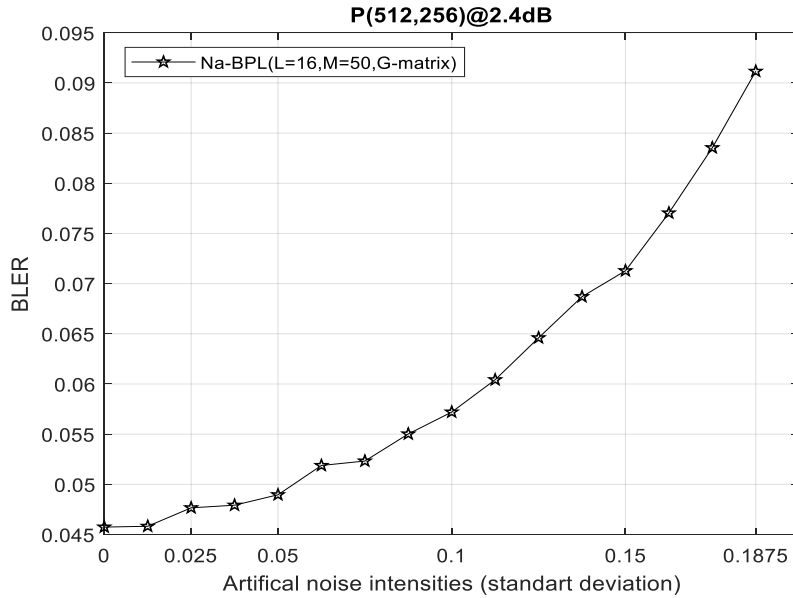
Figure 4.45 BLER performance of parallel branches of Na-BP list decoder $\mathcal{P}(512,256)$

c. Average based Assumption (ABA$_{type}$): In this method, the branch results (in terms of bits or likelihood) that meet the early detection conditions are evaluated and selected when they are above a certain threshold value. This method can be applied in two different ways.

Type 1: In ABA$_1$, the decision results of the branches are considered and voting is performed. According to the vote, the value of the bit is determined. For example, for a list size of 16 assume that the result of 10 branches, e.g. $\hat{\mathbf{u}}_1, \hat{\mathbf{u}}_2, \hat{\mathbf{u}}_5, \hat{\mathbf{u}}_6, \hat{\mathbf{u}}_7, \hat{\mathbf{u}}_9, \hat{\mathbf{u}}_{11}, \hat{\mathbf{u}}_{14}, \hat{\mathbf{u}}_{15}, \hat{\mathbf{u}}_{16}$, satisfy the early detection condition. The results of 10 successful branches are summed bit by bit and denoted by $\hat{\mathbf{u}}_{sum}$. As a result, there is a contribution between 0 and 10 for each bit in the vector $\hat{\mathbf{u}}_{sum}$. The result can be achieved by applying a predetermined threshold value (e.g. 8 (80%)) on $\hat{\mathbf{u}}_{sum}$.

Type 2: In ABA$_2$, the absolute log-likelihood values of the branches that satisfy early detection criteria are summed [126]. For example, let's assume that for a list size of 16, five branches ($\hat{\mathbf{u}}_1, \hat{\mathbf{u}}_9, \hat{\mathbf{u}}_{11}, \hat{\mathbf{u}}_{15}, \hat{\mathbf{u}}_{16}$) achieve convergence. The sum of the absolute values of the vector elements is calculated, e.g. $\sum|\hat{\mathbf{u}}_1|$. The branch giving the maximum summation result is accepted as the winner decoder.

d. Correlation based decision (Corr): Correlation between the codewords $\hat{\mathbf{x}}_i$ estimated by Na-BPL branches satisfying early detection criteria and the channel output $\mathbf{y}$ is calculated. Then, the branch with maximum correlation is accepted as the winner decoder.

4.4.2 Simulation Results of Na-BPL Decoders with Different Early Detection Methods and Post Decision Mechanisms

In section 4.3, the theoretical PKB approach is used in the Na-BPL decoder. In this section, we consider the use of Na-BPL for realistic scenarios. First, comparison between different scenarios that can be applied to Na-BPL decoder is inspected. Next, comparison with literature is presented in order to show that our idea works and it is a promising candidate for upcoming 5G frameworks. Fig. 4.46 shows the different cases of Na-BPL decoder.

P(512,256)
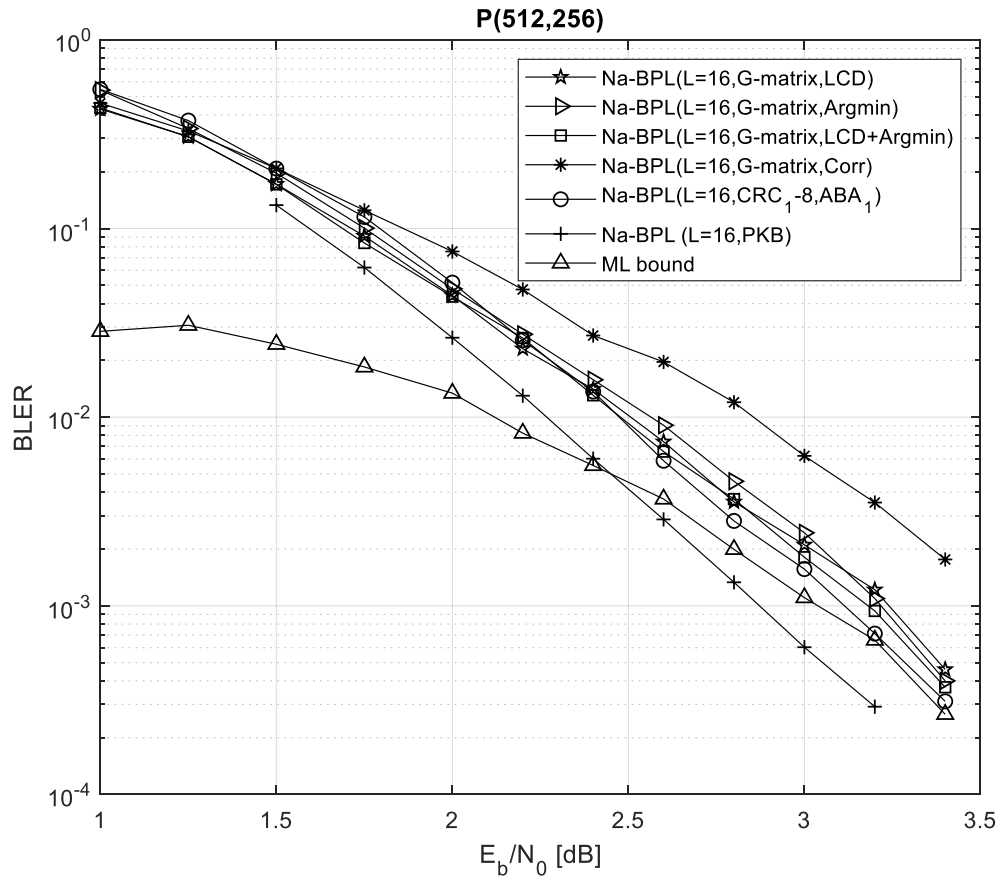
Figure 4.46 BLER performance comparison of different types of Na-BPL polar

In Fig. 4.46, list size is kept the same as 16 for all cases. Noise intensities ranges from 0 to 0.1875 with increment step size of 0.0125. Maximum iteration number is set to 50. For simulations employing **G**-matrix-based method, CRC and PKB early detection techniques are utilized. ML decoder lower bound is also evaluated. Half-way scheduling is applied. From Fig. 4.46, we can infer that

- Na-BPL decoder with approach-I of CRC-8 and post decision ABA_1 almost achieves the BLER performance of ML decoder after 3.2 dB. With CRC approach-I, code rate becomes 0.484.
- Correlation based post decision mechanism has the worst error correction performance while LCD approach is promising as mentioned in 4.4.1.

