

BELIEF PROPAGATION DECODING USING FACTOR GRAPH  
PERMUTATIONS

A THESIS SUBMITTED TO  
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES  
OF  
MIDDLE EAST TECHNICAL UNIVERSITY

BY

BERNA TOSUN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS  
FOR  
THE DEGREE OF MASTER OF SCIENCE  
IN  
ELECTRICAL AND ELECTRONICS ENGINEERING

SEPTEMBER 2019



Approval of the thesis:

**BELIEF PROPAGATION DECODING USING FACTOR GRAPH  
PERMUTATIONS**

submitted by **BERNA TOSUN** in partial fulfillment of the requirements for the degree  
of **Master of Science in Electrical and Electronics Engineering Department,**  
**Middle East Technical University** by,

Prof. Dr. Halil Kalıpçılar  
Dean, Graduate School of **Natural and Applied Sciences** \_\_\_\_\_

Prof. Dr. İlkey Ulusoy  
Head of Department, **Electrical and Electronics Engineering** \_\_\_\_\_

Assoc. Prof. Dr. Melek Diker Yücel  
Supervisor, **Electrical and Electronics Engineering, METU** \_\_\_\_\_

**Examining Committee Members:**

Prof. Dr. Yalçın Tanık  
Electrical and Electronics Engineering, METU \_\_\_\_\_

Assoc. Prof. Dr. Melek Diker Yücel  
Electrical and Electronics Engineering, METU \_\_\_\_\_

Prof. Dr. Erdal Arıkan  
Electrical and Electronics Engineering, Bilkent Uni. \_\_\_\_\_

Assist. Prof. Dr. Gökhan Muzaffer Güvensen  
Electrical and Electronics Engineering, METU \_\_\_\_\_

Assist. Prof. Dr. Barış Nakiboğlu  
Electrical and Electronics Engineering, METU \_\_\_\_\_

Date: 20.09.2019

**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, Surname: Berna Tosun

Signature:

## ABSTRACT

### BELIEF PROPAGATION DECODING USING FACTOR GRAPH PERMUTATIONS

Tosun, Berna  
Master of Science, Electrical and Electronics Engineering  
Supervisor: Assoc. Prof. Dr. Melek Diker Yücel

September 2019, 88 pages

Capacity-achieving polar codes, introduced by Arikan have attracted significant attention over a decade. The bottleneck in coding is the decoder structure that achieves good performance with low hardware implementation cost and high throughput. Unlike the successive cancellation decoder, belief propagation decoder that can be improved by decoding on multiple factor graphs, allows for parallel decoding. For a polar code of length  $N$ , there are  $(\log_2 N)! = n!$  different permutations of the layers in the factor graph. Multiple factor graph belief propagation decoders that employ  $n$  factor graphs have the complexity of  $O(N(\log N)^2)$ , and the choice of proper sets among  $n!$  factor graphs for performance optimization is a challenging topic that has not yet been fully explored.

In this thesis, belief propagation decoding performance of polar codes over the additive white Gaussian noise channel is studied, by using single or multiple factor graphs within the decoder. The performance gap between the best and worst single factor graph decoders is found; and for multiple factor graph decoders, it is shown that random choice of factor graphs is incompetent for long code lengths. Two set-choice methods, MaxSON and MaxofMax rules are suggested for multiple factor graph decoders with  $n$  elements, as an alternative to the cyclically shifted set of factor graphs.

Performance of proposed set-choice rules are compared with cyclic, random and two other multiple factor graph belief propagation decoders given in the literature, for different code lengths with  $6 \leq \log_2 N \leq 14$ .

Keywords: Polar Codes, Belief Propagation, Multiple Factor Graph Decoder

## ÖZ

### FAKTÖR DİYAGRAM PERMÜTASYONLARI KULLANARAK İNANÇ YAYILIMLI KOD ÇÖZME

Tosun, Berna  
Yüksek Lisans, Elektrik ve Elektronik Mühendisliği  
Tez Danışmanı: Doç. Dr. Melek Diker Yücel

Eylül 2019, 88 sayfa

Arıkan tarafından önerilmiş ve kanal kapasitesine ulaştığı kanıtlanmış olan kutupsal kodlar, on yıldan beri büyük ilgi toplamaktadır. Kodlamadaki en önemli sorun, iyi bir başarıımı, düşük donanım masrafı ve yüksek hızda sağlayabilecek bir kod çözücü yapısıdır. Çok faktör diyagram kullanarak başarıımı iyileştirilebilen inanç yayımlı kod çözücüsü, ardışık götürme kod çözücüsünün aksine, paralelleştirmeye olanak tanır.  $N$  uzunluğunda bir kutupsal koda ait faktör diyagramının iç kademeleri  $(\log_2 N)! = n!$  farklı şekilde değiştirilebilir.  $n$  faktör diyagramı kullanan inanç yayımlı kod çözücülerin karmaşıklığı,  $O(N(\log N)^2)$  büyüklüğündedir ve başarıımı eniyileştirmek için  $n!$  faktör diyagram arasından en uygun kümelerin seçimi, henüz tümüyle araştırılmamış, tetikleyici bir konudur.

Bu tezde, beyaz Gaussian gürültüsü eklenmiş kanalda, kutupsal kodların, tek veya çok faktör diyagramlı inanç yayımlı kod çözücüsü başarımları üzerinde çalışılmıştır. Tek faktör diyagramlı kod çözücülerin en iyi ve en kötöleri arasındaki başarımlar farkları bulunmuş; ayrıca çok faktör diyagramlı kod çözücülerde rassal küme seçiminin büyük  $n$ 'ler için yararsızlığı gösterilmiştir. Döngüsel kaymalarla elde edilen döngüsel kümeden farklı bir seçenek olarak, çok faktör diyagramlı kod çözücüler için MaxSON ve MaxofMax diye adlandırılan iki tane  $n$ -elemanlı küme seçim yöntemi önerilmiştir.

Önerilen küme seçim yöntemlerinin başarımları, döngüsel, rassal ve literatürde geçen iki diğer çok faktör diyagramlı inanç yayımlı kod çözücünün başarımlarıyla,  $6 \leq \log_2 N \leq 14$  eşitsizliğini sağlayan farklı kod boyları için karşılaştırılmıştır.

Anahtar Kelimeler: Kutupsal Kodlar, İnanç Yayılımı, Çok Faktör Diyagramlı Kod Çözücü

To my significant other

## **ACKNOWLEDGEMENTS**

I would like to express my gratitude to my supervisor Assoc. Prof. Dr. Melek Diker Yücel for her help, patience and sharing her knowledge.

I also would like to thank my betterhalf for his encouragement, support and endless love from the beginning to the end, all the time.

My close friends always supported me mentally and shared my stress. So, thanks to all of them.

More importantly, I would like to thank my family so much for always being there.

## TABLE OF CONTENTS

ABSTRACT .....	v
ÖZ .....	vii
ACKNOWLEDGEMENTS .....	x
TABLE OF CONTENTS .....	xi
LIST OF TABLES .....	xiii
LIST OF FIGURES .....	xiv
LIST OF ABBREVIATIONS .....	xix
CHAPTERS	
1. INTRODUCTION .....	1
1.1. Channel Coding .....	1
1.2. Overview of Polar Codes.....	3
1.3. Overview of Decoding Algorithms of Polar Code .....	5
1.4. Aim and Organization of the Thesis.....	8
2. POLAR CODES .....	11
2.1. Preliminaries.....	11
2.2. Polar Coding.....	14
2.2.1. Channel Combining .....	14
2.2.2. Channel Splitting .....	17
2.2.3. Channel Polarization.....	20
2.2.4. Polar Encoding.....	23
2.2.4.1. Factor Graph (FG) Representation of Polar Codes .....	24
2.2.4.2. Selecting Frozen Nodes in AWGN Channel.....	26

2.2.5. Belief Propagation Decoding Algorithm.....	30
3. SIMULATION RESULTS .....	35
3.1. Choice of the Required Number of iterations According to the Code Length	36
3.2. Single Factor Graph BP Decoding Performance of Polar Codes over AWGN .....	38
3.3. Multiple Factor Graph BP Decoding Performance of Polar Codes over the AWGN .....	45
3.3.1. Randomly Chosen Multiple Factor Graph BP Decoding Performance of Polar Codes over the AWGN .....	46
3.3.2. Predetermined Multiple Factor Graph BP Decoding Performance of Polar Codes over the AWGN.....	52
3.4. Performance Comparison of Multiple Factor Graph BP Decoding for Polar Codes and Reed-Muller Codes .....	59
3.5. Performance Comparison with Other Multiple Factor Graph BP Decoders over the AWGN .....	63
4. CONCLUSION .....	73
REFERENCES .....	77
APPENDICES .....	83
A. RFG over IRFG BP Decoder Gain for Polar Code Designed over Binary Erasure Channels .....	83
B. Some Considerations about Multiple-FG Set Choice .....	85
C. BLER Comparison of Some BP Decoders over the AWGN for Polar Code Constructions Using Fixed or Channel-Specific Design-SNRs .....	87

## LIST OF TABLES

### TABLES

<i>Table 2.1.</i> Algorithm based on the Bhattacharyya bounds .....	27
<i>Table 3.1.</i> BLER performances with changing $n$ and changing iteration number. ....	36
<i>Table 3.2.</i> Stage order numbers (SONs) and number of decoded words out of 10000 blocks for a sample set of random factor graphs (FGs) used in 7-FG decoders of $\mathcal{P}(128, 64)$ at SNR=2dB. ....	47

## LIST OF FIGURES

### FIGURES

<i>Figure 1.1.</i> Simplified communication system.....	1
<i>Figure 1.2.</i> Performance comparison of polar, turbo and LDPC codes (reproduced from [Niu et al. 2014]).....	3
<i>Figure 2.1.</i> Conditional probability distribution for BPSK modulation over AWGN channel.....	14
<i>Figure 2.2.</i> Channel combining for $W_2$ . .....	15
<i>Figure 2.3.</i> The relation of channel $W_4$ with $W_2$ and $W_1$ . .....	16
<i>Figure 2.4.</i> The construction of $W_N$ from two copies of $W_N/2$ . .....	17
<i>Figure 2.5.</i> The split channels $W_{21}$ and $W_{22}$ after channel splitting for $N = 2$ . ..	18
<i>Figure 2.6.</i> Channel splitting of $W_N$ into $N$ distinct channels $W_N(i)$ . .....	19
<i>Figure 2.7.</i> The constructed channel $W_8$ .....	24
<i>Figure 2.8.</i> Z-shape factor graph representation corresponding to the polar code generator matrix $G_8$ .....	24
<i>Figure 2.9.</i> All different FG representations for polar code with $N = 8$ . .....	26
<i>Figure 2.10.</i> Factor graph representation of $P(8, 4)$ polar code encoding for the information word .....	28
<i>Figure 2.11.</i> Factor graph representation of $P(8, 4)$ polar code encoding for the information word $[1, 1, 0, 1]$ .....	29
<i>Figure 2.12.</i> Diagram of single Z-shape processing element in the polar BP decoder. ....	31
<i>Figure 2.13.</i> Check (square) and variable (circle) nodes of $P(8, 4)$ on RFG representation.....	31
<i>Figure 3.1.</i> Required iteration numbers versus $n$ , where the code length $N = 2n$ . ..	38

*Figure 3.2.* Performances of RFG BP decoders for  $P(1024, 512)$  codes, constructed using i) fixed design-SNR of 0 dB (yellow), ii) channel-specific design-SNR at each SNR (red). .....40

*Figure 3.3.* BLER of the RFG decoder for polar codes with design-SNR of 0 dB, versus the channel SNR. From top to bottom - Blue:  $n = 6$ ,  $P(64, 32)$ , Black:  $n = 7$ ,  $P(128, 64)$ , Pink:  $n = 8$ ,  $P(256, 128)$ , Green:  $n = 9$ ,  $P(512, 256)$ , Yellow:  $n = 10$ ,  $P(1024, 512)$ , Purple:  $n = 11$ ,  $P(2048, 1024)$ , Red:  $n = 12$ ,  $P(4096, 2048)$ , Brown:  $n = 13$ ,  $P(8192, 4096)$ , Gray:  $n = 14$ ,  $P(16384, 8192)$ .....41

*Figure 3.4.* BLER performances of the RFG (solid) and the IRFG (dashed) BP decoders for  $n = 6$ ,  $P(64, 32)$ ,  $n = 7$ ,  $P(128, 64)$ ,  $n = 8$ ,  $P(256, 128)$ ,  $n = 9$ ,  $P(512, 256)$ ,  $n = 10$ ,  $P(1024, 512)$  and  $n = 11$ ,  $P(2048, 1024)$  codes, constructed using i) fixed design-SNR of 0 dB (yellow curves), ii) channel-specific design-SNR at each SNR (red curves).....42

*Figure 3.5.* BLER of the RFG and IRFG decoders versus channel SNR with design-SNR of 0 dB. Solid lines refer to RFG, and dotted ones of the same color refer to IRFG performances for the same code length. Blue:  $n = 6$ ,  $P(64, 32)$ , Black:  $n = 7$ ,  $P(128, 64)$ , Pink:  $n = 8$ ,  $P(256, 128)$ , Green:  $n = 9$ ,  $P(512, 256)$ , Yellow:  $n = 10$ ,  $P(1024, 512)$ , Purple:  $n = 11$ ,  $P(2048, 1024)$ , .....43

*Figure 3.6.* Gain of the RFG decoder over the IRFG decoder at  $\text{BLER} = 10^{-2}$  versus  $n$  for polar codes constructed using i) fixed design-SNR of 0 dB (yellow curve), ii) channel-specific design-SNR at each channel SNR (red curve). .....44

*Figure 3.7.* BLER performance of ten random 7-FG decoders for  $P(128, 64)$  at  $\text{SNR}=2\text{dB}$ , versus the number of factor graphs. ....47

*Figure 3.8.* BLER comparison of the cyclic 7-FG BP decoder with ten random 7-FG BP decoders for  $P(128, 64)$  at  $\text{SNR} = 2$  dB. ....48

*Figure 3.9.* BLER comparison of the cyclic 10-FG BP decoder with ten random 10-FG BP decoders for  $P(1024, 512)$  at  $\text{SNR} = 1.5$  dB. ....50

<i>Figure 3.10.</i> BLER of multiple-FG BP decoder for $P(1024, 512)$ at $SNR = 1.5$ dB versus the number of FGs, for cyclic 10-FG and ten random choices of 100-FG decoders. ....	50
<i>Figure 3.11.</i> BLERs of single-FG BP decoders for $P(64, 32)$ over the AWGN with $SNR = 1.5$ dB, versus their SONs sorted in ascending order (where circles indicate the selection of MaxSON, and triangles show the selection of MaxofMax sets).....	53
<i>Figure 3.12.</i> BLER performances of the cyclic shift, MaxSON and MaxofMax choices for the FG set of the $n$ -FG BP decoder for various code lengths $N = 2n=128, 256, 512, 1024, 2048$ and $4096$ ; where the AWGN channel SNRs are adjusted as $2, 1.8, 1.75, 1.7, 1.4$ and $1.35$ dB respectively.....	55
<i>Figure 3.13.</i> BLER performances of RFG as compared to $n$ -FG BP decoders chosen randomly or deterministically by cyclic, MaxSON and MaxofMax rules for $P(1024, 512)$ , $P(2048, 1024)$ , $P(4096, 2048)$ ; and by cyclic, MaxSON, MaxofMax rules for $P(8192, 4096)$ and $P(16384, 8192)$ . ....	57
<i>Figure 3.14.</i> SNR gains at $BLER = 10^{-3}$ of the MaxSON and MaxofMax $n$ -FG BP decoders over the cyclic $n$ -FG BP decoder versus $n$ . ....	59
<i>Figure 3.15.</i> BLER of the RFG decoder for polar (solid lines) and RM codes (dashed lines), versus the channel SNR, where red, green and purple curves are corresponding respectively to $(N,K) = (128, 64)$ , $(512, 256)$ , and $(2048, 1024)$ codes. ....	60
<i>Figure 3.16.</i> BLER performance comparison between the RFG and 120-FG BP decoders of RM codes and the RFG decoder of polar codes for $(N, K) = (512, 256)$ . ....	60
<i>Figure 3.17.</i> BLER performance comparison of multiple-FG BP decoders for the RM codes and adaptive polar codes for $(N, K) = (128, 64)$ over a $BEC(\epsilon)$ with $\epsilon = 0.35$ .. ....	61
<i>Figure 3.18.</i> BLER performance comparison of multiple-FG BP decoders for the RM codes and polar codes with parameters $(N, K) = (128, 64)$ over an AWGN at $SNR = 2$ dB. ....	62