- Post decision mechanism of argmin and LCD can be utilized together, argmin is applied whenever at least two branch of the Na-BPL decoder satisfies **G**-matrix-based stopping condition.

It is possible to evaluate a lower bound for the error probability of linear codes under ML decoding [10]. To do this, each time BP decoding fails then it is checked that whether decoded codeword was more likely close to transmitted codeword rather than received codeword or not. After making this comparison, if decoded codeword dominates then we understand that the optimal ML decoder would surely misdecode received codeword, **y**, as well. Frequency of this event is counted as the lower bound on the error probability of the ML decoder and so ML bound curve is constructed.

The performance comparison between the proposed Na-BPL decoder and BPL decoder in [62,125] is presented in Fig. 4.47 where two variants of Na-BPL and BPL structure with CRC based stopping are compared. CRC approach-II is utilized for all the cases and in this approach code rate is kept the same. Besides, conventional scheduling, list size of 32 and maximum iteration number of 200 is used in order to lead fair comparison. Artificial noise intensities of Na-BPL ranges from 0 to 0.3875 with increment step size of 0.0125. Simulation results show that the proposed Na-BPL decoder achieves the same error correction performance as the original BPL structure when a decent post decision mechanism is selected. In addition, CRC control is performed for every iteration after the completion of 20th iteration to achieve accurate convergence of Na-BPL.
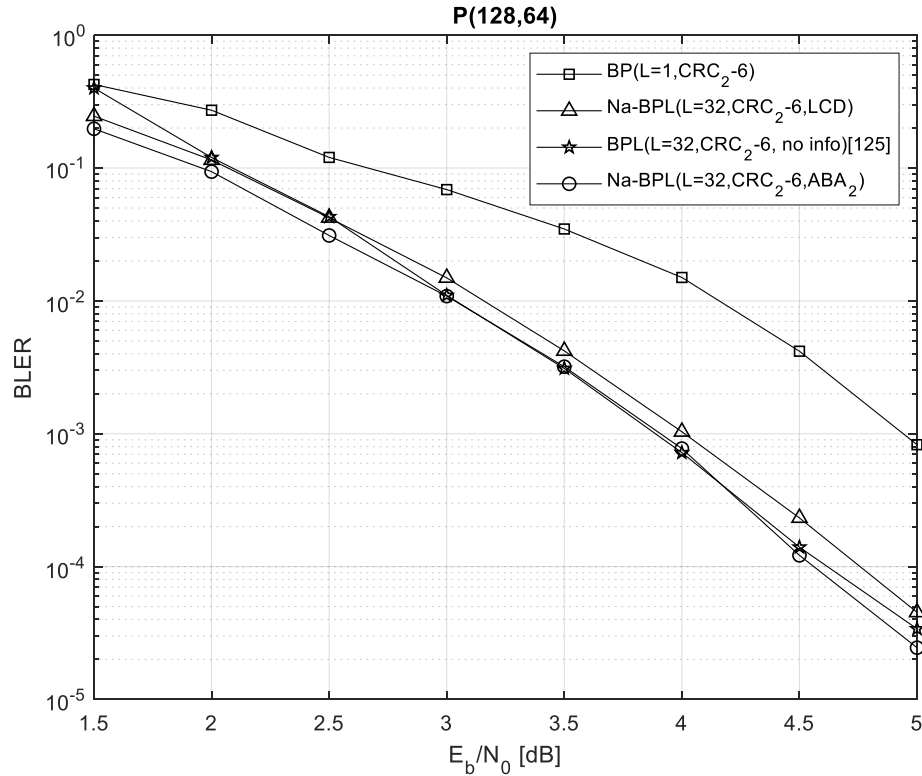
Figure 4.47 BLER performance comparison of different types of Na-BPL and BPL polar decoders for $\mathcal{P}(128,64)$

Another comparison for the proposed Na-BPL decoder is done with the literature results using the code $\mathcal{P}(1024,512)$. We use simulation results provided in [124] which uses multi-trellis structures presented in [60,62] and selects permuted trellis structures in a cleverer manner. The same simulation scenarios of [124] are used to show that our Na-BPL decoder is a promising candidate among BP-based polar decoders. Convectional scheduling, maximum iteration number of 200 and equal list size (10) are utilized. Noise intensities of Na-BPL ranges from 0 to 0.225 with increment step size of 0.025. The same CRC polynomial with approach-I, CRC_1, is utilized. Method presented in [124] is named as MAXSON and it uses 10 different factor graphs (FG) to increase error correction performance of the BP-based polar decoder. Results of MAXSON and Na-BPL with perfect knowledge based early detection is also depicted in Fig. 4.48 and Fig. 4.49 which is a zoomed version of Fig. 4.48.
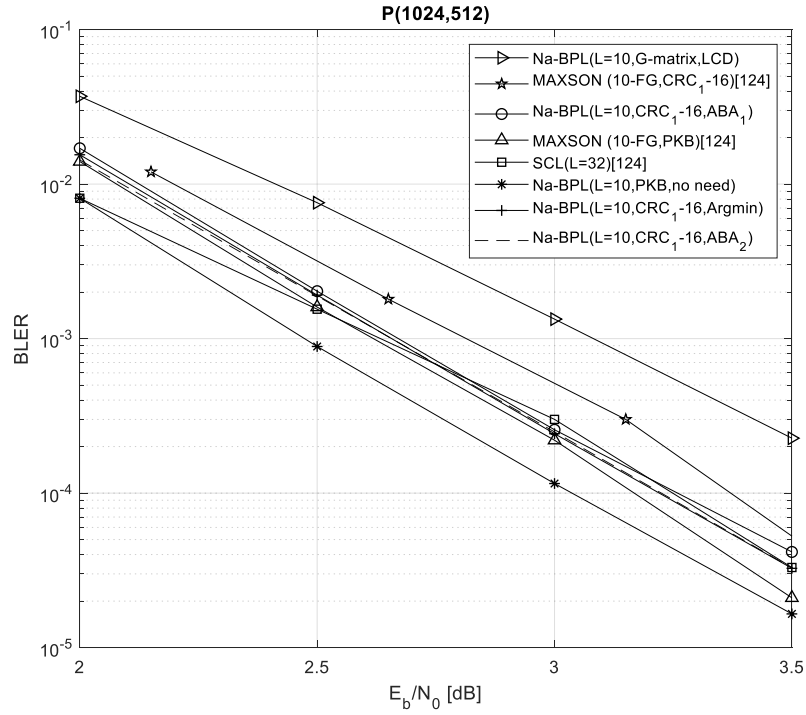
Figure 4.48 BLER performance comparison of different types of Na-BPL, MAXSON and SCL polar decoders for $\mathcal{P}(1024,512)$
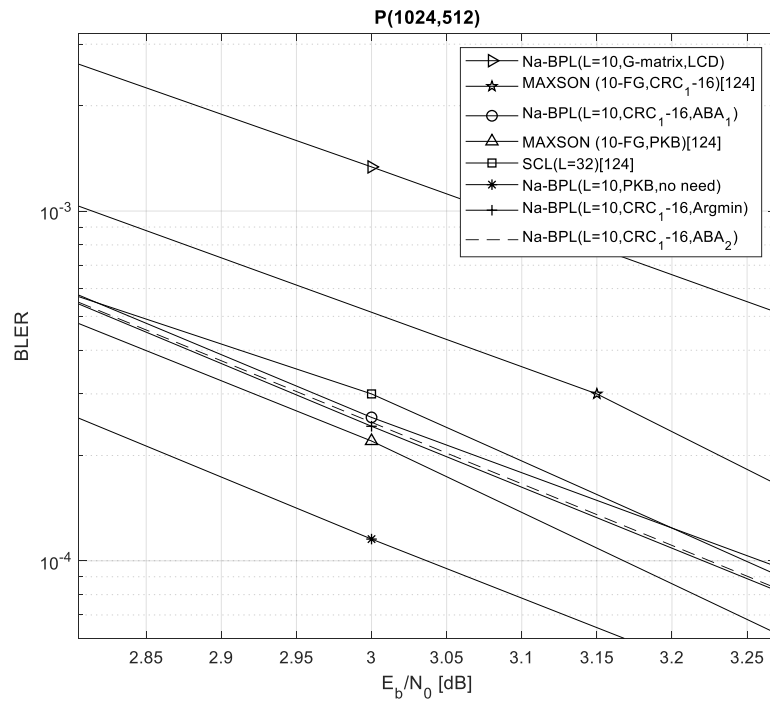
Figure 4.49 Zoomed BLER performance comparison of different types of Na-BPL, MAXSON and SCL polar decoders for $\mathcal{P}(1024,512)$

Simulation results in Fig. 4.48 and Fig. 4.49 show that Na-BPL decoder employing the same simulation parameters as MAXSON methods has better error correction performance. Even, BLER of Na-BPL (L=10, CRC_1-16, Argmin) is very close to MAXSON (10-FG, PKB). Moreover, the performance of SCL decoder with list size 32 is achieved. Further comparison can be accomplished using other studies on multi-trellis BP polar decoder's outcomes. In this manner, we are going to use the studies of [10,60,62] on polar code $\mathcal{P}$(2048,1024). In Fig. 4.50, four different decoder results are shown. First of them is multi-trellis approach with CRC based early detection method [60]. In this approach, 100 different permuted factor graphs (FG) are used in a sequential manner until CRC is satisfied. BPL [62] and SCL [10] results are also given. The proposed Na-BPL decoder utilizes half-way scheduling, maximum iteration number of 200, and list size of 32. Artificial noise intensities ranges from 0 to 0.1938 with increment step size of 0.00625. Early detection, in this case it is CRC, is activated after 20th iteration.
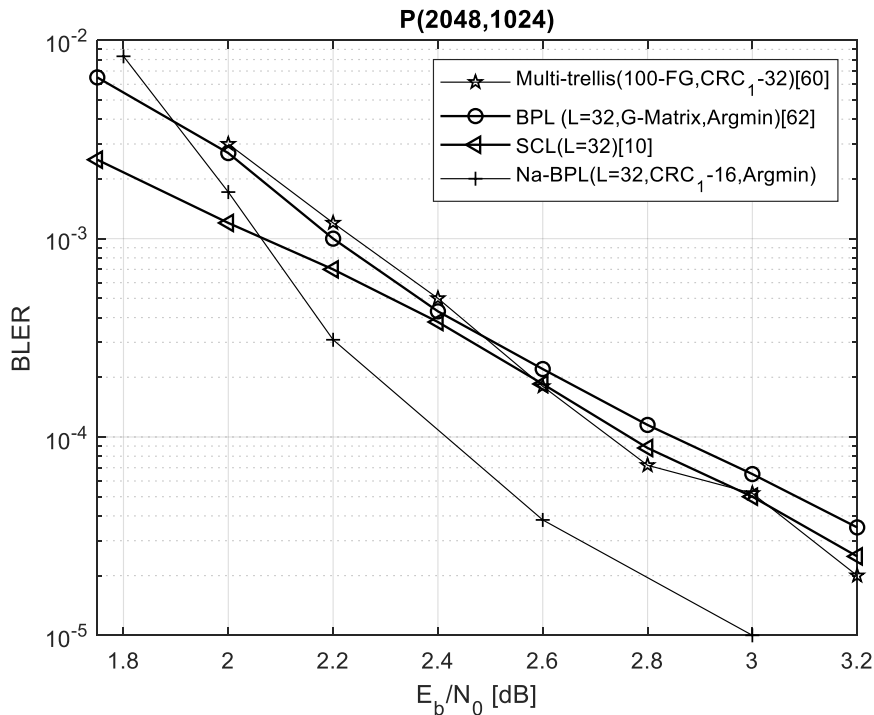
Figure 4.50 BLER performance comparison of different types of polar decoders for $\mathcal{P}$(2048,1024)

As a result, Na-BPL decoder with list size 32 achieves 0.3 dB advantage at 10^{-4} BLER over other polar decoders shown in Fig. 4.50.

GenAlg based polar code construction is also applied for SMS BP polar decoder. GenAlg is employed for polar codes $\mathcal{P}(512,256)$ and $\mathcal{P}(1024,512)$ on BPSK modulated AWGN channel with channel SNR 2 dB. Maximum iteration of SMS BP polar decoder is set to 200. Moreover, **G**-matrix-based early detection, keeping frozen nodes constant and conventional scheduling strategies are used during GenAlg based polar code construction. The flowchart shown in Fig. 4.25 is used. It is important to state that GenAlg uses single BP polar decoder, however, we use its results (choice of frozen set) for our Na-BPL structure and achieve significant error correction performance as shown in Fig. 4.51. Moreover, despite to the fact that GenAlg is run for **G**-matrix-based early detection, it achieves better results even with CRC based early detection. On the other hand, performance of polar code that is designed by using GenAlg for 2 dB SNR, degrades after 3 dB SNR and closes to the performance of polar code with Bhattacharyya based construction which is designed for 0.5 dB.
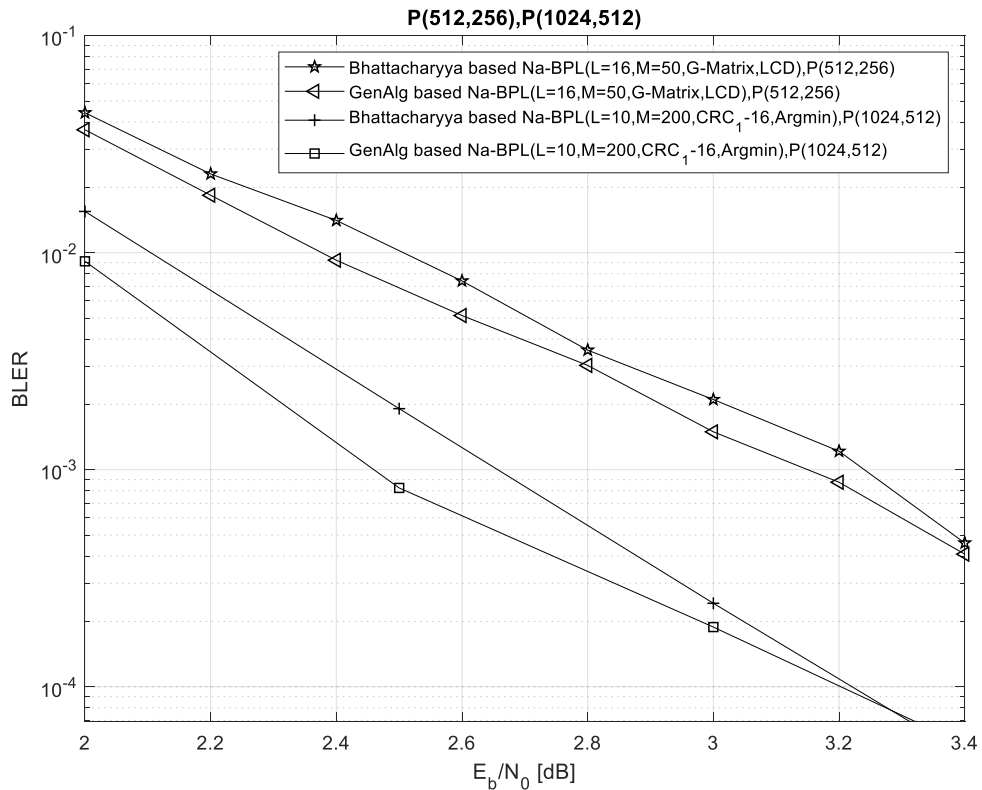
Figure 4.51 Genetic algorithm based polar code construction comparison for $\mathcal{P}(512,256)$ and $\mathcal{P}(1024,512)$