*Figure 3.19.* BLER performance comparison between the BP decoders of our study (shown by dashed lines) and another study in the literature reproduced from [Doan et al., 2018]) (shown by solid lines); where our first three figures are obtained for  $P(1024, 512)$  constructed at fixed design-SNRs ( $-1.59, 0, 0.5$  dB) and last one is constructed with variable, channel-specific design-SNR.....64

*Figure 3.20.* BLER performance comparison for  $P(2048, 1024)$ , constructed at different design-SNRs, between the decoders of our study (shown by dashed lines) and another one in the literature (shown by solid lines, reproduced from [Elkelesh et al., 2018a] and [Elkelesh et al., 2018b]). .....66

*Figure 3.21.* BLER performance for  $P(1024, 512)$ , constructed at 0.5 dB design-SNR, of our RFG, cyclic and MaxSON decoders using perfect knowledge based (PKB) and CRC based early termination, (Doan’s curves are reproduced from [Doan et al., 2018]).....68

*Figure 3.22.* BLER performance for  $P(2048, 1024)$ , constructed at 0.5 dB design-SNR, of our RFG and MaxSON decoders using perfect knowledge based (PKB) and CRC based termination, (Elkelesh’s curves are reproduced from [Elkelesh et al., 2018a] and [Elkelesh et al., 2018b]).....69

*Figure 3.23.* BLER (solid lines) and BER(dashed lines) performances for  $P(1024, 512)$  and  $P(2048, 1024)$ , constructed at 0.5 dB design-SNR, between our RFG and MaxSON decoders using CRC16 based early termination. ....71

*Figure A.1.* BLER of the RFG and IRFG decoders versus the channel erasure rate. Solid lines refer to RFG and dotted ones of the same color refer to IRFG performances for the same code lengths.....84

*Figure A.2.* BLER performances of the RFG and IRFG BP decoders for  $P(1024, 512)$  codes, constructed using i) fixed design-erasure-rate of 0.5,.....84

*Figure B.1.* Effects of individual 1-FG performances of the FGs in the n-element BP decoder sets (formed either by cyclic shifts or by the MaxSON and MaxofMax rules), to the overall n-FG performance; for  $P(1024, 512)$  at SNR = 2 dB and  $P(2048, 1024)$  at SNR = 1.7 dB., .....86

*Figure C.1.* BLER performances of multiple-FG BP decoder performances for P(1024, 512) codes, constructed using either fixed design-SNR of 0 dB (solid curves), or channel-specific design-SNR (dashed curves), ..... 87

*Figure C.2.* BLER performances of multiple-FG BP decoder performances for P(2048, 1024) codes, constructed using either fixed design-SNR of 0 dB (solid curves), or channel-specific design-SNR (dashed curves). ..... 88

## LIST OF ABBREVIATIONS

### ABBREVIATIONS

AWGN	.....	Additive White Gaussian Noise
BDMC	.....	Binary-input Discrete Memoryless Channel
BEC	.....	Binary Erasure Channel
BER	.....	Bit Error Ratio
BLER	.....	Block Error Ratio
BP	.....	Belief Propagation
BPL	.....	Belief Propagation List
BPSK	.....	Binary Phase-Shift Keying
BSC	.....	Binary Symmetric Channel
CER	.....	Codeword Error Ratio
CRC	.....	Cyclic Redundancy Check
CS	.....	Capacity Sum
FG	.....	Factor Graph
FV	.....	Number of Frozen Variables
FPGA	.....	Field Programmable Gate Array
IRFG	.....	Inverse Reference Factor Graph
LDPC	.....	Low Density Parity Check
LLR	.....	Log Likelihood Ratio
<i>M</i> -FG	.....	Multiple Decoder That Uses <i>M</i> Factor Graphs
ML	.....	Maximum Likelihood
MS	.....	Min-Sum
PC	.....	Polar Codes
RFG	.....	Reference Factor Graph
RM	.....	Reed-Muller
SC	.....	Successive Cancellation
SCL	.....	Successive Cancellation List
SMS	.....	Scaled Min-Sum
SNR	.....	Signal to Noise Ratio
SON	.....	Stage Order Number
SSC	.....	Simplified Successive Cancellation



# CHAPTER 1

## INTRODUCTION

“The fundamental problem of communication is that of reproducing at one point either exactly or approximately a message selected at another point” [Shannon, 1948]. Main purpose of a communication system is transmitting the information to the receiver in an efficient and reliable way, in the presence of noise. To attain this objective, the common procedure is to add a structured redundancy to the data before transmission. The way of adding the redundancy is called channel coding. An illustration of a simplified communication system is shown in Figure 1.1.

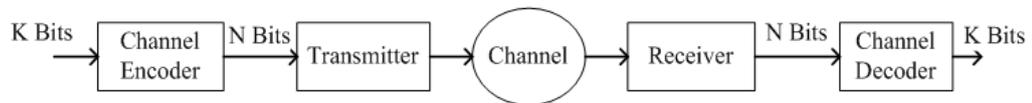


Figure 1.1. Simplified communication system.

### 1.1. Channel Coding

The field of channel coding began with Shannon’s information theory [Shannon, 1948], which tells us the amount of information that a channel can carry; in other words, the capacity of the channel. One of his main results is that data transmission is possible with arbitrarily small error probabilities, if the transmission rate is below or equal to the channel capacity, which can only be achieved asymptotically by coding schemes with codewords of infinite length. However, such an increase in codeword length has implications on the complexity of encoders and decoders. For the next half of the 20<sup>th</sup> century, the main objective has been to find practical coding schemes that approach the Shannon’s capacity limit.

The majority of practical channel codes have been developed in the early ages of coding theory, including the Golay codes [Golay, 1949], Hamming codes [Hamming, 1950], Reed Muller codes [Reed, 1954], [Muller, 1954], convolutional codes [Elias, 1955], Reed-Solomon codes [Reed and Solomon, 1960], Low-Density Parity-Check (LDPC) codes [Gallager, 1962], and turbo codes [Berrou & Glavieux, 1993].

Turbo codes are the first practically implemented codes that have performed close to Shannon's capacity limit. Because of their relatively low complexity, turbo codes have been the core of 3G/4G communication systems. With the technological developments and advancement of simpler decoder structures, the invention of the turbo codes have started a revolution that have caused the rediscovery of LDPC codes [MacKay and Neal, 1996] and they have been serious competitors to turbo codes in practical applications. Turbo codes and LDPC codes are then unified within the concept of "codes defined on graphs" by Wiberg [Wiberg, Loeliger & Kötter, 1995], [Wiberg, 1996] that is widely used in many applications since then.

With the introduction of polar codes, which are invented by Arikan [Arikan, 2008] a decade ago, one can achieve reliable data transmission with low computational complexity, at rates close to the capacity for any binary input discrete memoryless channel (BDMC) [Arikan, 2009]. Polar codes are commonly decoded by simple decoders like successive cancellation (SC) or belief propagation (BP), or by their more complicated versions such as the successive cancellation list (SCL), or the multiple factor graph BP decoders. The performance of polar codes in comparison with LDPC and turbo codes is presented in Figure 1.2. For all the coding schemes in the figure, the code length is  $N = 1024$  (only the LDPC code has  $N = 1056$ ) and the code rate is  $R = 1/2$ . Turbo codes in Figure 1.2 are encoded according to the WCDMA and LTE standards and LDPC codes are encoded according to the WiMax standard. One observes from Figure 1.2 that polar codes under the CRC aided decoding algorithms (i.e., CA-SCL(32) and aCA-SCL(1024) in the figure, with list sizes of 32 and 1024

respectively) outperform the turbo or LDPC codes by up to 0.7 dB at the BLER of  $10^{-4}$ . Because of their promising performance, we focus on the belief propagation decoders for polar codes in this thesis.

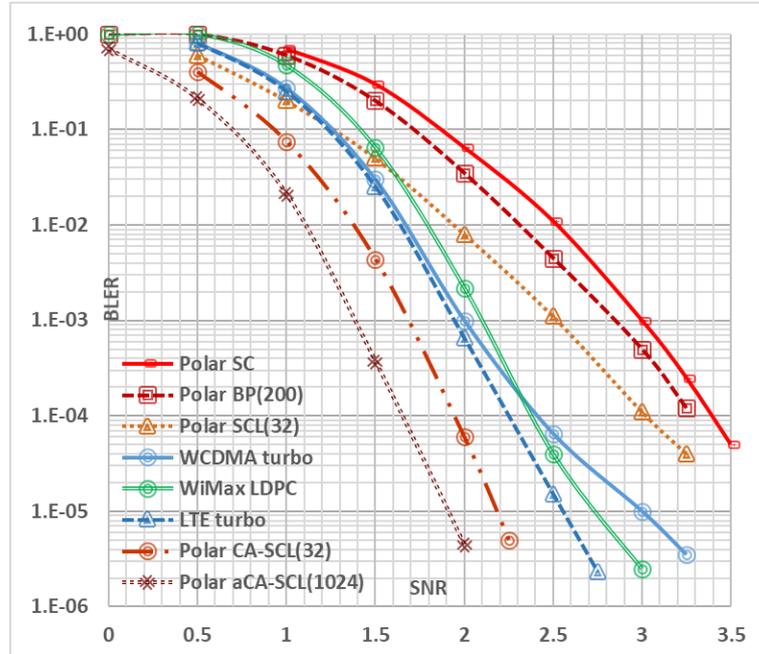


Figure 1.2. Performance comparison of rate-0.5 polar, turbo codes of length 1024 and LDPC code of length 1056 (reproduced from [Niu et al. 2014]).

## 1.2. Overview of Polar Codes

Polar codes are based on channel polarization, which transforms two independent binary-input discrete memoryless channels (BDMC) into two polarized channels that can be called “a good channel and a bad channel” with regard to their reliabilities. When channel polarization is recursively applied to the polarized channels of the previous step, the transformed channels start to dissociate so much after a number of steps, creating extreme channels such that the reliabilities of the good and bad channels differ remarkably. The good channels become almost noiseless, and the bad ones become very noisy channels.

The code construction of an  $(N, K)$  polar code with block length  $N = 2^n$  is based on choosing the  $K$  best channels among  $N$  polarized channels for sending the information bits, and freezing the remaining  $N - K$  channels, which have lower capacities. This process corresponds to mapping input bits into codewords by using a  $K \times N$  generator matrix, whose  $K$  rows are selected with respect to the reliabilities of the corresponding channels from the rows of  $F^{\otimes n}$ ; i.e., the  $n^{\text{th}}$ -Kronecker product of the base matrix  $F = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$ . Although both polar and Reed Muller (RM) codes use generator matrices constructed from the rows of  $F^{\otimes n}$ ; the design philosophy of polar codes is fundamentally different from that of the RM codes. Generator matrix of the RM code chooses the rows of  $F^{\otimes n}$ , which maximize the minimum distance of the code; whereas the polar code selects the paths with the highest channel capacities.

To construct the polar codes, Arikan uses the symmetric capacity  $I(W)$  and the Bhattacharyya parameter  $Z(W)$ , as measures of rate and reliability respectively. In his seminal study [Arikan, 2009], Arikan states that although “the code construction problem can be solved in principle by computing all Bhattacharyya parameters, there is unfortunately no efficient algorithm for doing this. One exception is the binary erasure channel for which  $Z(W_i)$  can all be calculated in time  $O(N)$ , thanks to the recursive formulas”. So, a wide range of approximate construction methods are proposed starting from early studies [Mori and Tanaka, 2009a], [Zhao et al., 2011], [Bonik et al., 2012], [Trifonov, 2012], [Li and Yuan, 2013], [Tal and Vardy, 2013]; and many algorithms have been devised for the additive white Gaussian noise (AWGN) channel case [Kern et al., 2014], [Wu et al., 2014 ], [Zhang et al., 2014 ]. Along with the estimation of Bhattacharyya parameters for which Arikan has suggested the Monte-Carlo approach [Arikan, 2009], the density evaluation [Mori and Tanaka, 2009a], [Tal and Vardy, 2013] and Gaussian approximation methods [Trifonov, 2012], [Li and Yuan, 2013] are also proposed.

In coding theory, most of the codes are *universal*. One of the drawbacks of the polar code construction is that polar codes are not universal. This means code construction depends on the respective channel parameter, e.g. the signal to noise ratio (SNR) for the additive white Gaussian noise (AWGN) channel, or the erasure probability ( $\epsilon$ ) for the binary erasure channel (BEC). A change in code construction with changing SNR is not desired; therefore, there are a few recent attempts to design universal polar codes [Sasoglu & Wang, 2014], [Hassani & Urbanke, 2014], [Alsan, 2014]. However, in return for universality, their designs require higher complexity at the decoder or encoder. On the other hand, Vangala et al. propose a simple search algorithm to find the best design-SNR and use it for a range of possible SNRs [Vangala, Viterbo & Hong, 2015]. They compare various polar code constructions and draw the conclusion that, the Bhattacharyya parameters computed with a design-SNR of 0 dB, works well for the AWGN.

### **1.3. Overview of Decoding Algorithms of Polar Code**

As the first polar decoder, Arkan proposed successive cancellation (SC) decoding algorithm that has a complexity of  $O(N \log_2 N)$ , in which all information bits are sequentially decided subject to the previously estimated bits and the channel information [Arkan, 2009]. The error performance of polar codes with the SC algorithm can be asymptotically optimum for infinitely long code lengths; however, it is worse than those of turbo or LDPC codes for short and moderate code lengths. To improve the performance of polar code decoders, successive cancellation list (SCL) [Tal and Vardy, 2011] decoding algorithm has been introduced, which achieves a performance comparable to that of the low-density parity-check (LDPC) codes. Another decoding algorithm, cyclic redundancy check (CRC) aided SCL (CA-SCL) decoding [Niu & Chen, 2012] has shown even better performance than turbo codes. Nevertheless, due to the serial processing nature of the SC, it suffers a high latency and limited throughput. With that specific aim to reduce the latency while increasing the throughput, some decoding algorithms, such as simplified successive cancellation

(SSC) [Yazdi & Kschischang, 2011], maximum likelihood SSC (ML-SSC) [Sarkis & Gross, 2013], and repetition single parity check ML-SSC (RSM-SSC) [Giard, Sarkis, Thibeault & Gross, 2014] have been proposed. Recent studies show that the SC bit-flip decoder (SCF) has similar BLER performance with the CA-SCL [Zhang, Qin, Zhang, Zhang, & Chen, 2017], [Zhang, Qin, Zhang, & Chen, 2018], [Chandesris, Savin & Declercq, 2018].

Apart from serial processing algorithms, some researchers investigate the usage of belief propagation (BP) decoding, which works more in parallel and suitable for high-speed and low-latency applications. BP decoding is an iterative message passing algorithm, which is based on the encoding graphs, which will be referred to as factor graphs (FGs) of the polar code in this work. Log-likelihood ratios (LLR) of the messages pass along the factor graph [Forney, 2001],  $N_{it}$  (number of iteration) times iteratively, if there is no other early stopping condition. Arıkan has shown that the BP decoding algorithm has performance advantages for polar codes over Reed-Muller codes [Arıkan, 2008]. There is an extensive literature on the comparison of polar code decoders with the SC, some improved forms of the SC and the BP decoding algorithms [Hussami, Korada, & Urbanke, 2009], [Korada, 2009], [Arıkan, 2010]. Eslami and Pishro-Nik have performed simulations showing that the error floor performance is superior to that of the LDPC codes [Eslami & Pishro-Nik, 2010], [Eslami & Pishro-Nik, 2013]. Implementing BP on field programmable gate arrays (FPGA) has been attempted by Pamuk [Pamuk, 2011], where he also states that for efficient hardware design, the message passing algorithm can be approximated to min-sum (MS) algorithm at the cost of some performance degradation. Yuan and Parhi further have suggested scaled min-sum (SMS) algorithm to remove this performance loss [Yuan & Parhi, 2013]. Then the same authors have improved the efficiency of their algorithm by suggesting some early termination criteria [Yuan & Parhi, 2014a]. Furthermore, Xu et al. show that the same decoding performance of the SMS algorithm with 92.8% reduced amount of computations can be achieved with the scheduling method that

they call XJ-BP MS algorithm [Xu, Che, & Choi, 2015]. When they compare with the conventional MS BP decoding, their proposed method reduces the computations by 90.4% and significantly improves the decoding performance. Some other studies show that the bit mapping scheme can improve the performance of concatenated polar codes with the LDPC codes [Yu, Shi, Deng & Li, 2018]. Hybrid BP-SC(L) decoders also achieve good BLER performance [Yuan et al., 2014b] [Cammerer et al., 2017]. Inspired by the successive cancellation flip (SCF) decoder, bit-flip is introduced to the BP decoder; the proposed belief propagation flip (BPF) decoder achieves significant SNR gain comparable to that of the CA-SCL decoder with a moderate list size [Yu et al., 2019].