114

In [92] Na-BPL polar decoder with PKB early detection, which is one of the best BP-based decoder having BER/BLER performance close to the performance of the state-of-the-art CRC-aided SCL decoder, is introduced. Operation of Na-BPL is simple and it is similar to BPL of [62] in the way of using list concept. Na-BPL decoder diagram is depicted in Fig. 4.30 where it is seen that there are L parallel decoders, and each parallel branch is fed with artificially generated noise, **n**, and after each decoding operation an early detection and termination criteria is applied. A post decision is applied among the branches that satisfies early detection condition. It is also important to state that the factor graphs of the polar decoders are the same of each other (not permuted versions as utilized in [62]), and artificial noise level increases as the list size increases. Na-BPL with list size 32, maximum iteration number $M = 200$, utilizing realistic early detection techniques and post decision mechanisms has a BLER performance advantage of 0.3 dB when compared to SCL and BPL decoders employing $P(2048,1024)$. Moreover, further improvement is achieved applying new polar code construction method. In brief, the proposed approach is a promising candidate to substitute CRC-aided SCL decoder, which is used in 5G physical layer as the forward error correction method of control channels.

CHAPTER 5

CONCLUSION

Channel codes are used to overcome the degrading effects of channel noise and maintain reliable communication. Polar codes are chosen for control channels of eMBB communication services in 5G standard, and they are candidate for data and control channels of mMTC and URLLC. It is anticipated that polar coding will be a promising technique for the future communication standards. Belief propagation decoders employed for polar codes have parallel decoding capability, and they can be integrated with other decoders processing soft information.

In this thesis, different types of BP polar decoders are inspected. The design of BP polar decoders show difference considering throughput, latency, complexity and BER or BLER performance issues. Although BP polar decoder has parallel processing advantage, it has two main disadvantages, which are lower BER/BLER performance over CRC aided SCL, and higher complexity over SC. For this reason, to further enhance the performance of BP decoder; SMS BP decoder, parity-check matrix based decoder, modified BP polar decoder with check nodes, concatenated polar decoder, hybrid decoders and multi-trellis BP decoders are presented in the literature. Besides, improved BP decoder with modified kernel matrix, node classification and unification-based decoder, stage combined decoders and stochastic BP polar decoder are presented to decrease the complexity of the BP scheme without any performance loss. A variety of early detection and termination methods to decrease the iteration number of BP decoders are also studied in the literature. Decoding latency decrement and throughput increment are possible if an early detection method is performed when decoder is converged. Another topic of interest is the selection of frozen bits in polar code design process. As mentioned before, Bhattacharyya parameter based selection is designed for SC decoding scheme, and it is shown that it is not the

optimal method when other decoding schemes, and scheduling algorithms are used. A number of polar code construction methods are presented to handle the need for an optimal model for BP-based polar decoder. To sum up, if performance improvement of BP over CRC aided SCL is achieved with acceptable complexity, then polar codes will be subject to 5G and 6G framework standards that needs low latency, high reliability and low complexity.

In belief propagation algorithm, messages, i.e., probabilities, propagate forward and backward in an iterative manner, and in these propagations some messages may enter into deadlocks, and the reasons for these deadlocks are the initial unreliable messages obtained from the received signal. To prevent the appearance of deadlocks, or at least to decrease its probability of occurrence, we can use the weak noise for the received signal as a perturbation factor. Motivating from this fact, we consider the use of weak noise for the belief propagation decoders used for polar codes. The proposed decoder used for polar codes is called as noise aided belief propagation list decoder, i.e., Na-BPL. It is shown via simulation results that the proposed decoder improves the performance of polar codes. It is well-known in the literature that the systematic polar codes outperform non-systematic polar codes with additional complexity increment at the decoder side. However, due to the structure of belief propagation decoders, the systematic polar codes do not bring extra overhead at the decoder side. Motivating from this fact, we employed systematic polar codes for Na-BPL decoders to achieve better performance. The design strategy of the polar code is a very critical issue for the performance of the polar decoder. We considered the design of polar codes using the genetic algorithm. It is shown via simulation results that the polar codes designed by genetic algorithm show improved performance for Na-BPL decoders. The Na-BPL decoder employing perfect knowledge based early detection with polar code designed by genetic algorithm has a performance achievement only 0.1 dB away from the performance of the state-of-the-art polar decoder. On the other hand, Na-BPL decoder with realistic early detection and post decision mechanisms cannot compete with the state-of-the-art polar decoder of CRC aided SCL. However, its performance is still much better than the performance of SCL and BPL polar decoders. For future work, the controlled addition of weak noise to the received signals, and to the propagating messages showing unreliability can be a forthcoming study.

REFERENCES

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

2. **Arıkan E., (2009),** "*Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels*", IEEE Trans. Inf. Theory, vol. 55, pp. 3051 – 3073.

3. **3GPP, (2018**), "*Multiplexing and channel coding (Release 10) 3GPP TS 21.101 v10.4.0*", Oct. 2018.[Online]. Available:http://www.3gpp.org/ftp/Specs/2018-09/Rel-10/21-series/21101-a40.zip

4. **Hussami N., Korada S. B. and Urbanke R., (2009),** "*Performance of polar codes for channel and source coding*", 2009 IEEE International Symposium on Information Theory, Seoul, pp. 1488-1492.

5. **Çaycı S., (2013),** "*Lossless data compression with polar codes*", A Thesis Submitted to The Graduate School of Engineering and Science of Bilkent University, 62 pages.

6. **Önay S., (2014),** "*Polar codes for distributed source coding*", A Thesis Submitted to The Graduate School of Engineering and Science of Bilkent University, 170 pages.

7. **Arıkan E., (2010),** "*Polar codes: A pipelined implementation*", 4th Int. Symp. Broadband Commun. (ISBC 2010), pp. 2–4.

8. **Goela N., Korada S. B., and Gastpar M., (2010**), "*On LP decoding of polar codes*", in 2010 IEEE Information Theory Workshop, ITW 2010 - Proceedings.

9. **Poyraz E., Kim H., and Markarian G., (2009),** "*Performance of short polar codes under ML decoding*", in ICT-Mobile Summit 2009 Conference Proceedings. pp 1-6.

10. **I. Tal and A. Vardy, (2015),** "*List Decoding of Polar Codes*", IEEE Trans. Inf. Theory, vol. 61, no. 5, pp. 2213–2226.

11. **Forney G. D., (2001),** "*Codes on graphs: normal realizations*", in IEEE Transactions on Information Theory, vol. 47, no. 2, pp. 520-548, February.

12. **Balatsoukas-Stimming A., Giard P., and Burg A., (2017**), "*Comparison of Polar Decoders with Existing Low-Density Parity-Check and Turbo Decoders*", in 2017 IEEE Wireless Communications and Networking Conference Workshops, WCNCW 2017.

13. **Richardson T. and Kudekar S., (2018),** "*Design of Low-Density Parity Check Codes for 5G New Radio*", IEEE Communications Magazine, vol. 56, no. 3, pp. 28-34.

14. **Xiang L., Kaykac Egilmez Z. B., Maunder R. G. and Hanzo L., (2019**), "*CRC-Aided Logarithmic Stack Decoding of Polar Codes for Ultra Reliable Low Latency Communication in 3GPP New Radio*", in IEEE Access, vol. 7, pp. 28559-28573.

15. **Sharma A. and Salim M., (2017),** "*Polar Code: The Channel Code contender for 5G scenarios*", in 2017 International Conference on Computer, Communications and Electronics, pp. 676-682.

16. **Gallager R. G., (1962),** "*Low Density Parity Check Codes*", IRE Trans. Inf. Theory, vol. IT-8, pp. 21-28.

17. **Tanner R. M., (1981),** "*A Recursive Approach to Low Complexity Codes*", IEEE Trans. Inf. Theory, pp. 533 - 547.

18. **Fossorier M. P. C., Mihaljevic M., and Imai H., (1999),** "*Reduced complexity iterative decoding of low-density parity check codes based on belief propagation*", IEEE Trans. Commun., pp. 673 – 680.

19. **Yuan B. and Parhi K. K., (2013),** "*Architecture optimizations for BP polar decoders*", in ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing – Proceedings, pp. 2654-2658.

20. **Yuan B. and Parhi K. K., (2014),** "*Early stopping criteria for energy-efficient low-latency belief-propagation polar code decoders*", IEEE Trans. Signal Process., pp. 6496 – 6506.

21. **Abbas S. M., Fan Y., Chen J., and Tsui C. Y., (2015),** "*Low complexity belief propagation polar code decoder*", in IEEE Workshop on Signal Processing Systems, SiPS: Design and Implementation, pp. 1-6.

22. **Sha J., Liu X., Wang Z., and Zeng X., (2015),** "*A memory efficient belief propagation decoder for polar codes,*" China Commun. pp. 34 - 41.

23. **Ren Y., Zhang C., Liu X., and You X., (2015**), "*Efficient early termination schemes for belief-propagation decoding of polar codes*", in Proceedings - 2015 IEEE 11th International Conference on ASIC, ASICON 2015.

24. **Sha J., Liu J. and Wang Z., (2016),** "*Improved BP decoder for polar codes based on a modified kernel matrix*", in Electronics Letters, vol. 52, no. 24, pp. 1982-1984.

25. **Abbas S. M., Fan Y. Z., Chen J., and Tsui C. Y., (2017),** "*High-Throughput and Energy-Efficient Belief Propagation Polar Code Decoder*", IEEE Trans. Very Large Scale Integr. Syst. pp. 1098 -1111.

26. **Simsek C. and Turk K., (2017),** "*Hardware optimization for belief propagation polar code decoder with early stopping Criteria using high-speed parallel-prefix ling adder*", in 2017 40th International Conference on Telecommunications and Signal Processing, TSP 2017, pp. 182-185.

27. **Zhang Q., Liu A. and Tong X., (2017**), "*Early stopping criterion for belief propagation polar decoder based on frozen bits*", IET Electronics Letters, vol. 53, no. 24, pp. 1576-1578.

28. **Xiaojun Z. et al., (2018),** "*Polar code BP decoding method and device based on multi-stage updating processes*", C.N. Patent CN108039891 (A).

29. **Song H. Y., Kim J. H., Kim I. S. and Kim G. S., (2018),** "*Method and Apparatus for Approximated Belief Propagation Decoding of Polar Code*", K.R. Patent 101817168 (B1).

30. **Xu J., Che T., and Choi G., (2015),** "*XJ-BP: Express journey belief propagation decoding for polar codes*", in 2015 IEEE Global Communications Conference, GLOBECOM 2015, pp. 1-6.

31. **Li G., Mu J., Jiao X., Guo J., and Liu X., (2017),** "*Enhanced belief propagation decoding of polar codes by adapting the parity-check matrix*", Eurasip J. Wirel. Commun. Netw., pp. 1-9.

32. **Iqbal S., Hashmi A. A., and Choi G. S., (2017),** "*Improved Belief Propagation Decoding Algorithm for Short Polar Codes*", Wirel. Pers. Commun., vol. 96, pp. 1437–1449.

33. **Zhang Y., Liu A., Pan X., Ye Z., and Gong C., (2014**), "*A modified belief propagation polar decoder*", IEEE Commun. Lett., vol. 18, pp. 1091 - 1094.

34. **Sun C., Fei Z., Cao C., Wang X. and Jia D., (2019),** "*Low Complexity Polar Decoder for 5G Embb Control Channel*", in IEEE Access, vol. 7, pp. 50710-50717.

35. **Wang X., Zheng Z., Li J., Shan L. and Li Z., (2019**), "*Belief Propagation Bit-Strengthening Decoder for Polar Codes*", in IEEE Communications Letters, vol. 23, no. 11, pp. 1958-1961.

36. **Forney G. D., (1966),** "*Concatenated codes*", Massachusetts Institute Of Technology Research Laboratory Of Electronics.

37. **Bakshi M., Jaggi S., and Effros M., (2010),** "*Concatenated polar codes*", in IEEE International Symposium on Information Theory – Proceedings, pp. 918-922.

38. **Eslami A. and Pishro-Nik H., (2011),** "*A practical approach to polar codes*", in IEEE International Symposium on Information Theory – Proceedings, pp. 16-20.

39. **Eslami A. and Pishro-Nik H., (2013),** "*On finite-length performance of polar codes: Stopping sets, error floor, and concatenated design*", IEEE Trans. Commun., vol. 63, pp. 919-929.

40. **Dongliang X., W. Mingke, S. Na and M. Haibo, (2014),** "*Method for constructing Polar-LDPC concatenated codes*", C.N. Patent 103746708 (A).

41. **Mondelli M., Hassani S. H. and Urbanke R. L., (2016),** "*Unified Scaling of Polar Codes: Error Exponent, Scaling Exponent, Moderate Deviations, and Error Floors*", in IEEE Transactions on Information Theory, vol. 62, no. 12, pp. 6698-6712.

42. **Liu J., Jing S., You X. and Zhang C., (2017),** "*A merged BP decoding algorithm for polar-LDPC concatenated codes*", in 22nd International Conference on Digital Signal Processing (DSP), pp. 1-5.

43. **Zhang C., Liu J., Jing S. and You X., (2017),** "*Polar-LDPC cascaded code merging BP decoding algorithm and device*", C.N. Patent 107204780 (A), Sep. 26.

44. **Guo J., Qin M., Guillen I Fabregas A., and Siegel P. H., (2014),** "*Enhanced belief propagation decoding of polar codes through concatenation*", in IEEE International Symposium on Information Theory – Proceedings, pp. 2987-2991.

45. **Elkelesh A., Ebada M., Cammerer S., and Ten Brink S., (2016),** "*Improving Belief Propagation decoding of polar codes using scattered EXIT charts*", in 2016 IEEE Information Theory Workshop, ITW 2016.

46. **Yu Q. P., Shi Z. P., Deng L., and Li X., (2018),** "*An Improved Belief Propagation Decoding of Concatenated Polar Codes with Bit Mapping*", IEEE Commun. Lett., vol. 22, pp. 1160-1163.

47. **Abbas S. M., Fan Y., Chen J. and Tsui C., (2017),** "*Concatenated LDPC-Polar Codes Decoding Through Belief Propagation*", arXiv:1703.05542 [cs.IT], March.

48. **Presman N. and Litsyn S., (2012),** "*Recursive Descriptions of Decoding Algorithms and Hardware Architectures for Polar Codes*", arXiv:1209.4818 [cs.IT].

49. **Elkelesh A., Ebada M., Cammerer S., and Brink S., (2017),** "*Flexible Length Polar Codes through Graph Based Augmentation*", SCC 2017; 11th Int. ITG Conf. Syst. Commun. Coding, Hamburg, Ger., pp. 1–6.