The subject of permuted factor graphs (FGs) under BP decoding is mentioned in the studies for error-correction performance of Korada [Korada, 2009]. Different permutations of the layers in the factor graph can construct  $n!$  (where  $n = \log_2 N$ ) different FG representations for a polar code of length  $N$ . Due to the different order of processing in the decoding graph, for each codeword and noise realization, each individual factor graph representation may have different performance. That means, if one FG used for decoding does not succeed in, the other may successfully decode that same code block. Decoding on different FGs in parallel and combining all obtained decoding results has also been mentioned in Korada's studies [Korada, 2009]. In the same studies, it is suggested to use only the  $n$  of  $n!$  permutations, obtained by the cyclic shifts, as multiple-FG decoders. Multiple factor graph BP decoders has also been studied recently. It is shown that based on different permutations of the polar code factor graphs, a new CRC-aided variant of the BP decoder approaches the error ratio performance of the state-of-the-art SCL decoder of a plain polar code, in the high SNR region [Elkelesh, Ebada, Cammerer, & Brink, 2018a]. However, the required number of randomly selected parallel BP decoders to achieve a reasonable error probability, is too high for practical applications. Based on this study, a BP list (BPL) decoder is proposed that also includes the cyclically shifted set of  $n$  FGs among  $L > n$  different FG sets, which reaches the performance of the

randomly selected FG decoder sets using less number of FGs [Elkelesh, Ebada, Cammerer, & Brink, 2018b]. According to these results, FG decoder set needs to be constructed wisely. Some design algorithms are proposed to find suitable FG sets for decoding with multiple factor graph realizations [Doan, Hashemi, Mondelli, Gross, 2018], [Doğan, 2015], [Peker, 2018] and [Akdoğan, 2018]. We also consider the same multiple BP decoding scheme and propose a new way of choosing  $n$  FGs with good performance for polar BP decoders.

#### 1.4. Aim and Organization of the Thesis

In this thesis, single factor graph (single-FG) and multiple factor graph (multiple-FG), belief propagation (BP) decoding performances of  $(N, K) = (2^n, 2^{n-1})$  polar codes over a binary input AWGN channel are examined. Throughout the simulations, a perfect knowledge-based early stopping criterion is used in the BP decoder, except for the last three figures of Chapter 3, where a more practical stopping condition is utilized. The aim of the study can be summarized as:

- To examine the BP decoding performance difference of polar codes between the reference factor graph (RFG) decoder, which has the stage order  $n-1-2-1$  and the inverse RFG (IRFG) decoder, whose stage order is  $1-2-\dots-n$ , in terms of the block (codeword) error ratio (BLER).
- To explore the performance difference between those constructed with the PCC-0 algorithm suggested in [Vangala et al., 2015] and those designed with respect to the specific SNR of the utilized channel, for polar BP decoders at different code lengths.
- To compare the single-FG and multiple-FG decoder performances of polar BP decoders.
- To find good performing multiple-FG sets of FGs with permuted stage orders. To compare their performance with the ones suggested in the literature.

- To make a performance comparison of multiple-FG belief propagation decoders for the polar and RM codes.

The remainder of this paper is organized as follows:

In Chapter 2, the basic concepts of polar code construction and BP decoding of polar codes are briefly introduced. The core concepts of channel polarization, capacity calculation for AWGN channel and the relation between capacities and Bhattacharyya parameters are reviewed.

In Chapter 3, after determining the required number of iterations for the BP decoder at different code lengths, BLER performances of single FG decoders; i.e., the reference FG (RFG) or its inverse (IRFG), are found in simulations over the binary AWGN channels for  $(N, K) = (2^n, 2^{n-1})$  polar codes. Performances of polar BP decoders are evaluated for polar codes designed at fixed design-SNRs, and also at variable, channel-specific design-SNRs. Then, BLER performances of single-FG decoders are compared with those of the multiple-FG decoders, which consist of either randomly or deterministically selected FG sets, such as the cyclic  $n$ -FG decoder constructed from cyclically shifted forms of the RFG. In order to improve the choice of the FGs in the  $n$ -FG belief propagation decoder, two methods, MaxSON and MaxofMax, are proposed for selecting  $n$  FGs with respect to their stage order numbers (SONs). Moreover, the difference between multiple-FG belief propagation decoder performances of the RM codes and polar codes is investigated. Lastly, a performance comparison between the methods proposed in this work and multiple-FG BP decoders suggested in the literature is given.

In Chapter 4, main contributions of this thesis are discussed.



## CHAPTER 2

### POLAR CODES

In this chapter, we review the polar code construction based on the work of Arıkan [Arıkan, 2009]. After some preliminary information, the calculation of channel capacities and the selection of information channels by channel combining and splitting are summarized [Arıkan, 2009]. The construction of polar codes for a binary input AWGN channel is reviewed. The factor graph (FG) representation to be used in encoding and decoding operations is given. Finally, belief propagation (BP) decoding is described briefly.

#### 2.1. Preliminaries

Let  $W: X\{0, 1\} \rightarrow Y$  be an arbitrary binary-input discrete memoryless channel (B-DMC) where  $X$  is input alphabet,  $Y$  is output alphabet, and  $W(y|x)$  is channel transition probability  $\{W(y|x): x \in X, y \in Y\}$ . Given a B-DMC  $W$ , let  $I(W)$  denote the symmetric capacity defined as the mutual information (in bits) between the input and output terminals of  $W$  when the input is chosen from the uniform distribution on  $X$ .  $I(W)$  is the highest rate at which reliable communication is possible across  $W$ . Another parameter of primary interest for this study is Bhattacharyya parameter  $Z(W)$  which is an upper bound on probability of maximum likelihood (ML) decision error for each use of  $W$  to transmit a 0 or 1.

$$\text{Symmetric capacity: } I(W) = \sum_{y \in Y} \sum_{x \in X} \frac{1}{2} W(y|x) \log \frac{W(y|x)}{\frac{1}{2}(W(y|0)+W(y|1))} \quad (2.1)$$

$$\text{Bhattacharyya parameter: } Z(W) = \sum_{y \in Y} \sqrt{W(y|0)W(y|1)}. \quad (2.2)$$

Let  $u = [u_i]_{1 \times N}$  and  $x = [x_i]_{1 \times N}$  be the information and the code sequences with  $N = 2^n$  respectively.  $W_N: X^N \rightarrow Y^N$  with transition probability  $(y_1^N | x_1^N) = \prod_{i=1}^N (y_i | x_i)$  represents the  $N$  times employments of the channel  $W$ .  $a_1^N$  is also used to denote a row vector  $(a_1, \dots, a_N)$ . The length of information and code sequences is equal to  $N$  since information sequence consists of  $K$  information bits and  $N - K$  frozen bits for a polar code  $P(N, K)$ . Code rate  $R = K/N$  comes from the frozen bits in the information sequence. The encoding carried out in GF (2) is

$$x = u \cdot G_N. \quad (2.3)$$

The Kronecker product of two matrices  $A$  and  $B$  is written as  $A \otimes B$ , and the  $n^{\text{th}}$  Kronecker power of  $A$  is  $A^{\otimes n}$ , where Kronecker power is defined by  $A^{\otimes n} \triangleq A \otimes A^{\otimes n-1} = A^{\otimes n-1} \otimes A$ . Arıkan's input transformation matrix is given by  $G_N = G_2^{\otimes n}$  where  $G_2 = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$  and  $n = \log_2 N$ . Unless specified otherwise, all vectors, matrices, and operations on them is carried out over the binary field GF (2). For  $a_1^N$ ,  $b_1^N$  vectors over GF (2) we use  $a_1^N \oplus b_1^N$  to denote their component wise modulo-2 summation.

Choosing the rows of  $G_2^{\otimes n}$ , which forms the generator matrix of the code, is the main idea of the polar code construction which explained in the next sections.

Through the thesis, binary phase-shift keying (BPSK) modulation and binary input additive white Gaussian noise (BAWGN) channel model are considered. The BAWGN channel is the most common approach to model the effect of random sources which occur in the nature. For  $W: X \{0, 1\} \rightarrow Y$ , the channel output is given by

$$y_i = (1 - 2x_i) \sqrt{\frac{RE_b}{\frac{N_0}{2}}} + n_i, \quad (2.4)$$

where  $x \in X \{0, 1\}, y \in Y, n_i \sim \mathcal{N}(0, 1)$ .  $E_b$  denotes the energy spent per each information bit. In our work, we used BPSK modulation such that it maps  $0 \rightarrow \sqrt{\frac{RE_b}{N_0}}$  and  $1 \rightarrow -\sqrt{\frac{RE_b}{N_0}}$  and the channel had normal distribution  $\mathcal{N}(0, 1)$ . For a given code rate  $R$ , SNR (dB) for the given channel is  $E_b/N_0$ . The distribution of  $n_i$  is equal to

$$p_{n_i}(n_i) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{n_i^2}{2\sigma^2}\right), \quad \sigma^2 = 1. \quad (2.5)$$

The decoders used in the thesis work in the log-likelihood ratio (LLR) domain which corresponds to how many times more likely the data are under one model than the other. The corresponding soft information for the input alphabet  $\{0, 1\}$  can be calculated as follows:

$$LLR(y_i) = \ln \frac{p(y_i|x_i=0)}{p(y_i|x_i=1)}.$$

In our case it can be simplified as:

$$LLR(y_i) = 2y_i \sqrt{\frac{RE_b}{N_0}}. \quad (2.6)$$

For SNR =  $E_b/N_0$  (dB) and the  $E_c = \frac{RE_b}{N_0}$ .

Figure 2.1 shows the conditional probability distributions of the AWGN channel output for BPSK modulation.

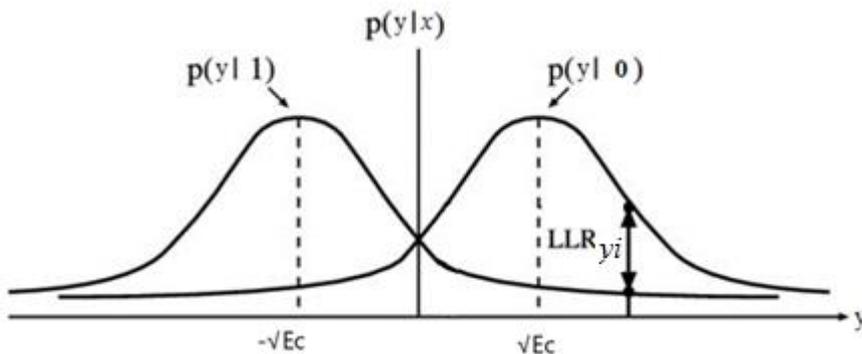


Figure 2.1. Conditional probability distribution for BPSK modulation over AWGN channel.

## 2.2. Polar Coding

Channel polarization is an operation by which one creates, from  $N$  independent copies  $W$  of a given B-DMC, a set of  $N$  channels  $\{W_N^{(i)}: 1 \leq i \leq N\}$  that show a polarization effect; in the sense that as  $N$  becomes large, the symmetric capacity terms  $\{I(W_N^{(i)})\}$  tend towards either 0 or 1 for almost all indices  $i$ . This operation consists of two phases called channel combining and channel splitting.

### 2.2.1. Channel Combining

To produce a vector channel  $W_N: X^N \rightarrow Y^N$  one needs to combine copies of given a B-DMC,  $W$ , in a recursive manner; first starting with two independent copies of  $W_1 \triangleq W$  to create  $W_2: X^2 \rightarrow Y^2$  by applying the transform  $G_2 = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$  as shown in Figure

2.2. The transition probabilities for the channel  $W_2$  are

$$W_2(y_1, y_2 | u_1, u_2) = W(y_1 | u_1 \oplus u_2) W(y_2 | u_2). \quad (2.7)$$

Since a linear transformation is applied for mapping  $u_1^2 \rightarrow x_1^2$  and since  $u_i$ 's have an identically independent distribution (i.i.d.), the symmetric capacity for the channel  $W_2$  is equal to two times the symmetric capacity of the channel  $W$ .

$$\begin{aligned}
I(W_2) &= I(U_1, U_2; Y_1, Y_2) = I(X_1, X_2; Y_1, Y_2) = I(X_1; Y_1) + I(X_2; Y_2) \\
&= 2I(W) \quad (2.8)
\end{aligned}$$

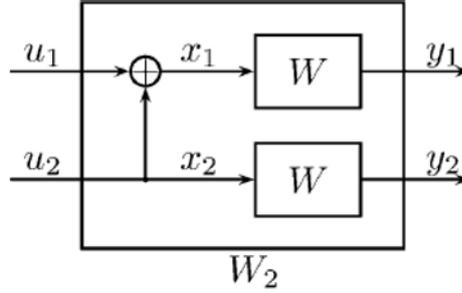


Figure 2.2. Channel combining for  $W_2$ .

The next recursion, which is shown in Figure 2.3, is applied by combining two independent copies of  $W_2$  to create  $W_4: X^4 \rightarrow Y^4$  with transition probabilities

$$W_4(y_1^4 | u_1^4) = W_2(y_1^2 | u_1 \oplus u_3, u_2 \oplus u_4) W_2(y_3^2 | u_3, u_4). \quad (2.9)$$

The mapping  $u_1^4 \rightarrow x_1^4$  can be written as  $x_1^4 = u_1^4 G_4$ ,

where

$$G_4 = G_2^{\otimes 2} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}.$$

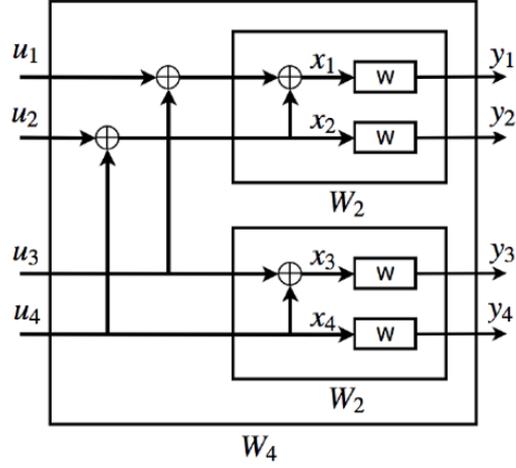


Figure 2.3. The relation of channel  $W_4$  with  $W_2$  and  $W_1$ .

The recursion occurs until creating  $N$  channels  $W_N: X^N \rightarrow Y^N$ . The general form of the channel is shown in Figure 2.4. The general transition probability for the channel  $W_N$  is

$$W_N(y_1^N | u_1^N) = W^N(y_1^N | u_1^N G_N) = W_{\frac{N}{2}}\left(y_1^{\frac{N}{2}} \left| u_1^{\frac{N}{2}} \oplus u_{\frac{N}{2}+1}^N \right.\right) W_{\frac{N}{2}}\left(y_{\frac{N}{2}+1}^N \left| u_{\frac{N}{2}+1}^N \right.\right), \quad (2.10)$$

given that

$$G_N = G_2^{\otimes n}, \quad x_1^N = u_1^N G_N = u_1^N G_2^{\otimes n},$$

$$u_1^{\frac{N}{2}} = \left( u_1, u_2, \dots, u_{\frac{N}{2}} \right) \quad \text{and}$$

$$u_{\frac{N}{2}+1}^N = \left( u_{\frac{N}{2}+1}, u_{\frac{N}{2}+2}, \dots, u_N \right).$$

According to the chain rule for mutual information, the symmetric capacity for the combined channel becomes

$$I(W_N) = I(U^N; Y^N) = I(X^N; Y^N) = NI(W). \quad (2.11)$$

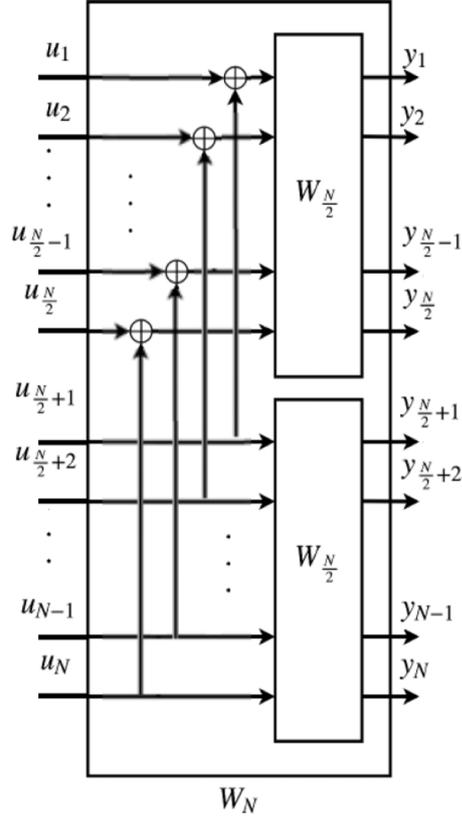


Figure 2.4. The construction of  $W_N$  from two copies of  $W_{N/2}$ .

### 2.2.2. Channel Splitting

Next step of channel polarization is channel splitting of previously combined vector channel  $W_N$  back into a set of  $N$  binary input coordinate channels.

Starting with  $N = 2$ , a new channel can be designed by assuming  $u_1$  as the input  $y_1, y_2$  as the output and  $u_2$  as random for the channel shown in Figure 2.5. Resulting channel can be represented as  $W_2^{(1)}: X \rightarrow Y^2$ . The transition probability for this split channel is

$$W_2^{(1)}(y_1^2|u_1) \triangleq \sum_{u_2} \frac{1}{2} W(y_1|u_1 \oplus u_2)W(y_2|u_2). \quad (2.12)$$

The other channel, represented with  $W_2^{(2)}: X \rightarrow Y^2 \times X$ , can be defined so as  $u_2$  is the input and  $y_1, y_2$  and  $u_1$  are the outputs. Its transition probability becomes

$$W_2^{(2)}(y_1^2, u_1 | u_2) \triangleq \frac{1}{2} W(y_1 | u_1 \oplus u_2) W(y_2 | u_2). \quad (2.13)$$

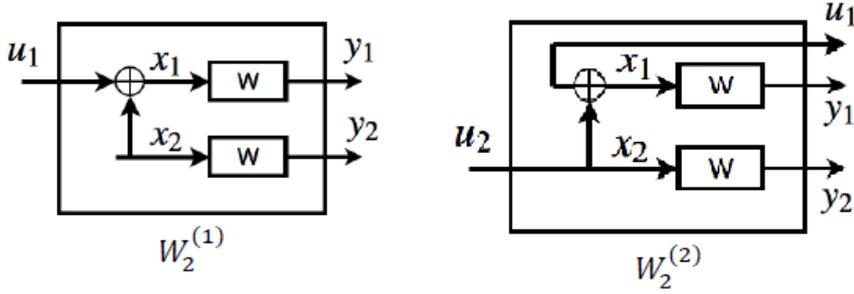


Figure 2.5. The split channels  $W_2^{(1)}$  and  $W_2^{(2)}$  after channel splitting for  $N = 2$ .

The mutual information of the channel  $W_2$  can be split as  $I(W_2) = I(U_1, U_2; Y_1, Y_2) = I(U_1; Y_1, Y_2) + I(U_2; Y_1, Y_2, U_1) = I(W_2^{(1)}) + I(W_2^{(2)})$  such that equation (2.14) holds for symmetric capacities of split channels, with equality only if  $I(W)$  equals 0 or 1. In other words, they are polarized.

$$I(W_2^{(1)}) \leq I(W) \leq I(W_2^{(2)}) \quad (2.14)$$

The other important parameter for polar codes is the reliability parameter, Bhattacharyya parameter  $Z(W)$ , shown in equation (2.2). After applying channel splitting, Arıkan shows that [Arıkan, 2009] Bhattacharyya parameters of bit channels have bounds

$$Z(W_2^{(2)}) = Z(W)^2, \quad (2.15)$$

$$Z(W_2^{(2)}) \leq Z(W) \leq Z(W_2^{(1)}) \leq 2Z(W) - Z(W)^2. \quad (2.16)$$

When the channel splitting operation is recursively applied, the channels, which are represented as  $W_N^{(i)}: X \rightarrow Y^N \times X^{i-1}$ ,  $1 \leq i \leq N$ , can be defined by the transition probabilities

$$W_N^{(i)}(y_1^N, u_1^{i-1} | u_i) \triangleq \sum_{u_{i+1}^N \in X^{N-i}} \frac{1}{2^{N-i}} W_N(y_1^N | u_1^N), \quad (2.17)$$

where  $u_1^{i-1}, y_1^N$  are the input and output of the channel  $W_N^{(i)}$  successively. Generalized representation of channel splitting can be seen in Figure 2.6.

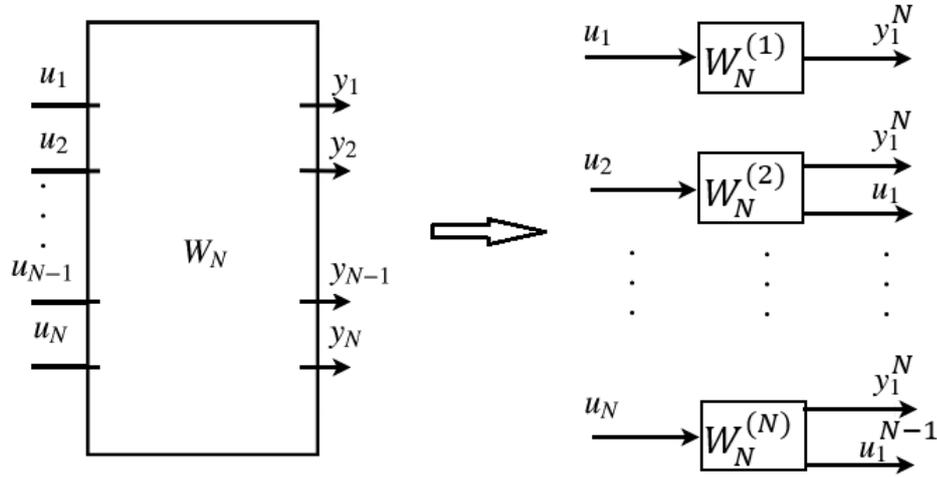


Figure 2.6. Channel splitting of  $W_N$  into  $N$  distinct channels  $W_N^{(i)}$ .

Applying the chain rule of the mutual information one obtains

$$\begin{aligned} I(W_N) &= NI(W) = I(X^N; Y^N) = I(U^N; Y^N) \\ &= \sum_{i=1}^N I(U_i; Y^N, U^{i-1}). \end{aligned} \quad (2.18)$$

### 2.2.3. Channel Polarization

For each bit channel  $W_N^{(i)}$ , channel splitting moves the rate and reliability away from the center such that they are pushed to the extremes 0 and 1. For any B-DMC

$$I(W_N^{(2i-1)}) \leq I\left(W_{\frac{N}{2}}^{(i)}\right) \leq I(W_N^{(2i)}), \quad (2.19)$$

$$Z(W_N^{(2i-1)}) \geq Z\left(W_{\frac{N}{2}}^{(i)}\right) \geq Z(W_N^{(2i)}). \quad (2.20)$$

The Bhattacharyya parameter further satisfies that

$$Z(W_N^{(2i)}) = Z\left(W_{\frac{N}{2}}^{(i)}\right)^2, \quad (2.21)$$

$$Z(W_N^{(2i-1)}) \leq 2Z\left(W_{\frac{N}{2}}^{(i)}\right) - Z\left(W_{\frac{N}{2}}^{(i)}\right)^2. \quad (2.22)$$

The cumulative rate and reliability for the split channels satisfy equations (2.23) and (2.24),

$$\sum_{i=1}^N I(W_N^{(i)}) = NI(W), \quad (2.23)$$

$$\sum_{i=1}^N Z(W_N^{(i)}) \leq NZ(W). \quad (2.24)$$

The Bhattacharyya parameter is an upper bound on the transmission error probability of maximum-likelihood (ML) decision for each use of the channel to transmit a 0 and 1. That means it can be used to measure the error performance of the bit channel. By selecting bit channels which has smaller  $Z(W_N^{(i)})$  values one can assign the noiseless information channels. However, the equalities in equation (2.22) and (2.24) are

achieved only if  $W$  is a binary erasure channel ( $\text{BEC}(\epsilon)$ ), where  $\epsilon$  is the erasure probability of the channel [Arikan, 2009].  $Z(W_N^{(i)})$ , the erasure probability of the channel  $W_N^{(i)}$  can be computed recursively with initial value  $Z(W_1^{(i)}) = Z_0 = \epsilon$ . Due to its simplicity, equality in (2.22) has been widely used for the class of binary-input discrete memoryless channels as well. What we are concerned in this study is polar code construction in AWGN channels.

By using (2.21) and (2.22), Zhao, Shi, and Wang have constructed polar codes, whose information channels are selected by doing modifications on the Bhattacharyya parameter recursion formulas [Zhao et al., 2011] given by

$$\text{Type I:} \quad Z(W_N^{(2i-1)}) = 2Z\left(W_{\frac{N}{2}}^{(i)}\right) - Z\left(W_{\frac{N}{2}}^{(i)}\right)^2 \quad (2.25)$$

$$\text{Type II:} \quad Z(W_N^{(2i-1)}) = Z\left(W_{\frac{N}{2}}^{(i)}\right) \quad (2.26)$$

$$\text{Type III:} \quad Z(W_N^{(2i-1)}) = 0.5 \left( 2Z\left(W_{\frac{N}{2}}^{(i)}\right) - Z\left(W_{\frac{N}{2}}^{(i)}\right)^2 + Z\left(W_{\frac{N}{2}}^{(i)}\right) \right). \quad (2.27)$$

They have compared the polar code performances over BSC, AWGN and Rayleigh channels by selecting the Bhattacharyya parameter of the split channel  $W_N^{(2i-1)}$  as Type I, II and III successively. According to their simulation results, Type I Bhattacharyya parameter has shown the best performance for these channel types [Zhao et al., 2011]. That result supports the idea of using the equality in (2.22) for the Bhattacharyya parameter calculation in AWGN channels; therefore we also use Type I in our simulations.

The definition of the Bhattacharyya parameter given in (2.2) can be extended from discrete to continuous channels in order to find the best initial value of the Bhattacharyya parameter for AWGN channels [Zhao et al., 2011].

$$Z(W) = \int \sqrt{W(y|0)W(y|1)} dy. \quad (2.28)$$

For the AWGN channel with  $\mathcal{N}(0, \sigma)$ , suppose that signal energy of a BPSK signal is 1, then the Bhattacharyya parameter can be simplified by substituting the conditional probability distributions in equation (2.28) with equations (2.29) and (2.30).

$$W(y|0) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y-1)^2}{2\sigma^2}\right). \quad (2.29)$$

$$W(y|1) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y+1)^2}{2\sigma^2}\right). \quad (2.30)$$

$$\begin{aligned} Z(W) &= \int \sqrt{\frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y-1)^2}{2\sigma^2}\right) \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y+1)^2}{2\sigma^2}\right)} dy \\ &= \exp\left(-\frac{1}{2\sigma^2}\right) = \exp(-SNR) = \exp\left(-\frac{RE_b}{N_0}\right). \end{aligned}$$

Then the initial value,  $Z(W_1^{(1)}) = Z_0$  of the recursive algorithm can be obtained by

$$Z(W_1^{(1)}) = Z_0 = \exp(-SNR). \quad (2.31)$$

Since the operational SNR might be modified and the Bhattacharya parameter depends on it, code construction may change which is not practical. Therefore, for other channels, Arıkan [Arıkan, 2008] proposed a heuristic method which offers behaving other channel as an equivalent of BEC, which has the same channel capacity because

of the fact that the exact code construction rule for arbitrary binary-input channels is too complicated. He suggested that given an arbitrary binary-input channel with capacity  $C$  bits, use the polar code that is matched to the BEC with erasure rate  $= 1 - C$ , meaning that the BEC that has same capacity as the given channel. The original recursive algorithm requires an initial value and this was proposed by Arıkan as 0.5 for any channel [Arıkan, 2008]. On the other hand, it shown that channel-specific designs (specific SNR corresponding to the specific AWGN) has better performances over the polar codes designed at constant design SNR,  $Z_0 = 0.5$  [Zhao et al., 2011]. Some researchers attempt to find universal polar code construction. Vangala suggested that choosing the constant design-SNR of 0 dB and obtaining  $Z_0$  by using equation (2.31) one can obtain polar codes with good performances [Vangala et al. 2015]. This design method is preferred in our studies mostly. In addition to this, in some cases, polar codes which are adaptively constructed with changing SNR is also included for comparison.

#### 2.2.4. Polar Encoding

In the previous parts, we have presented how to obtain polarized  $N$  distinct channels. The basic idea of polar coding is sending the data only through noiseless channels, which have smaller  $Z(W_N^{(i)})$  values by the polarization effect of channel combining and channel splitting operations. For a polar code  $\mathcal{P}(N, K)$  with block length  $N = 2^n$ ,  $n \geq 0$ , and code rate  $R = K/N$ , let  $u = [u_i]_{1 \times N}$  be the information and  $x = [x_i]_{1 \times N}$  be the code sequences.  $K$  information bits are sent through noiseless channels and the remaining  $N - K$  bits are set as frozen bits; i.e., one sends predetermined 0's through that noisy channels. The encoding carried out in GF(2) is  $x = u \cdot G_N$  where  $G_N = G_2^{\otimes n}$  and  $G_2 = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$ .

### 2.2.4.1. Factor Graph (FG) Representation of Polar Codes

The encoding operation  $x = u \cdot G_N$  can be performed by using corresponding factor graph (FG) representation. There are  $\log_2 N = 3$  stages for channel combining operation when  $N$  is equal to 8 as illustrated in Figure 2.7 and Figure 2.8 shows its FG representation.

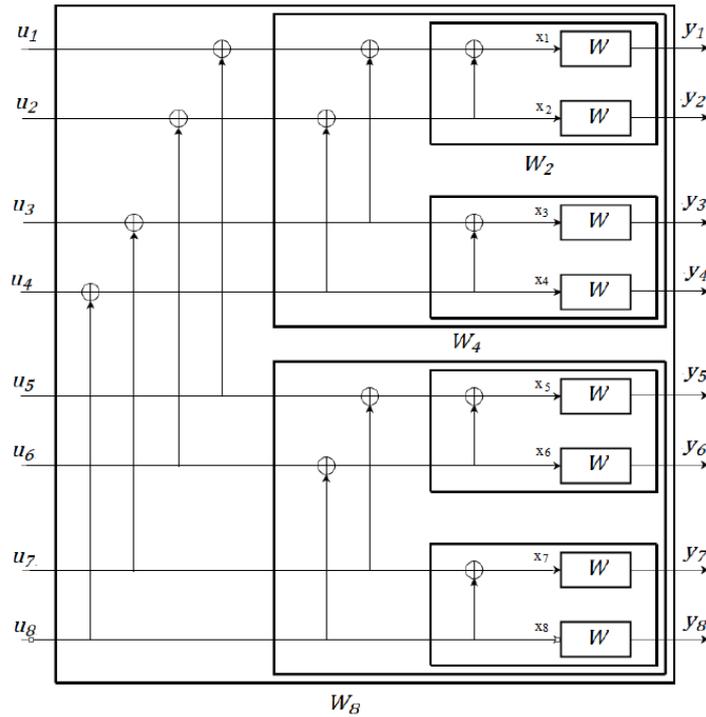


Figure 2.7. The constructed channel  $W_8$ .

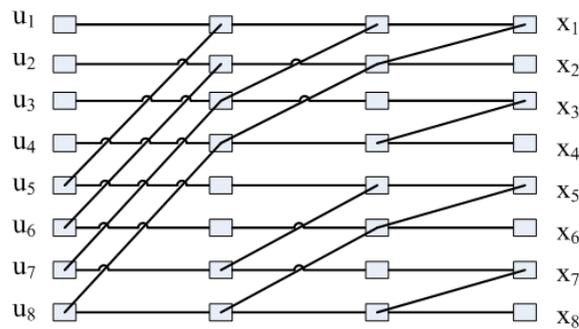


Figure 2.8. Z-shape factor graph representation corresponding to the polar code generator matrix  $G_8$ .

Polar code with length  $N = 2^n$  there are  $n!$  different FG representations which can be obtained by changing the order of the stages used for generating the FG. For code length  $N = 8$  there are 6 different FG types. The one shown in Figure 2.8 with simplified Z-shape connections, is named as “3-2-1” FG that is used for construction of polar codes.