50. **Zhang C., Jing S. and You X., (2017),** "*Joint detection decoding algorithm and apparatus for polar code-coded SCMA*", C.N. Patent 106941394 (A), Nov. 11.

51. **Yuan B. and Parhi K. K., (2015),** "*Algorithm and architecture for hybrid decoding of polar codes*", in Conference Record - Asilomar Conference on Signals, Systems and Computers, vol. 2015–April, pp. 2050–2053.

52. **Zhou X., Shen Y., Tan X., You X., Zhang Z. and Zhang C., (2018),** "*An Adjustable Hybrid SC-BP Polar Decoder*", 2018 IEEE Asia Pacific Conference on Circuits and Systems (APCCAS), Chengdu, pp. 211-214.

53. **Cammerer S., Leible B., Stahl M., Hoydis J., and Brink S., (2017),** "*Combining belief propagation and successive cancellation list decoding of polar codes on a GPU platform*", in ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings., pp. 3664-3668.

54. **Zhiwen P., Yongrun Y., Nan L. and Xiaohu Y., (2018),** "*Adaptive polar code decoding method* ", C.N. Patent 107659318 (A).

55. **Yang N., Jing S., Yu A., Liang X., Zhang Z., You X. and Zhang C., (2018),** "*Reconfigurable Decoder for LDPC and Polar Codes*", 2018 IEEE International Symposium on Circuits and Systems (ISCAS), Florence, pp. 1-5.

56. **Shusen J., Xiao L., Ningyuan Y., Xiaohu Y., Anlan Y. and Chuan Z., (2018),** "*Configuration method of BP decoder capable of simultaneously processing polar code and LDPC code*", C.N. Patent 108809329 (A).

57. **Arıkan E., (2008),** "*A performance comparison of polar codes and reed-muller codes*", IEEE Commun. Lett., vol. 12, no. 6, pp. 447–449.

58. **Peker A. G., (2018),** "*Belief Propagation Decoding of polar codes under factor graph permutations*", A Thesis Submitted to The Graduate School of Natural and Applied Sciences of Middle East Technical University, 106 pages.

59. **Elkelesh A., Cammerer S., Ebada M. and Brink S., (2017),** "*Mitigating clipping effects on error floors under belief propagation decoding of polar codes*", in

Proceedings of the International Symposium on Wireless Communication Systems, vol. 2017–August, pp. 384–389.

60. **Elkelesh A., Ebada M., Cammerer S. and Brink S., (2018),** "*Belief propagation decoding of polar codes on permuted factor graphs*", in IEEE Wireless Communications and Networking Conference, WCNC.

61. **Doan N., Hashemi A., Mondelli M. and Gross W. J., (2018),** "*On the Decoding of Polar Codes on Permuted Factor Graphs*", arXiv:1806.11195 [cs.IT].

62. **Elkelesh A., Ebada M., Cammerer S., Brink S. and Bler B. E. R., (2018),** "*Belief Propagation List Decoding of Polar Codes*", IEEE Commun. Lett., vol. 22, pp. 1536-1539.

63. **Akdoğan Ş. C., (2018),** "*A study on the set choice of multiple factor graph belief propagation decoders for polar codes*", A Thesis Submitted to The Graduate School of Natural and Applied Sciences of Middle East Technical University, 111 pages.

64. **Nachmani E., Beery Y. and Burshtein D., (2016), "***Learning to decode linear codes using deep learning*", 2016 54th Annual Allerton Conference on Communication, Control, and Computing (Allerton), Monticello, IL, pp. 341-346.

65. **Lugosch L. and Gross W. J., (2017),** "*Neural offset min-sum decoding*", 2017 IEEE International Symposium on Information Theory (ISIT), Aachen, pp. 1361-1365.

66. **Gruber T., Cammerer S., Hoydis J. and Brink S., (2017),** "*On deep learning-based channel decoding*", 2017 51st Annual Conference on Information Sciences and Systems (CISS), Baltimore, pp. 1-6.

67. **Cammerer S., Gruber T., Hoydis J. and Brink S., (2017**), "*Scaling Deep Learning-Based Decoding of Polar Codes via Partitionin*", GLOBECOM 2017 - 2017 IEEE Global Communications Conference, Singapore, pp. 1-6.

68. **Doan N., Hashemi S. A. and Gross W. J., (2018),** "*Neural Successive Cancellation Decoding of Polar Codes*", 2018 IEEE 19th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC), Kalamata, pp. 1-5.

69. **Xu W., Wu Z., Ueng Y., You X. and Zhang C., (2017),** "*Improved polar decoder based on deep learning*", 2017 IEEE International Workshop on Signal Processing Systems (SiPS), Lorient, pp. 1-6.

70. **Nghia D., Ali H. S., Elie N. M., Thibaud T. and Warren J. G., (2018),** "*Neural Belief Propagation Decoding of CRC-Polar Concatenated Codes*", arXiv:1811.00124v1 [cs.IT].

71. **Yuan C., Wu C., Cheng D. and Yang Y., (2018),** "*Deep Learning in Encoding and Decoding of Polar Codes*", IOP Conf. Series: Journal of Physics: Conf. Series 1060.

72. **Wang Y., Zhang Z., Zhang S., Cao S. and Xu S., (2018),** "*A Unified Deep Learning Based Polar-LDPC Decoder for 5G Communication Systems*", 2018 10th International Conference on Wireless Communications and Signal Processing (WCSP), Hangzhou, pp. 1-6.

73. **Xie G., Luo Y., Chen Y. and Ling X., (2019),** "*Belief Propagation Decoding of Polar Codes using Intelligent Post-processing*", 2019 IEEE 3rd Information Technology,

Networking, Electronic and Automation Control Conference (ITNEC), Chengdu, China, pp. 871-874.

74. **Cammerer S., Ebada M., Elkelesh A. and Brink S., (2018),** "*Sparse Graphs for Belief Propagation Decoding of Polar Codes*", 2018 IEEE International Symposium on Information Theory (ISIT), Vail, CO, pp. 1465-1469.

75. **Orlitsky A., Urbanke R., Viswanathan K. and Zhang J., (2002),** "*Stopping sets and the girth of Tanner graphs*", Proceedings IEEE International Symposium on Information Theory, Lausanne, pp. 2.

76. **Zhang Y., Zhang Q., Pan X., Ye Z., and Gong C., (2014),** "*A simplified belief propagation decoder for polar codes*", in 2014 IEEE International Wireless Symposium, pp. 2–5.

77. **Sha J., Lin J., and Wang Z., (2016),** "*Stage-combined belief propagation decoding of polar codes*", in Proceedings - IEEE International Symposium on Circuits and Systems, vol. 2016–July, pp. 421–424.

78. **Yuan B. and Parhi K. K., (2016),** "*Belief propagation decoding of polar codes using stochastic computing*", in Proceedings - IEEE International Symposium on Circuits and Systems, vol. 2016–July, no. 1, pp. 157–160.

79. **Xu M., Liang X., Zhang C., Wu Z., and You X., (2016),** "*Stochastic BP Polar Decoding and Architecture with Efficient Re-Randomization and Directive Register*", in 2016 IEEE International Workshop on Signal Processing Systems (SiPS), pp. 315–320.

80. **Han K., Wang J. and Gross W. J., (2018),** "*Bit-Wise Iterative Decoding of Polar Codes using Stochastic Computing*", in Proceedings of the 2018 on Great Lakes Symposium on VLSI (GLSVLSI '18), pp. 409-414.

81. **Han K., Wang J., Gross W. J. and Hu J., (2019),** "*Stochastic Bit-Wise Iterative Decoding of Polar Codes*", in IEEE Transactions on Signal Processing, vol. 67, no. 5, pp. 1138-1151.

82. **Ho L. M. and Guo Y., (2013),** "*Decoding Method Using Polar Code Sequence*", K.R. Patent 20130001494 (A).