Each factor graph contains  $n$  stages and each stage has  $N/2$  many Z-shape connections. Each stage contains  $N$  input nodes and  $N$  output nodes so that the output of a stage is the input for another stage. Numbering the stages from 1 to  $n$ , we refer to each FG by the left-to right appearance of stages [Doğan, 2015]. In every stage, input nodes are connected as Z-shape by skipping  $2^{(n-1)} - 1$  nodes, where  $n$  is the stage number. For example, the stage named as 1, connects inputs by omitting  $2^{(1-1)} - 1 = 0$  nodes that means it connects consecutive input nodes. The stage 2 connects the nodes by omitting  $2^{(2-1)} - 1 = 1$  and stage 3 connects them by passing over  $2^{(3-1)} - 1 = 3$  nodes. Figure 2.9 shows all different FG representations for the polar code with code length  $N = 8$ . In this study, the FG having stage order  $n \dots 2-1$  is called the reference factor graph (RFG) and the one with the stage order  $1-2 \dots n$  is called the inverse-RFG (IRFG).

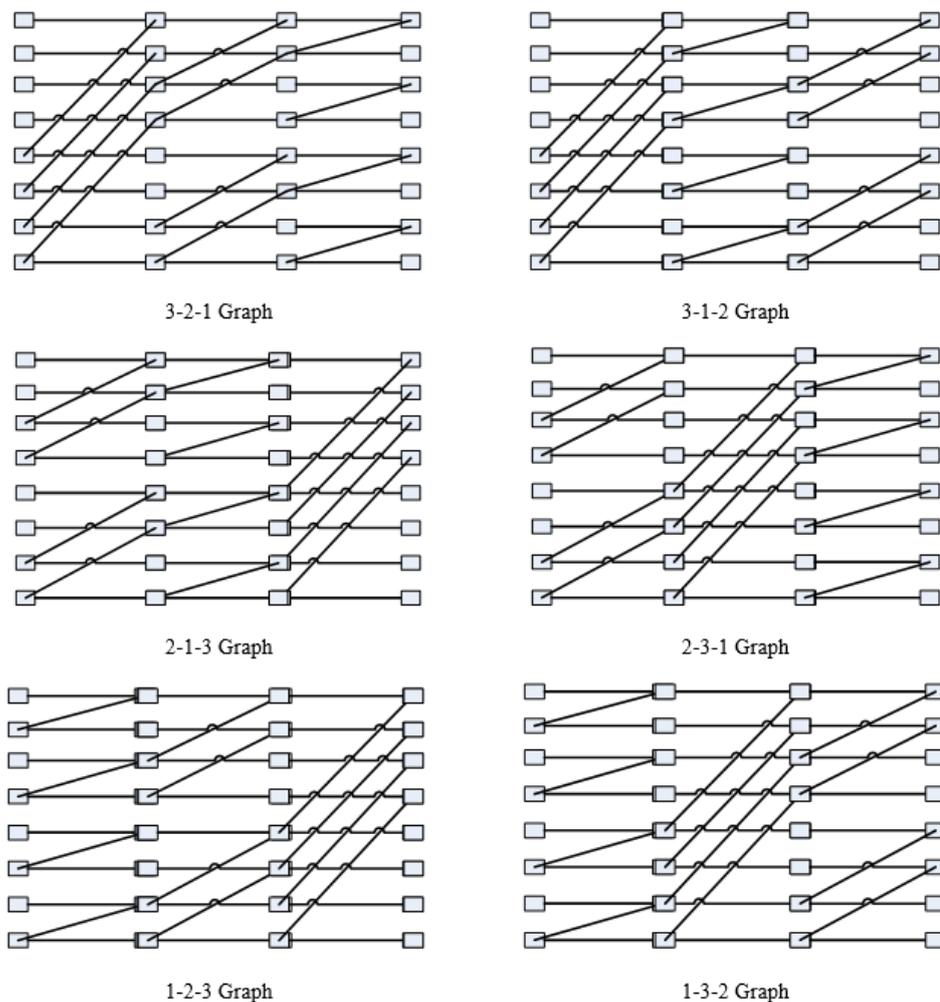


Figure 2.9. All different FG representations for polar code with  $N = 8$ .

#### 2.2.4.2. Selecting Frozen Nodes in AWGN Channel

As we have explained in previous sections, the reliability parameter  $Z(W)$  is used for selecting the nodes in which information is sent. Choosing the design-SNR as 0 dB as suggested [Vangala et al. 2015], with the algorithm given below, one can calculate  $N$  distinct channel parameters in equations (2.21) and (2.25) recursively.

Table 2.1. Algorithm based on the Bhattacharyya bounds

Algorithm based on the Bhattacharyya bounds	
<b>INPUT</b> : $N, K$ , and design-SNR $E_{dB} = (RE_b/N_0$ in dB)	
<b>OUTPUT</b> : $F_i, i \in (0, 1, \dots, N)$	
$z^{(0)}[N] = \{0\}, idx[N] = \{0\}$	
1 :	$SNR = 10^{E_{dB}/10}$ and $n = \log_2 N$
2 :	initialize $z^{(0)}[0] = \exp(-SNR)$
3 :	<b>for</b> $j = 1 : n$ <b>do</b> $\rightarrow$ For each stage in IRFG, left-to-right
4 :	$u = 2j$
5 :	<b>for</b> $t = 0 : u/2 - 1$ <b>do</b> $\rightarrow$ For each connection
6 :	$T = z^{(0)}[t]$
7 :	$z^{(0)}[t] = 2T - T^2$ $\rightarrow$ Upper channel
8 :	$z^{(0)}[u/2 + t] = T^2$ $\rightarrow$ Lower channel
9 :	<b>end</b>
10:	<b>end</b>
11:	$[z^{(0)}, idx]$ sort ( $z^{(0)}$ , “descending”)
12:	$F = idx[0:N-K-1]$
// $F$ : indices of the greatest $N - K$ elements	

As an example, choosing the code length  $N = 8$ , let  $u = [u_i]_{1 \times N}$  be the information sequence, and  $x = [x_i]_{1 \times N}$  be the code sequence where  $i \in \{1, 2, 3, 4, 5, 6, 7, 8\}$ . For code rate  $R = 0.5$  there are 4 information bits and 4 frozen bits. Bhattacharyya parameters calculated with the algorithm given in Table 2.1 are  $z^{(0)}[8] = [0.9745, 0.4410, 0.5911, 0.0363, 0.7062, 0.0637, 0.1300, 0.0003]$ . The detail of the calculation of the Bhattacharyya parameters is also explained in Figure 2.10 by using the factor

graph representation. After the final recursion, the indices of small Bhattacharyya parameters, i.e.  $i_{info} = \{4, 6, 7, 8\}$ , are selected to send information bits and the other ones, i.e.  $i_{frozen} = \{1, 2, 3, 5\}$ , are selected for frozen bits which are predetermined as 0's. Now, let the information word be  $u_i^{(4)} = [1, 1, 0, 1]$  for  $i_{info} = \{4, 6, 7, 8\}$ , then  $u = [u_i]_{1 \times 8}$  becomes  $u = [0, 0, 0, 1, 0, 1, 0, 1]$ . The code sequence is  $x = u \cdot G_N$ . This is the same operation with selecting the rows of  $G_N$  matrix such that they corresponds to the indices of information word,  $i_{info}$ , in the information sequence  $u$  and applying the matrix transform to the information word  $u_i^{(4)}$ . The encoding operation carried out in (2.32) is presented in Figure 2.11 with factor graph implementation.

$$u_i^{(4)} \cdot G_{N_{i_{info}}}^{(4)} = [1 \ 1 \ 0 \ 1] \cdot \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

$$= [1 \ 1 \ 0 \ 0 \ 0 \ 0 \ 1 \ 1] \quad (2.32)$$

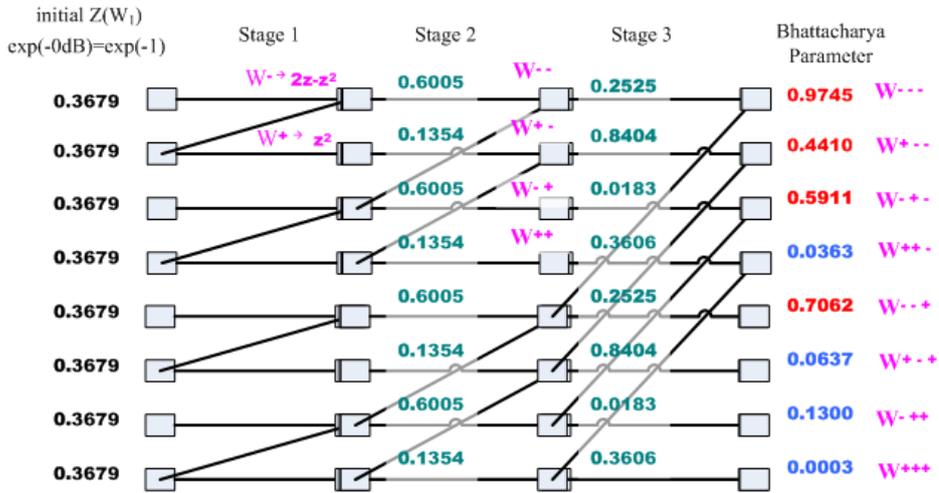


Figure 2.10. Factor graph representation of  $P(8, 4)$  polar code encoding for the information word  $[1, 1, 0, 1]$ .

In Figure 2.10, the recursive algorithm of Bhattacharyya parameter calculation is illustrated by using inverse reference factor graph (IRFG) representation because splitting operations in Figure 2.7 is conducted inside out, which can be represented using the RFG and operating from right to left, or using the IRFG and calculating from left to right. The reason why the least possible Bhattacharyya parameters are only ensured by the RFG stage order is explained in [Arıkan, 2009]. Each initial node is fed with design-SNR of 0 dB, for the initialization of the Bhattacharyya parameter. Each Z-shape creates one “bad” and one “good” channel, represented as  $W^-$  and  $W^+$  successively in Figure 2.10. The good channels have smaller values compared to the bad channels at the end of that stage. The calculations for “ $n$ ” stages are done recursively to obtain final values of polarized bit channels at the end.

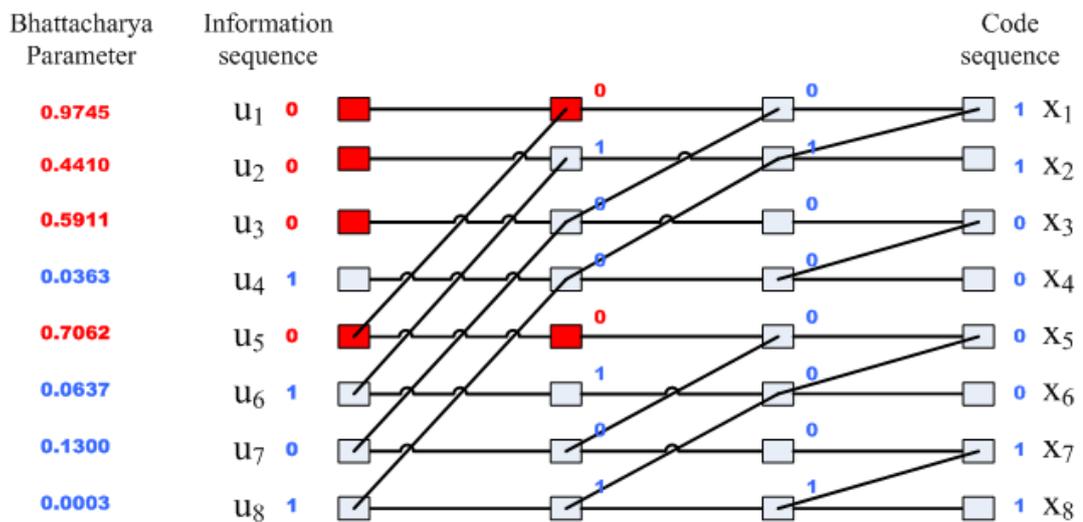


Figure 2.11. Factor graph representation of  $P(8, 4)$  polar code encoding for the information word [1, 1, 0, 1].

After one calculates the Bhattacharyya parameters, fixed frozen input nodes are chosen as the ones with the largest Bhattacharyya parameters. Polar code encoding is done through the medium of reference factor graph (RFG) as given in Figure 2.11,

where the red rectangles correspond to frozen nodes and the white rectangles indicate the information nodes. If a node is made up of only frozen nodes, it becomes a frozen node as well. The first and the fifth output nodes at the end of the first stage can be shown as examples for that situation. Frozen nodes serve an important function in decision making mechanism of polar decoders, which is explained in following section.

### 2.2.5. Belief Propagation Decoding Algorithm

Belief propagation (BP) decoding algorithm is a message passing algorithm, in which one retrieves the information bits through iterations by using factor graphs [Forney, 2001]. As explained in previous sections, polar code FGs are composed of  $n = \log_2 N$  stages, which can be permuted in  $n!$  ways. Hence the BP decoder can use any one of  $n!$  different FGs to decode a received channel sequence. We use the log-likelihood ratios (LLRs) of the channel output in (2.6). The LLR values are iteratively propagated through the FG until the maximum number of iterations is reached. Then a hard decision based decoding algorithm is applied to the final LLR values and the output is compared with the perfect knowledge-based input data.

Two types of the LLR messages are used through the decoding process; one is left-to-right messages ( $L$ -messages) and the other is right-to-left messages ( $R$ -messages). Each Z-shape structure has two input and two output nodes. Processing detail of a single Z-shape connection is demonstrated in Figure 2.12, where a Z-shape contains 4 variable nodes and 2 check nodes and each Z-shape has two input nodes and two output nodes, represented with  $v_i$  and  $v_o$  respectively. The  $j$  and  $k$  indicates the rows of the diagram, where  $j \& k \in \{1, 2, \dots, N\}$ ,  $i$  indicates stage number, where  $i \in \{1, 2, \dots, n\}$  and iteration number is represented with  $t$ .

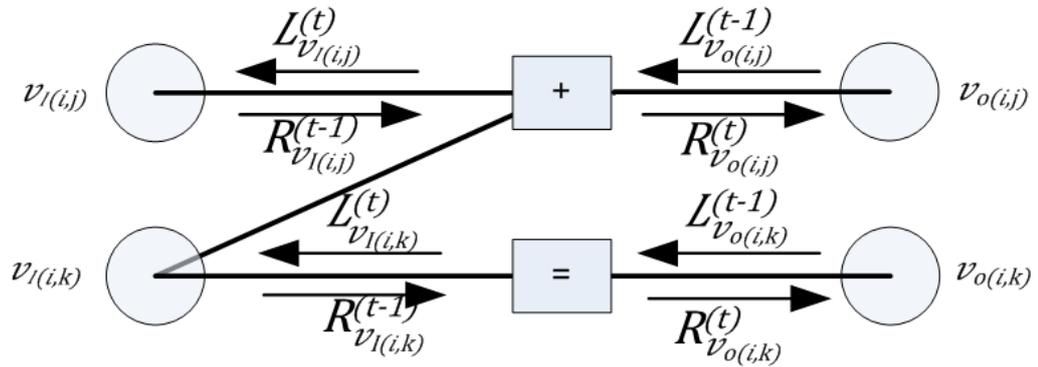


Figure 2.12. Diagram of single Z-shape processing element in the polar BP decoder.

Output nodes of each stage feed input nodes of the following stage; thus, in an FG representation of a polar code  $P(N, K)$ , there are  $nN$  check nodes and  $(n + 1)N$  variable nodes. In Figure 2.13, check nodes and variable nodes of  $P(8, 4)$  are presented with squares and circles successively on the RFG representation.

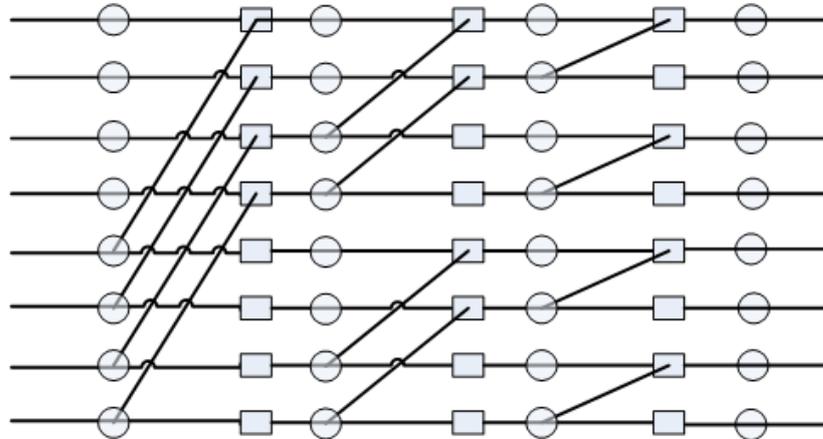


Figure 2.13. Check (square) and variable (circle) nodes of  $P(8, 4)$  on RFG representation.

The initial values of  $L$ -messages at the first stage are set using the a priori information available to the decoder and, thus, zero for non-frozen bits and infinity for frozen bits, respectively. The initial values of  $R$ -messages at the  $(n + 1)^{\text{th}}$  stage are initialized with the LLR values of the channel output. All other nodes except for the frozen nodes are designated as zero. The important point here is to define all the frozen nodes in the FG

representation, since it changes according to the factor graph of selected decoder structure. The  $L$  and  $R$ -messages are updated in each processing element with the following equations:

$$R_{v_o(i,j)}^t = f(R_{v_l(i,j)}^{t-1} \boxplus L_{v_o(i,k)}^{t-1} + R_{v_l(i,k)}^{t-1}), \quad (2.33)$$

$$R_{v_o(i,k)}^t = f(R_{v_l(i,j)}^{t-1} \boxplus L_{v_o(i,j)}^{t-1}) + R_{v_l(i,k)}^{t-1}, \quad (2.34)$$

$$L_{v_l(i,j)}^t = f(L_{v_o(i,j)}^{t-1} \boxplus L_{v_o(i,k)}^{t-1} + R_{v_l(i,k)}^{t-1}), \quad (2.35)$$

$$L_{v_l(i,k)}^t = f(R_{v_l(i,j)}^{t-1} \boxplus L_{v_o(i,j)}^{t-1}) + L_{v_o(i,k)}^{t-1}. \quad (2.36)$$

where  $f(x \boxplus y) = x \boxplus y$  is commonly referred to as the box-plus operator, which corresponds to binary XOR operation and it is defined as follows

$$f(x \boxplus y) = \log\left(\frac{1+e^{x+y}}{e^x+e^y}\right). \quad (2.37)$$

In our decoding version, iterations start from the rightmost of the FG, with right-to-left message (L-messages) propagations as it is used in [Xu et al., 2015] and in [Akdoğan, 2018]; and continues with left-to-right ( $R$ -messages) propagations. This process is called as one iteration. When a processing element has a frozen node with the LLR value of infinity, the importance of frozen variables arises. It may even correct a wrong estimation. Thus one can infer that the FGs having more frozen nodes may correct more decoding errors during iterations, as demonstrated by [Doğan, 2015].

To increase the decoder performance, we have a stop and check condition after every ten iterations, i.e. 10, 20, 30...,  $t_{max}$ . At each stop and check point, LLR values of estimated code sequence  $\check{x}$ , ( $LLR(\check{x})$ ), are computed for all bits according to

$$LLR(\hat{x}_j) = L_{v_l(n+1,j)}^{tmax} + R_{v_l(n+1,j)}^{tmax}. \quad (2.38)$$

The code bits are estimated using hard decisions and compared with the exact transmitted code bits, which is called perfect knowledge-based (PKB) early stopping. If the estimated codeword  $\hat{x}$  is the same as the transmitted codeword  $x$ , then the iterations stop and the codeword is counted as correctly decoded. If PKB stopping does not occur, iterations progress. After a predefined maximum number of iterations is reached, the estimated codeword  $\hat{x}$  is compared to the transmitted codeword  $x$  one last time to decide if it is correctly decoded or not. If the estimated codeword still does not matches to the transmitted codeword, it is counted as undecoded. Perfect knowledge-based early stopping yields simulation curves, which are more of a bound rather than real decoder performances. For that reason, while we compare our results with those in the literature in the last three figures of Chapter 3, we replace the PKB stopping with a practical stopping criterion.



## CHAPTER 3

### SIMULATION RESULTS

In this chapter, simulation results for single or multiple factor graph (FG) belief propagation (BP) decoders of  $(N, K, R) = (2^n, 2^{n-1}, 0.5)$  polar (and Reed-Muller) codes are presented. Section 3.1 is about determining the required number of iterations for the BP decoder. In Section 3.2, the block error ratio (BLER) performances of single-FG belief propagation decoders using either the reference factor graph (RFG) with stage order  $n-1-2-\dots-1$ , or its inverse with stage order  $1-2-\dots-n$  (IRFG) are found. Simulations are made for  $n = 6, \dots, 12$  (sometimes for  $n = 13$  and  $14$  as well) and performances of  $(2^n, 2^{n-1})$  polar codes are compared. The SNR gain of the RFG decoder over the IRFG decoder is found with respect to two design criteria: *i*) fixed design-SNR of 0 dB [Vangala et al. 2015], *ii*) channel-specific design-SNRs corresponding to the SNR of the specific channel. In Section 3.3, multiple factor graph BP decoding performances of polar codes are discussed and two methods that we call “MaxSON” and “MaxofMax” are proposed for choosing the  $n$ -FG sets of multiple-FG decoders. In Section 3.4, single-FG and multiple-FG BP decoding performances of the RM and polar codes are compared, and an exhaustive comparison is given for the (128, 64) codes over all possible FGs. Finally in Section 3.5, our set-choice methods for multiple factor graph BP decoders, MaxSON and MaxofMax, are briefly compared with similar decoders in the literature [Doan, Hashemi, Mondelli, Gross, 2018], [Elkelesh, Ebada, Cammerer, Brink, 2018a], [Elkelesh et al., 2018b].

In each simulation,  $N$ -bit codeword blocks are formed from  $K$  random information bits, generated with uniform distribution. Each bit is passed through an AWGN channel with a pre-chosen SNR. Output blocks of the channel are decoded either by single-FG or multiple-FG belief propagation decoders.

### 3.1. Choice of the Required Number of iterations According to the Code Length

In this part of the work, we have attempted to understand how the decoding performance of the BP decoder changes when the number of iterations varies. For this purpose, at a certain SNR value we have calculated the BLER (block error ratio) performance for different code lengths, such that  $N = 2^n$  for  $n = 6, \dots, 12$  with  $R$  being equal to 0.5. To make a fair comparison, SNR values for different  $N$  values are selected to have similar BLER performances. The BP decoder is implemented using the reference factor graph, RFG with stage order  $n-1$ . Table 3.1 shows the BLER values found in simulations.

Table 3.1. BLER performances with changing  $n$  and changing iteration number.

SNR	2 dB	2dB	1.8 dB	1.75 dB	1.7 dB	1.5 dB	1.3 dB
Iteration number	$n = 6$	$n = 7$	$n = 8$	$n = 9$	$n = 10$	$n = 11$	$n = 12$
5	0.195	0.206	0.329	0.536	0.860	1.000	1.000
10	0.180	0.180	0.222	0.279	0.295	0.511	0.949
20	0.173	0.168	0.193	0.214	0.187	0.259	0.458
30	0.171	0.164	0.182	0.201	0.170	0.203	0.332
40	0.170	0.162	0.180	0.195	0.162	0.171	0.291
50	0.170	0.159	0.175	0.195	0.155	0.160	0.256
60	0.171	0.159	0.176	0.191	0.151	0.141	0.231
70	0.169	0.158	0.172	0.189	0.148	0.135	0.216
80	0.170	0.158	0.171	0.189	0.144	0.125	0.206
90	0.168	0.158	0.171	0.187	0.142	0.123	0.191
100	0.168	0.158	0.171	0.186	0.138	0.121	0.185
120	0.168	0.158	0.165	0.186	0.137	0.121	0.176
130	0.168	0.157	0.165	0.184	0.137	0.121	0.170
140	0.168	0.157	0.164	0.182	0.136	0.118	0.167
150	0.168	0.157	0.163	0.180	0.135	0.117	0.163
160	0.168	0.155	0.163	0.180	0.135	0.116	0.163
200	0.168	0.155	0.163	0.180	0.133	0.107	0.156
220	0.168	0.155	0.163	0.180	0.133	0.105	0.154
250	0.168	0.155	0.163	0.180	0.132	0.104	0.152

In Table 3.1, each column shows the performance at a given code length and SNR value and each row corresponds to the BLER values after using a certain number of BP decoding iterations. “One iteration” in the given FG, which is the RFG with stage order  $n \dots 2-1$  in this case, indicates “the propagation of LLR values from right to left and updating  $L$ -messages then propagation of LLRs from left to right and updating  $R$ -messages”. Whenever the estimated codeword  $\hat{x}$  is the same as the transmitted codeword  $x$ , it is counted as decoded. Each simulation uses 1000 codewords and since the SNR values are adjusted for an approximate BLER on the order of  $10^{-1}$ , 100-170 undecoded blocks remain at the end. For example for  $n = 6$  ( $N = 64$ ) when the iteration number is 5, the RFG decoder recovers 805 of the codeword blocks and 195 out of 1000 remains undecoded. However, if the number of iterations is increased to 40, additional 25 blocks are solved and the number of undecoded word is reduced to 170. It seems that after a certain iteration number, increasing number of iterations has no significant effect on recovery. This number varies according to the chosen code length. In fact, it seems to increase proportionally with the logarithm of the code length according to Figure 3.1, in which we plot the selected number of iterations given in Table 3.1, versus the base 2 logarithm of the code length ( $n = \log_2 N$ ). The complexity of the single-FG BP decoder, that is  $O(N(\log N))$  also increases with length, so it requires more iterations to get more reliable LLR values. However, increasing the number of iterations improves the BLER performance at the cost of decreasing the decoder speed. Therefore, in practical decoders, the optimal number of iterations for each code length should be selected taking this conflict into account.

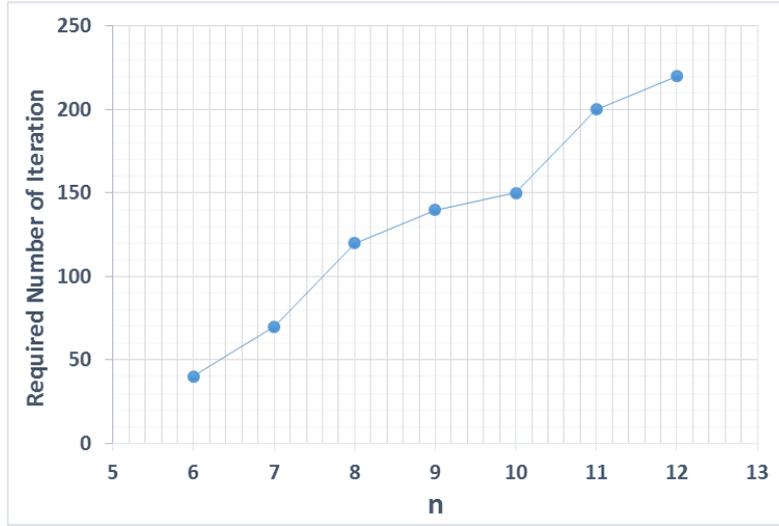


Figure 3.1. Required iteration numbers versus  $n$ , where the code length  $N = 2^n$ .

It can be observed from Table 3.1 that the numbers plotted in Figure 3.1 is sufficient for meaningful performance. For each codeword length, the rest of the study can be carried out with these selected iteration numbers. However, in our simulations, since we also have the knowledge of transmitted bits, we have used its advantage to increase the decoder speed. In compliance with the algorithm, BP decoding terminates after a check point, if the estimated codeword  $\hat{x}$  is the same as the transmitted codeword  $x$ . Check points are settled at every 10 iterations during simulations and the maximum number of iterations is limited to 200. If the condition  $\hat{x} = x$  is not fulfilled at any checkpoint and the maximum number of iterations is reached, the corresponding word is counted as not decoded.

### 3.2. Single Factor Graph BP Decoding Performance of Polar Codes over AWGN

The aim of this section is to observe the single factor graph BP decoding performance of polar codes when the code length changes. As the performance measure, the “block (codeword) error ratio (BLER)”; i.e., the ratio of undecoded words is computed over the simulated AWGN channel. As suggested by Akdoğan, we name each factor graph

(FG) by its “stage order number (SON)”, which is defined as the “ $n$ -digit integer” corresponding to its stage order from left to right [Akdoğan, 2018]; so that  $n!$  different FGs that have different stage permutations can be discriminated. The FG with the highest SON value that has the stage order  $n\dots-2-1$ , is called the “reference FG” and abbreviated as “RFG” [Peker, 2018]. If the stage order is reversed, one obtains the FG with the smallest SON that we call the “inverse RFG” and abbreviate as “IRFG” in this study. For instance, if  $N = 64 = 2^6$ , the RFG has SON = 654321 and the IRFG has SON = 123456. Single-FG decoding performance of the RFG BP decoder is known to be much better than that of the IRFG BP decoder, as observed in previous studies for the polar codes designed over BECs [Doğan, 2015], [Peker, 2018], [Akdoğan, 2018]. In this section, we provide a comparison of the RFG and IRFG BP decoders, for the polar codes designed over AWGN channels.

An adaptive polar code is designed specifically for an AWGN channel with a specific design-SNR and it is supposed to perform the best at this SNR value. However, Vangala, Viterbo and Hong compare four different polar code construction methods at different design-SNR values [Vangala, Viterbo & Hong, 2015] and they draw the conclusion that there are some fixed optimum design-SNRs for each construction method. For the construction method that they call PCC-0, which is also used in our work, they claim that the optimum design-SNR is 0 dB. In other words, they state that the polar code designed by PCC-0 at the design-SNR of 0 dB performs better than those designed at other design-SNRs for all SNR values.

In order to check the validity of this statement, we pick an example polar code  $P(1024, 512)$  for  $n = 10$ , and plot the BLERs of the RFG BP decoder for two cases: *i*) suggested fixed design-SNR of 0 dB [Vangala et al. 2015], *ii*) channel-specific design-SNRs corresponding to the SNR of the specific channel. We observe in Figure 3.2 that Vangala’s assertion is not true, because for the fixed design-SNR of 0 dB, the BLER found in our simulations is 2-3 times that of the channel-specific design at each  $SNR \geq 1.5$  dB. Similarly, at a given BLER, the gain of the channel-specific design over the

0-dB design is measured around 0.2-0.3 dB for the BLERs between  $10^{-1}$  and  $10^{-5}$  of Figure 3.2. Keeping this in mind, unless otherwise is specified, we continue with the fixed design-SNR of 0 dB in our simulations, for the sake of simplicity.

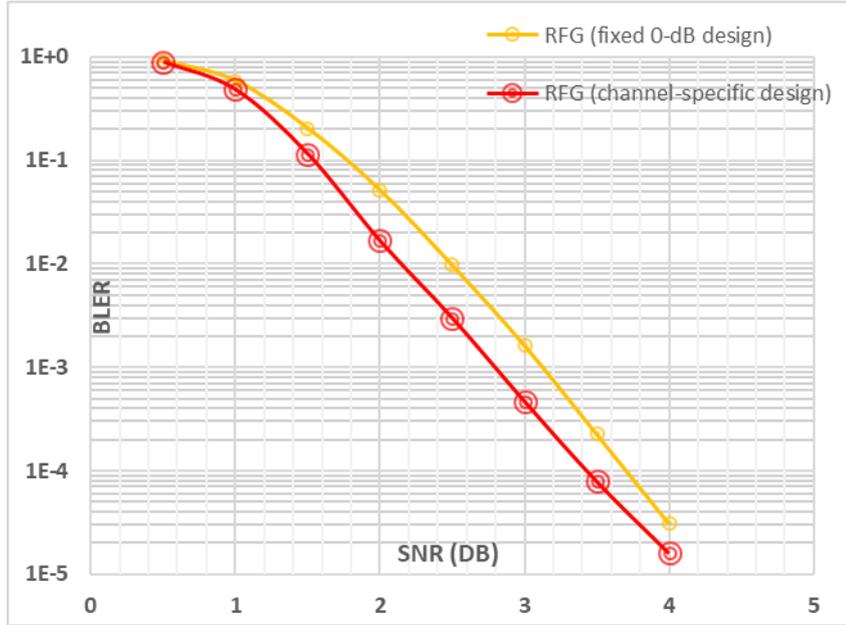


Figure 3.2. Performances of RFG BP decoders for  $P(1024, 512)$  codes, constructed using i) fixed design-SNR of 0 dB (yellow), ii) channel-specific design-SNR at each SNR (red).