83. **Park Y. S., Tao Y., Sun S., and Zhang Z., (2014),** "*A 4.68Gb/s belief propagation polar decoder with bit-splitting register file*", IEEE Symp. VLSI Circuits, Dig. Tech. Pap., pp. 2013–2014.

84. **Simsek C. and Turk K., (2016),** "*Simplified Early Stopping Criterion for Belief-Propagation Polar Code Decoders*", IEEE Commun. Lett., vol. 20, no. 8, pp. 1515–1518.

85. **Chuan Z., Yuanrui R. and Xiaohu Y., (2016),** "*Polar code BP decoding method with iterative early-stopping mechanism*", C.N. Patent 105262494 (A), Jan. 20.

86. **S Zhuo L., Lijuan X. and Junqi L., (2015),** "*Method for improving BP (belief propagation) decoding by use of polarisation code based on early termination of iterative strategy*", C.N. Patent 104539296 (A), Apr. 22.

87. **Kschischang F. R. and Frey B. J., (1998),** "*Iterative decoding of compound codes by probability propagation in graphical models*", IEEE J. Sel. Areas Commun., vol. 16, no. 2, pp. 219–230.

88. **Sun S. and Zhang Z., (2016**), "*Architecture and optimization of high-throughput belief propagation decoding of polar codes*", in Proceedings - IEEE International Symposium on Circuits and Systems, vol. 2016–July, pp. 165–168.

89. **Fayyaz U. U. and Barry J. R., (2014),** "*Low-Complexity Soft-Output Decoding of Polar Codes*", IEEE J. Sel. AREAS Commun. vol. 32, pp. 958-966.

90. **Schnelling C., Amraue Y., and Schmeink A., (2018),** "*On iterative decoding of polar codes: Schedule-dependent performance and constructions*", in 55th Annual Allerton Conference on Communication, Control, and Computing, Allerton 2017, 2018, vol. 2018–Janua, pp. 557–564.

91. **Qin M., Guo J., Bhatia A., Fabregas A. G., and Siegel P. H., (2017),** "*Polar Code Constructions Based on LLR Evolution*", IEEE Commun. Lett., vol. 21, no. 6, pp. 1221–1224.

92. **Arlı A. Ç. and Gazi O., (2019),** "*Noise-Aided Belief Propagation List Decoding of Polar Codes*", in IEEE Communications Letters, vol. 23, no. 8, pp. 1285-1288.

93. **Yuan B. and Parhi K. K., (2014),** "*Architectures for polar BP decoders using folding,*" in 2014 IEEE International Symposium on Circuits and Systems (ISCAS), pp. 205–208.

94. **Matthias K., Andrew B. and Sung P. Y., (2015),** "*Frozen-Bit Selection for a Polar Code Decoder*", U.S. Patent 2015333775 (A1), Nov. 19.

95. **Jingbo L. and Sha. J., (2017),** "*Frozen bits selection for polar codes based on simulation and BP decoding*", IEICE Electronics Express. 14. 10.1587/elex.14.20170026.

96. **Sun S., Member S., and Zhang Z., (2017),** "*Designing Practical Polar Codes Using Simulation-Based Bit Selection*", IEEE Journal on Emerging and Selected Topics in Circuits and Systems, vol. 7, no. 4, pp. 594–603.

97. **Bioglio V., Gabry F., Land I. and Belfiore J.-C., (2017),** "*Minimum-Distance Based Construction of Multi-Kernel Polar Codes*", in GLOBECOM 2017 - 2017 IEEE Global Communications Conference, pp. 1–6.

98. **Schwartz M. and Vardy A., (2006),** "*On the Stopping Distance and the Stopping Redundancy of Codes*", IEEE Trans. Inf. Theory, vol. 52, no. 3, pp. 922– 932.

99. **Elkelesh A., Ebada M., Cammerer S. and Brink S., (2019),** "*Genetic Algorithm-based Polar Code Construction for the AWGN Channel*", arXiv:1901.06444 [cs.IT], Jan. 2019.

100. **Di C., Proietti D., Telatar I. E., Richardson T. J. and Urbanke R. L., (2002),** "*Finite-length analysis of low-density parity-check codes on the binary erasure channel*", IEEE Trans. Inf. Theory, vol. 48, no. 6, pp. 1570–1579.

101. **Eslami A. and Pishro-Nik H., (2010),** "*On bit error rate performance of polar codes in finite regime*", in 2010 48th Annual Allerton Conference on Communication, Control, and Computing (Allerton), pp. 188–194.

102. **Sun S., Cho S. G. and Zhang Z., (2017),** "*Error patterns in belief propagation decoding of polar codes and their mitigation methods*", in Conference Record - Asilomar Conference on Signals, Systems and Computers, pp. 1199–1203.

103. **Sun S., Cho S. G. and Zhang Z., (2018),** "*Post-Processing Methods for Improving Coding Gain in Belief Propagation Decoding of Polar Codes*", in 2017 IEEE Global Communications Conference, GLOBECOM 2017 - Proceedings, vol. 2018–Jan., pp. 1–6.

104. **Pamuk A., (2011),** "*An FPGA implementation architecture for decoding of polar codes*", in Proceedings of the International Symposium on Wireless Communication Systems, pp. 437–441.

105. **Yang J., Zhang C., Zhou H. and You X., (2016),** "*Pipelined belief propagation polar decoders*", in Proceedings - IEEE International Symposium on Circuits and Systems, vol. 2016–July, no. 4, pp. 413–416.

106. **Reddy B. K. and Chandrachoodan N., (2012),** "*A GPU implementation of belief propagation decoder for polar codes*", Conf. Rec. - Asilomar Conf. Signals, Syst. Comput., pp. 1272–1276.

107. **Lin J., Sha J., Li L., Xiong C., Yan Z. and Wang Z., (2016),** "*A high throughput belief propagation decoder architecture for polar codes*", in 2016 IEEE International Symposium on Circuits and Systems (ISCAS), pp. 153–156.

108. **Ren A., Yuan B. and Wang Y., (2016),** "*Design of high-speed low-power polar BP decoder using emerging technologies*", in 2016 29th IEEE International System-on-Chip Conference (SOCC), 2016, pp. 312–316.

109. **Cui J., Dong Y., Li J., Zhang X. and Qingtian Z., (2018),** "*Simplified Early Stopping Criterion for Polar Codes*", 2018 IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC), Shandong, pp. 1-4.

110. **Sutera A., (1980),** "*Stochastic perturbation of a pure convective motion*", Journal of the Atmospheric Sciences, vol. 37, pp. 245-249.

111. **Niu K. and Chen K., (2012),** "*Stack decoding of polar codes*", in Electronics Letters, vol. 48, no. 12, pp. 695 -697, June.

112. **Arıkan E., (2011),** "*Systematic Polar Coding*", in IEEE Communications Letters, vol. 15, no. 8, pp. 860-862.

113. **Arıkan E., (2012),** "*Method and system for error correction in transmitting data using low complexity systematic encoder*", U.S. Patent US8347186B1.

114. **Mori R. and Tanaka T., (2009),** "*Performance of Polar Codes with the Construction using Density Evolution*", IEEE Commun. Lett., vol. 13, no. 7, pp. 519–521, July.

115. **Trifonov P., (2012),** "*Efficient Design and Decoding of Polar Codes*", IEEE Trans. Commun., vol. 60, no. 11, pp. 3221–3227, Nov.

116. **Li H. and Yuan J., (2013**), "*A practical construction method for polar codes in AWGN channels*", IEEE 2013 Tencon - Spring, Sydney, NSW, pp. 223-226.

117. **Holland J. H., (1992),** "*Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence*", Cambridge, MA, USA: MIT Press.

118. **Knoll C., Rath M., Tschiatschek S. and Pernkopf F**., **(2015),** "*Message Scheduling Methods for Belief Propagation*", In: Appice A., Rodrigues P., Santos Costa V., Gama J., Jorge A., Soares C. (eds) Machine Learning and Knowledge Discovery in Databases. ECML PKDD 2015. Lecture Notes in Computer Science, vol 9285. Springer, Cham.

119. **Elias P., (1991),** "*Error-correcting codes for list decoding*", in IEEE Transactions on Information Theory,vol. 37, no. 1, pp. 5-12, Jan.

120. **Koetter R., Ma J. and Vardy A., (2011),** "*The Re-Encoding Transformation in Algebraic List Decoding of Reed–Solomon Codes*", in IEEE Transactions on Information Theory, vol. 57, no. 2, pp. 633-647, Feb.

121. **Vangala H., Viterbo E. and Hong Y., (2015),** "*A Comparative Study of Polar Code Constructions for the AWGN Channel*", arXiv:1501.02473 [cs.IT], Jan..

122. **Liu Z., Liu R., Yan Z. and Zhao L., (2019),** "*GPU-based Implementation of Belief Propagation Decoding for Polar Codes*", ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brighton, United Kingdom, pp. 1513-1517.

123. **Y. Yan, X. Zhang and B. Wu, (2019),** "*Simplified Early Stopping Criterion for Belief-Propagation Polar Code Decoder Based on Frozen Bits*", in IEEE Access, vol. 7, pp. 134691-134696.

124. **Tosun B., (2019), "***Belief Propagation Decoding Using Factor Graph Permutations*", A Thesis Submitted to The Graduate School of Natural and Applied Sciences of Middle East Technical University, 88 pages.

125. **Geiselhart M., Elkelesh A., Ebada M., Cammerer S. and Brink S., (2020),** "*CRC-Aided Belief Propagation List Decoding of Polar Codes*", arXiv:2001.05303.

126. **Alrtaimi A. A. A. and Gazi O., (2020),** "*High Performance Low Latency Parallel Successive Cancelation Decoder Structures*", under submission.

<div align="center">**CURRICULUM VITAE**</div>

Ahmet Çağrı Arlı

E-mail : cagriarli@gmail.com
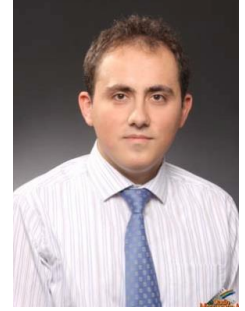
Personal Information

Marital Status : Married
Nationality : Turkish
Military Service : Completed
Birth Date : 03.10.1988

EDUCATION

Phd **Çankaya University** 3,83 / 4
09.2015 – 03.2020 Electronic and Communication Engineering
(Thesis: Soft Decision Based Decoding of Polar Codes)

Master **Çankaya University** 3,71 / 4
02.2013 - 08.2015 Electronic and Communication Engineering
(Thesis: Mathematical Modeling of Stochastic Resonance Systems)

Graduate **Çankaya University** 3,5 / 4
09.2006 - 07.2011 Electronic and Communication Engineering

WORK EXPERIENCE

04.2014-Continues   **Çankaya University**
Research Assistant

Within the scope of the education activities of the Department of Electronics and Communication Engineering of Çankaya University, I am teaching laboratory courses. In the laboratory departments of digital electronics courses, I teach electronic circuit setup, simulation in "Proteus" and "ORCAD" programs, programming on PIC microcontroller and programming on FPGA cards (VHDL). My academic studies include the work of accelerating Polar decoders in the field of coding theory and its implementation on FPGAs.

08.2013-   **GATE Elektronik**
04.2014   R&D Engineer

As part of the project, I did VHDL and Verilog programming on FPGA, circuit design and drawing in Altium Designer, and interface with C # .net on Visual Studio.

06.2012-   **ELOPSIS**
06.2013   R&D Engineer

I have gained experience in using CMOS imaging sensors. In addition, I worked on the design of embedded systems for ARM based processors, VHDL and Verilog programming on FPGAs.

Assisted Courses (Çankaya University)

- ECE 223 Digital Design I
- ECE 336 Microprocessors I
- ECE 428 Microprocessors II
- ECE 477 Advanced Digital VLSI System Design with VHDL
- ECE 478 VHDL Design of Electronic Device Interfaces
- ECE 494 System on Chip Design

Experience on Programming and Design Tools

-Quartus/ISE/Vivado Design Suite (Using VHDL/Verilog languages-9 Years)

-Altium Designer (Mostly digital designs-4 Years)

-Visual Studio (C#.Net based interface design-4 Years)

-PIC C and LPCxpresso (No-OS based designs-4 Years)

-MATLAB (Mostly script based studies, Simulink and HDL coder based designs-10 Years)

Projects

-"Energy Harvesting from Traffic Light" (Consultant): Ministry of Science, Industry and Technology, 2013-2014

-"Design of an Electrical Density Gauge" (Project Manager): Ministry of Science, Industry and Technology, 2015-2017

-"Uçtan Uca yerli ve milli 5G Şebekesi" (Consultant):Tübitak, 2018-2019.

Publications

1. Papers (SCI)

1.1. **A. Çağrı Arlı and O. Gazi,** "Noise-Aided Belief Propagation List Decoding of Polar Codes," in IEEE Communications Letters, vol. 23, no. 8, pp. 1285-1288, Aug. 2019.

2. Conferences

2.1. **A.Ç. Arlı, A.Çolak, O.Gazi**, The Implementation of a Successive Cancellation Polar Decoder on Xilinx System Generator, The 24th IEEE International Conference on Electronics, Circuits and Systems (ICECS), December 2017.

2.2. **A.Ç. Arlı, O.Gazi**, Mathematical modeling of stochastic resonance systems, 26th Signal Processing and Communications Applications Conference (SIU), 2018.

2.3. **A.Ç. Arlı, O.Gazi**, Successive papr reduction of OFDM systems via rotation of optimally selected information symbols, 26th Signal Processing and Communications Applications Conference (SIU), 2018.

3. Patent Applications

3.1. High-Speed Decoder For Polar Codes (PCT/TR2017/050627)

3.2 Soft Successive Cancellation Algorithm For Polar Codes (PCT/TR2018/050083)

3.3 PAPR reduction of OFDM communication systems via rotation of optimally selected information symbols (TurkPatent2018/P18/0782)

3.4. Noise and Cyclic Redundancy Check aided List Decoding of Error Correcting Codes (PCT/TR2019/050787)

Awards&Achievements

1. Electronic and Communication Engineering B.Sc. graduated with high honor, (2011)

2. Scholarship from Çankaya University for Ph.D. Study, 2015-2020

3. Teknogirişim Project 2015 (Electrical Density Gauge) from Ministry of Science, Industry and Technology

4. Digilent Contest 2017, "Optimum Implementation of Improved Successive Cancellation Decoders for Polar Codes in FPGA Platform", Honorable Mentions (as Advisor), May 2017.

5. Digilent Contest 2017, "Faster Successive Cancellation Decoding 2 bit", Honorable Mentions (as Advisor), May 2017.

6. Digilent Contest 2018, "Wireless Transmission of Compressed Image by ASK Modulation", Honarable Mentions (as Advisor), May 2018.

7. TUBITAK, Scholarship Program for PhD Students (2211-C), 2018-2020

8. 10th International Exhibition of Inventions & 3rd World Invention and Innovation Forum (Foshan)**,** "High-Speed Decoder For Polar Codes", Silver Medal, September 2018.

9. SIIF - Silicon Valley International Invention Festival, "High-Speed Decoder For Polar Codes", Bronze Medal, June 2019.