Next, we explore the performance variation of the RFG BP decoder with increasing code length, by obtaining the BLERs of the RFG decoders for rate-0.5 polar codes designed for the AWGN channels at 0 dB. The simulations are carried out for  $6 \leq n \leq 12$  (and also for  $n = 13, 14$  in some cases), until a minimum value of 150 block errors are found. The results given in Figure 3.3 for  $6 \leq n \leq 14$  show that the approximate gain obtained by doubling the code length is 0.27 dB at  $\text{BLER} = 10^{-2}$ , 0.35 dB at  $\text{BLER} = 10^{-3}$  and 0.4 dB at  $\text{BLER} = 10^{-4}$ .

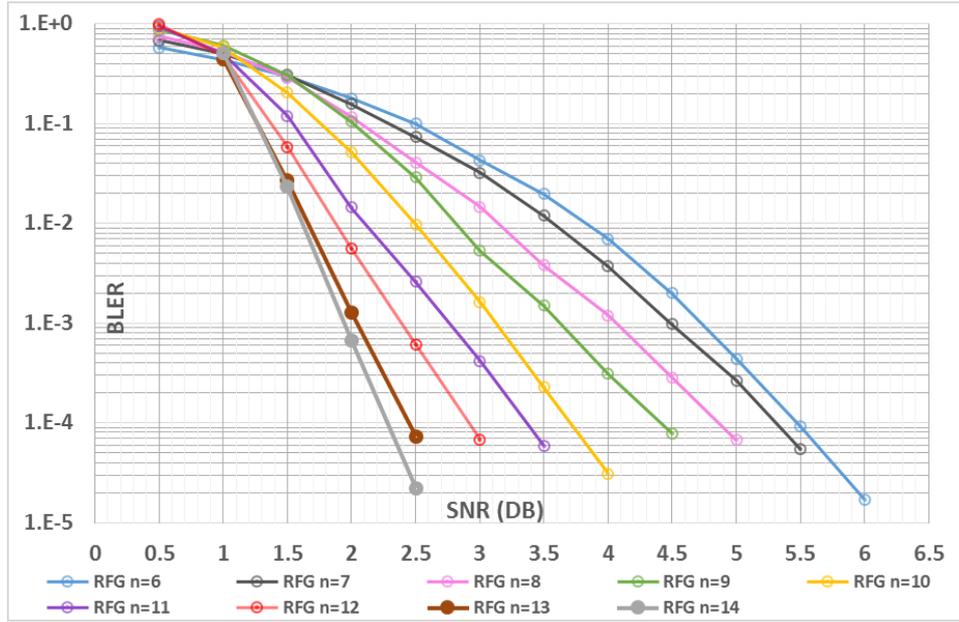


Figure 3.3. BLER of the RFG decoder for polar codes with design-SNR of 0 dB, versus the channel SNR. From top to bottom - Blue:  $n = 6$ ,  $P(64, 32)$ , Black:  $n = 7$ ,  $P(128, 64)$ , Pink:  $n = 8$ ,  $P(256, 128)$ , Green:  $n = 9$ ,  $P(512, 256)$ , Yellow:  $n = 10$ ,  $P(1024, 512)$ , Purple:  $n = 11$ ,  $P(2048, 1024)$ , Red:  $n = 12$ ,  $P(4096, 2048)$ , Brown:  $n = 13$ ,  $P(8192, 4096)$ , Gray:  $n = 14$ ,  $P(16384, 8192)$ .

Since we want to compare the BLER performances of two extreme decoders, namely the RFG and IRFG BP decoders; we repeat the channel-specific design-SNR experiment of Figure 3.2 for the IRFG BP decoders as well. Instead of the example polar code  $P(1024, 512)$ , we now plot the BLERs of the RFG and IRFG decoders for  $6 \leq n \leq 11$ , corresponding to two cases in Figure 3.4: *i*) suggested fixed design-SNR of 0 dB [Vangala et al. 2015], *ii*) channel-specific design-SNRs at each SNR, in addition to the BLERs of the RFG decoder. What we now observe is somewhat incomprehensible for  $n = 6$  and the RFG of  $n = 11$ , where the performance does not improve with channel-specific design; whereas in all other cases, channel-specific design outperforms the fixed-SNR design. At a given BLER, say  $10^{-3}$ , the IRFG BP decoding gain of the channel-specific design over the 0-dB design varies between 0 and 2.9 dB (IRFG curves of  $n = 11$ ) as can be observed in Figure 3.4.

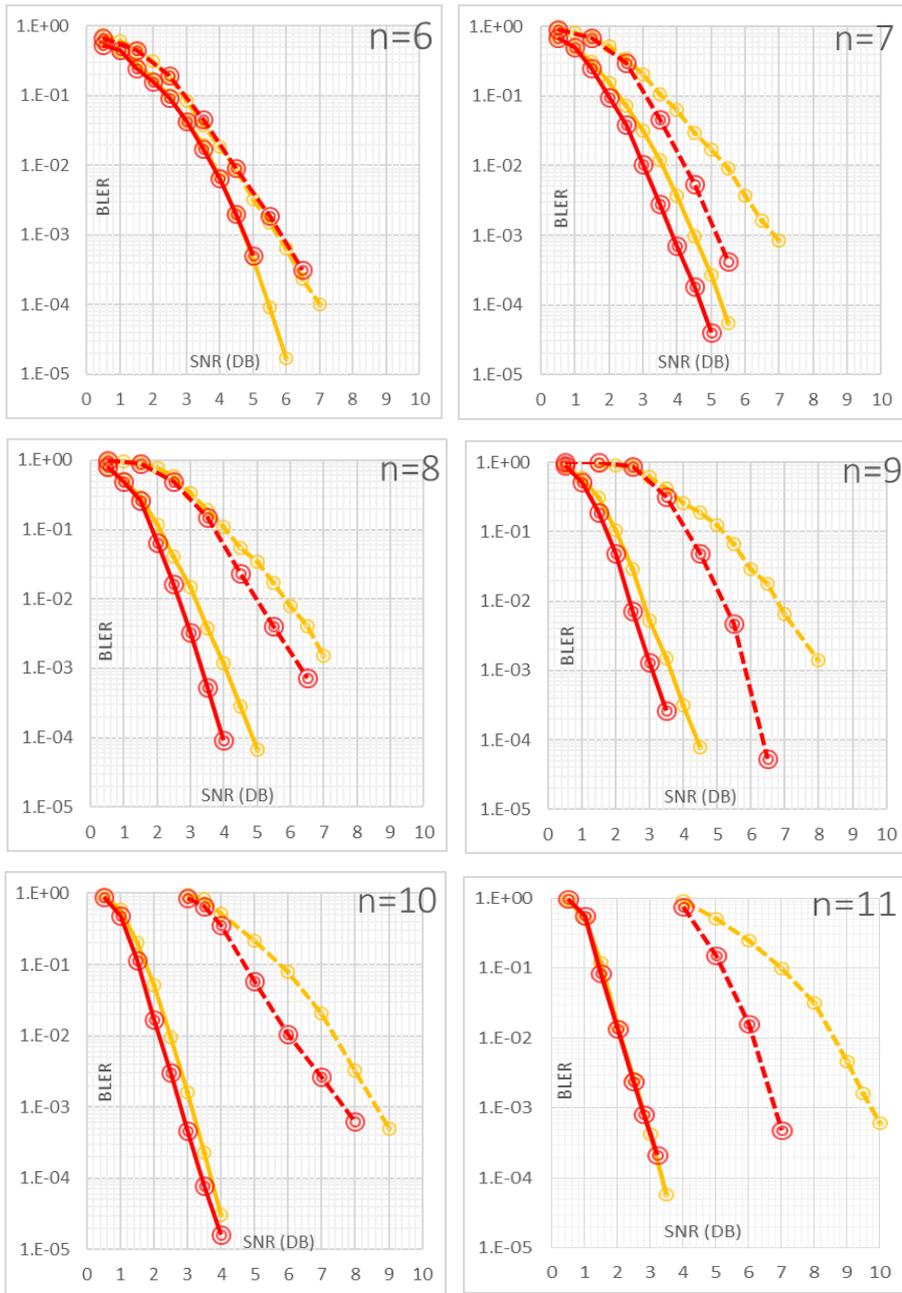


Figure 3.4. BLER performances of the RFG (solid) and the IRFG (dashed) BP decoders for  $n = 6$ ,  $P(64, 32)$ ,  $n = 7$ ,  $P(128, 64)$ ,  $n = 8$ ,  $P(256, 128)$ ,  $n = 9$ ,  $P(512, 256)$ ,  $n = 10$ ,  $P(1024, 512)$  and  $n = 11$ ,  $P(2048, 1024)$  codes, constructed using *i*) fixed design-SNR of 0 dB (yellow curves), *ii*) channel-specific design-SNR at each SNR (red curves).

Again keeping this information in mind, unless otherwise is specified, we simplify our simulations by choosing a fixed design-SNR of 0 dB in the remaining part of this work. In order to compare the RFG BP decoder performance of rate-0.5 polar codes given in Figure 3.3 with the performance of the inverse RFG BP-decoder; we plot in Figure 3.5 the BLER performance of the IRFG decoder for  $6 \leq n \leq 12$ , on top of Figure 3.3 using dotted curves. As in the simulations of Figure 3.3, each simulation point for the IRFG decoder is obtained by counting 150 block errors so that the reliabilities of all simulation points are equalized.

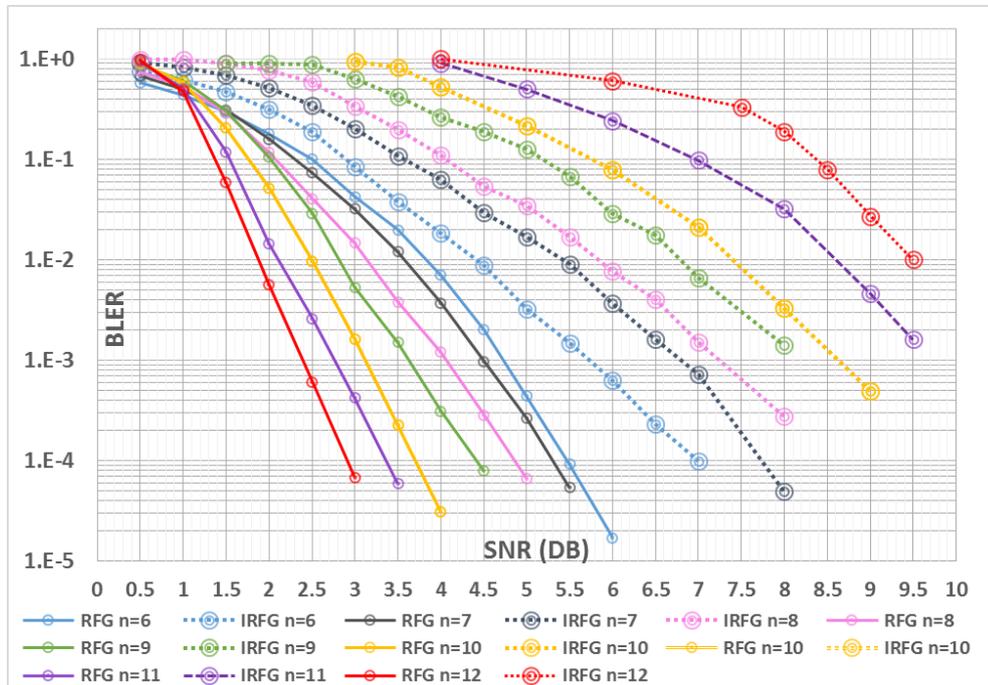


Figure 3.5. BLER of the RFG and IRFG decoders versus channel SNR with design-SNR of 0 dB. Solid lines refer to RFG, and dotted ones of the same color refer to IRFG performances for the same code length. Blue:  $n = 6$ ,  $P(64, 32)$ , Black:  $n = 7$ ,  $P(128, 64)$ , Pink:  $n = 8$ ,  $P(256, 128)$ , Green:  $n = 9$ ,  $P(512, 256)$ , Yellow:  $n = 10$ ,  $P(1024, 512)$ , Purple:  $n = 11$ ,  $P(2048, 1024)$ , Red:  $n = 12$ ,  $P(4096, 2048)$ .

Expectedly, all the dotted curves corresponding to the IRFG decoders have more degraded performances than solid curves corresponding to the RFG decoders, as

observed in Figure 3.5. Remarkably, the gap between the performances of the RFG and IRFG decoders seems to grow when the code length increases. Among all rate-0.5 polar codes with lengths  $N \leq 4096$ , the RFG performance of  $\mathcal{P}(4096, 2048)$  is the best one (solid red curve) as expected; however, its IRFG performance, quite unpredictably, is the worst (dotted red curve). On the other hand,  $\mathcal{P}(64, 32)$  that gives the anticipated worst RFG performance (solid blue curve) has astonishingly the best IRFG performance (dotted blue curve).

The gain of the RFG decoder over the IRFG decoder seems to increase linearly with increasing  $n$ , for instance at  $\text{BLER} = 10^{-2}$ , the gain starts from 0.6 dB for  $n = 6$ , and increases to 7.6 dB for  $n = 12$ . Figure 3.6 shows this gain; i.e., the difference of the SNR values to achieve a  $\text{BLER} = 10^{-2}$  for the RFG and IRFG decoders of the rate 0.5 polar codes with length  $2^n$ . On the same figure, we also include the equivalent gain (of the RFG over the IRFG decoder) for adaptive polar codes constructed using channel-specific design-SNR at each channel SNR.

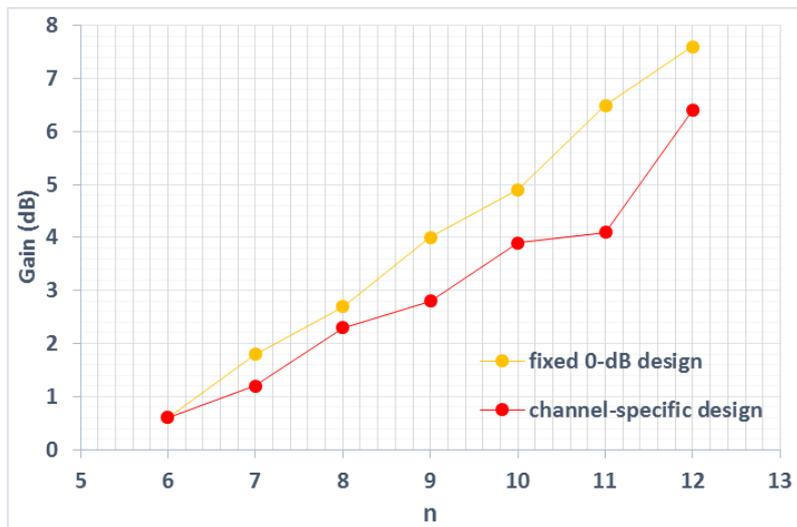


Figure 3.6. Gain of the RFG decoder over the IRFG decoder at  $\text{BLER} = 10^{-2}$  versus  $n$  for polar codes constructed using *i*) fixed design-SNR of 0 dB (yellow curve), *ii*) channel-specific design-SNR at each channel SNR (red curve).

This rapid increase of the RFG decoding performance over the IRFG performance with increasing  $n$  also indicates that random choice of FGs in multiple-FG decoder sets as suggested in the literature [Elkelesh et al., 2018a] may not be effective for high values of  $n$ . Instead, one needs to optimize the chosen FGs by considering a limited variety of FGs, in a group of “suitable” decoders, which exclude the IRFG as well as those having similar performances to that of the IRFG.

It is also of interest to repeat the simulations of Figure 3.5 for a BEC( $\epsilon$ ) with various erasure rates  $\epsilon$ . We present the associated results for  $6 \leq n \leq 12$  in Appendix A, for comparison with those given by the related theses completed at METU ([Doğan, 2015], [Peker, 2018] and [Akdoğan, 2018]).

### **3.3. Multiple Factor Graph BP Decoding Performance of Polar Codes over the AWGN**

A multiple-FG decoder consists of  $M > 1$  factor graphs, which attempt to decode cooperatively. If the received  $N$ -bit block is decoded by the first FG, the remaining FGs need not be used. Otherwise, the undecoded  $N$ -bit block is submitted to the second FG, which either decodes the block or submits it to the next FG. This procedure is repeated for each FG; and the decoding operation is finished if any one of the FGs in the set decodes the received  $N$ -bit block. Otherwise, the raw block is submitted to the next FG until all FGs in the set are exhausted. So, a received block remains undecoded, only if the BP decoders of all the FGs in the set are unsuccessful.

When a new factor graph is in use, rather than employing the log likelihood ratios (LLRs) of the previous factor graph, LLRs are computed from the original channel output data and the BP decoding algorithm uses the stage order of the new factor graph representation (starting from its last stage; i.e., from right to left). Akdoğan calls this method the “independent multiple-FG decoding” as opposed to the “dependent

multiple-FG decoding”, in which FGs operate successively by utilizing the final LLRs of the previous FG within the set [Akdoğan, 2018].

### 3.3.1. Randomly Chosen Multiple Factor Graph BP Decoding Performance of Polar Codes over the AWGN

In this section, BLER performance of randomly chosen multiple-FG decoders is studied. As the number of FGs in the set, we choose  $n = \log_2 N$ , and perform  $n$ -FG decoding as opposed to the single-FG decoders of Section 3.2. We know that the original factor graph of polar codes, the RFG, is one of the decoders with the best performance among  $n!$  FGs [Doğan, 2015], [Peker, 2018], [Akdoğan, 2018], [Elkelesh et al, 2018]. Therefore, to increase the performance of the multiple-FG decoder, the RFG that has the SON =  $n \dots 21$  is always kept in the decoder set and the rest of the  $(n - 1)$  factor graphs are randomly selected out of  $(n! - 1)$  FGs with equal probabilities. We first examine the performance of the  $\mathcal{P}(128, 64)$  polar code over the AWGN channel with constant SNR (2 dB). In each simulation,  $10^4$  channel output blocks are submitted to the decoder, and the RFG is used as the first decoder. If the RFG fails to decode, a random FG among  $(7! - 1) = 5039$  FG’s is invoked. Blocks that cannot be decoded by an FG are attempted by another randomly chosen FG, until the 7<sup>th</sup> FG is reached. 7-FG BP decoding performances of 10 different random 7-FG sets are shown in Figure 3.7.

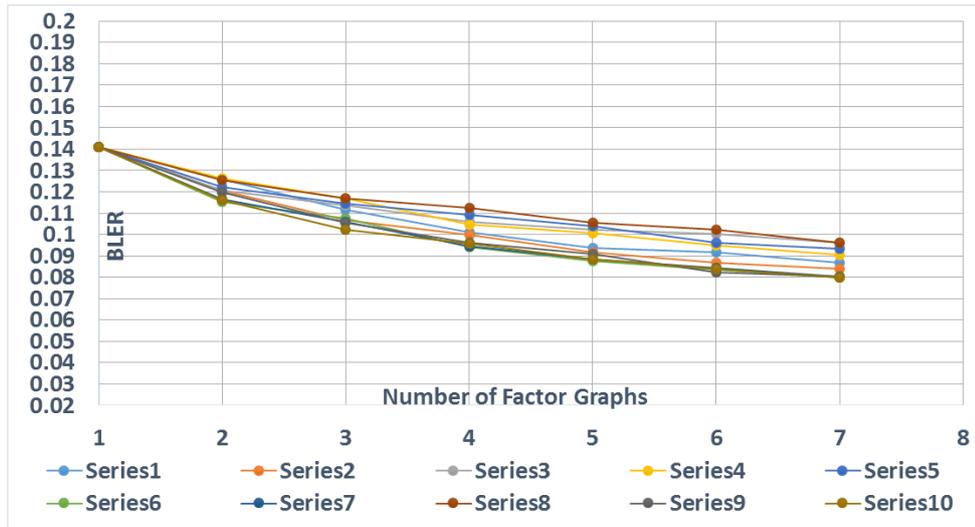


Figure 3.7. BLER performance of ten random 7-FG decoders for  $P(128, 64)$  at SNR=2dB, versus the number of factor graphs.

A sample decoder set for  $P(128, 64)$  corresponding to Series 4 in Figure 3.7 is exemplified in Table 3.2.

Table 3.2. Stage order numbers (SONs) and number of decoded words out of 10000 blocks for a sample set of random factor graphs (FGs) used in 7-FG decoders of  $P(128, 64)$  at SNR=2dB.

	1 <sup>st</sup> FG	2 <sup>nd</sup> FG	3 <sup>rd</sup> FG	4 <sup>th</sup> FG	5 <sup>th</sup> FG	6 <sup>th</sup> FG	7 <sup>th</sup> FG
SON	7654321	2174536	6132457	5672431	2345716	5216347	2463175
Number of decoded words	8590	148	94	121	41	11	46

It can be seen from Figure 3.7 that all 10 different random factor graph sets have similar performances. The RFG decoder initially performs ~0.14 error-ratio and after 6 more decoder's attempts, the BLER value decreases to ~0.09 at that specific noise realization, which is simulated using the same seed in all ten cases for fair comparison.

There are many studies on the performance of polar codes using cyclic shift multiple-FG BP decoders, the complexity of which is  $O(N(\log N)^2)$  [Korada, 2009], [Hussami et al., 2009], [Elkelesh et al., 2018a] [Elkelesh et al., 2018b], [Doan, Hashemi, Mondelli, Gross, 2018]. Cyclic shift factor graph sets have  $n$  elements that can be created from the reference factor graph by shifting each stage cyclically from right to left (or from left to right) one by one. Theoretically, the overall  $n$ -FG performance doesn't change since the order of the FGs in the set does not matter if the set remains the same. For  $n = 7$ , the factor graphs of the cyclic multiple-FG decoder have the SONs: 7654321, 6543217, 5432176, 4321765, 3217654, 2176543 and 1765432.

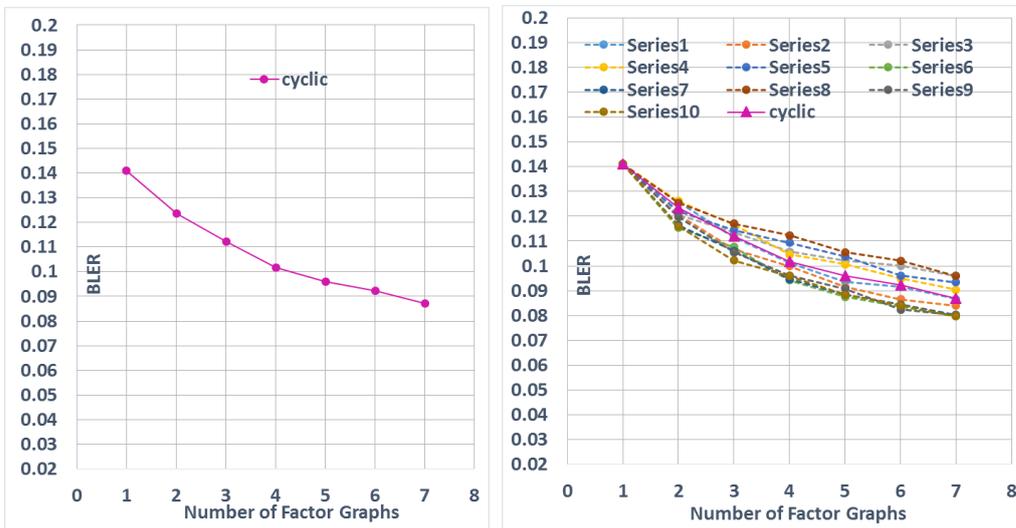


Figure 3.8. BLER comparison of the cyclic 7-FG BP decoder with ten random 7-FG BP decoders for  $P(128, 64)$  at  $SNR = 2$  dB.

In Figure 3.8, we plot the cyclic 7-FG performance on the left alone, and then on the right, we merge it onto the performances of all ten randomly chosen 7-FG sets shown in Figure 3.7. The graph on the right shows us that random and cyclic BP decoders follow a similar trend for  $n = 7$ . One may say that the deterministic cyclic 7-FG BP decoder for  $P(128, 64)$  performs better than approximately half of the ten randomly chosen 7-FG sets, and worse than the other half. Hence, it seems safer to use the cyclic set of 7 FGs rather than a random set.

Next, random  $n$ -FG BP decoder tests are carried out for  $P(1024, 512)$ , to see the performance for a larger code length. Again, the first FG of the set is selected as the RFG decoder, having the stage order:10-9-8-7-6-5-4-3-2-1, and the remaining nine FGs are chosen randomly. SNR value of the channel is set to 1.5 dB to have similar BLER performance with the  $P(128, 64)$  case and  $10^3$  codeword blocks are used for each simulation. It is also of interest to compare the performance of the cyclic 10-FG decoder with those of random 10-FG sets. So, in Figure 3.9, we plot the cyclic 10-FG performance on the left alone, and then on the right, we merge it onto the performances of ten random FG sets. One observes that for  $n = 10$ , the cyclic decoder is better than  $\sim 70\%$  of the ten random sets (Figure 3.9), as opposed to  $\sim 50\%$  observed for  $n = 7$  (Figure 3.8).

By comparing Figure 3.9 with Figure 3.8, one also observes that the performances of randomly chosen  $n$ -FG sets start to differ more as  $n$  increases. In some sets of Figure 3.9 such as Series 10, good FG combinations might coincide and create an effective increase in performance; whereas in some others, worse combination of FGs might meet by chance as in Series 1. As also suggested by rapidly increasing gain of the RFG over the IRFG in Figure 3.6; as  $n$  increases, an  $n$ -FG decoder with randomly chosen FGs may function much more poorly than the RFG alone. Hence, random choice of  $n$ -FG decoders seems not suitable for large values of  $n$ .

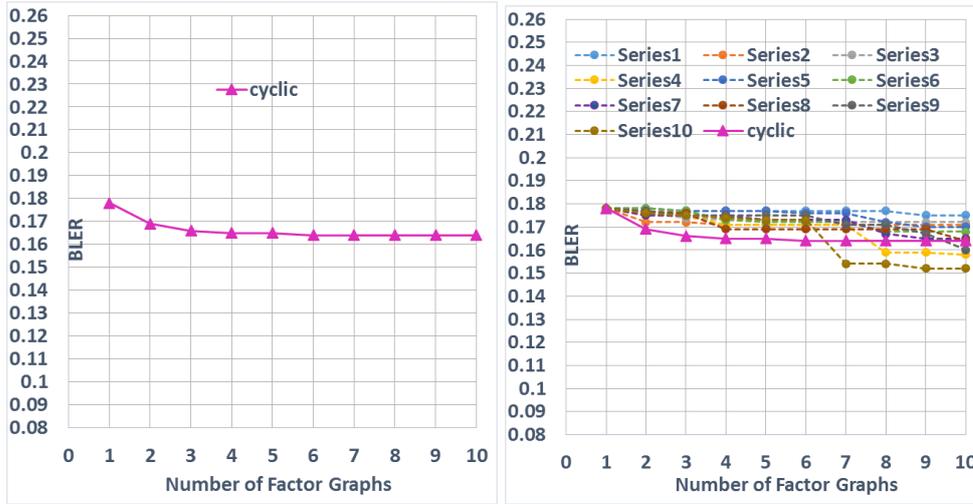


Figure 3.9. BLER comparison of the cyclic 10-FG BP decoder with ten random 10-FG BP decoders for  $P(1024, 512)$  at  $SNR = 1.5$  dB.

Still, the question arises that “How many FGs should be chosen for better performance in terms of both speed and BLER of  $P(1024, 512)$  BP decoders?” So, we increase the number of permuted factor graphs to 100 and make other ten simulations by using  $10^4$  different codewords at  $SNR = 1.5$  dB. Results are presented in Figure 3.10 at quantized steps of 10 FGs; i.e., for  $10k$ -FG decoders, where  $k = 1, \dots, 10$ .

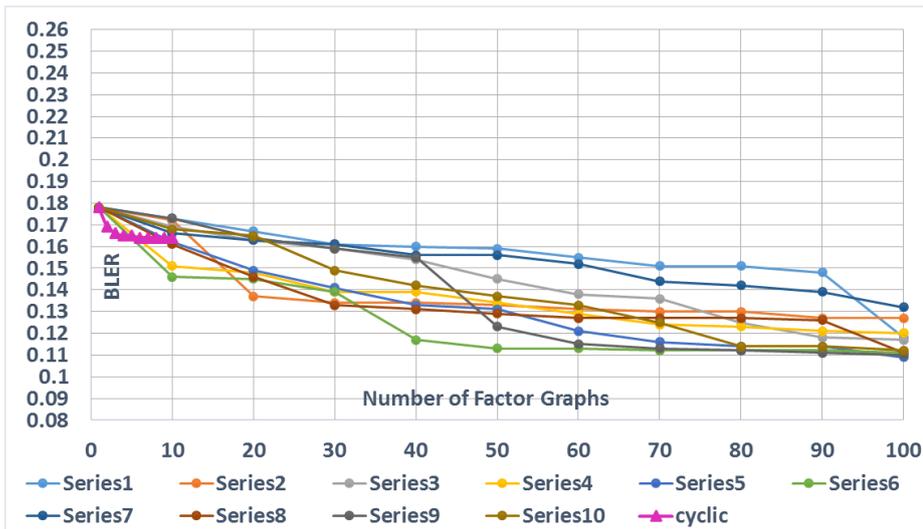


Figure 3.10. BLER of multiple-FG BP decoder for  $P(1024, 512)$  at  $SNR = 1.5$  dB versus the number of FGs, for cyclic 10-FG and ten random choices of 100-FG decoders.

Figure 3.10 shows that if one uses randomly chosen 99 FGs together with the RFG decoder instead of single RFG decoder, the block error-ratio decreases from  $\sim 0.18$  to  $\sim 0.12$  for  $P(1024, 512)$ . However, the slope of the BLER decrease may be very small if the set size is less than 90 (as in Series 1, blue curve of Figure 3.10) or larger than 20 (as in Series 2, orange curve of Figure 3.10). Besides, increasing the number of factor graphs more than enough, causes the decoder work more slowly.

As stated before, the results in this section are consistent with the results in Figures 3.5 and 3.6 of Section 3.2; which show that as  $n$  increases, IRFG and RFG decoder performances have huge differences in their BLER performances. For the polar code  $P(1024, 512)$ ,  $10! = 3,628,800$  different FG representations exist and randomly composed FG sets have poor decoding performances with a large probability. On the other hand, the cyclic 10-FG decoder, whose FGs are chosen deterministically is more promising, since it performs better than  $\sim 70\%$  of the random 10-FG sets generated in our simulations.

In order to determine well performing multiple-FG sets, one may need to focus on single decoding performances of each FG. For example, in Figure 3.9, Series 10 shows a drastic performance gain at the 7<sup>th</sup> decoder which has stage order number, SON = 968A374512 (calling 10 = A). Similarly, in Series 4, it is the 8<sup>th</sup> decoder with SON = 97548A6213 that shows an abrupt performance gain. Moreover, in Figure 3.10, the BLER of Series 1 decreases after the 94<sup>th</sup> FG with SON = 86A7953241 and that of Series 2 after its 15<sup>th</sup> FG having SON = A967538421.

All these factor graph representations with better decoding performances seem to possess a common characteristic: They all have the small stage numbers at the end. Therefore, we think that this characteristic might be taken into consideration while selecting the FG sets of multiple-FG decoders. Next section gives our deterministic set selection methods in more detail.

### 3.3.2. Predetermined Multiple Factor Graph BP Decoding Performance of Polar Codes over the AWGN

Previous section reveals that for the long code lengths, the FG sets of multiple-FG decoders should be selected deterministically; because, performance gain might not be obtained by random selection. Therefore, in this part of the study, we try to find a legitimate approach to construct  $n$ -FG sets (i.e., to find the SONs of FGs in the set) better than the cyclic  $n$ -FG set for the multiple-FG decoder.

Stage order number (SON) uniquely defines an FG representation [Akdoğan, 2018]. The performance of a single-FG BP decoder, is shown to be positively correlated with the number of frozen variables in its representation; and also with the sum of capacities of all information transmission paths [Doğan, 2015], [Peker, 2018]. These two parameters are abbreviated as FV (the number of frozen variables) and CS (capacity sum of information channels) and discussed extensively in the related three M.Sc. theses completed at METU [Doğan, 2015], [Peker, 2018], [Akdoğan, 2018], where polar codes are designed and simulated over BECs.

Akdoğan has also studied over the BEC channel and found the single-FG BP decoding performances of the FGs versus their SON [Akdoğan, 2018]. We perform a similar simulation over the AWGN channel and in Figure 3.11, we plot the BLER performances over an AWGN channel at  $SNR = 1.5$  dB for all single-FG BP decoders of  $P(64, 32)$ , against their SON. This experiment is only carried out for  $n = 6$  due to the formidable profusion of possible FG representations for larger block lengths. The horizontal axis of Figure 3.11 corresponds to the sorted SONs in ascending order; i.e., it starts with 123456, 123465, and ends with 654312, 654321.

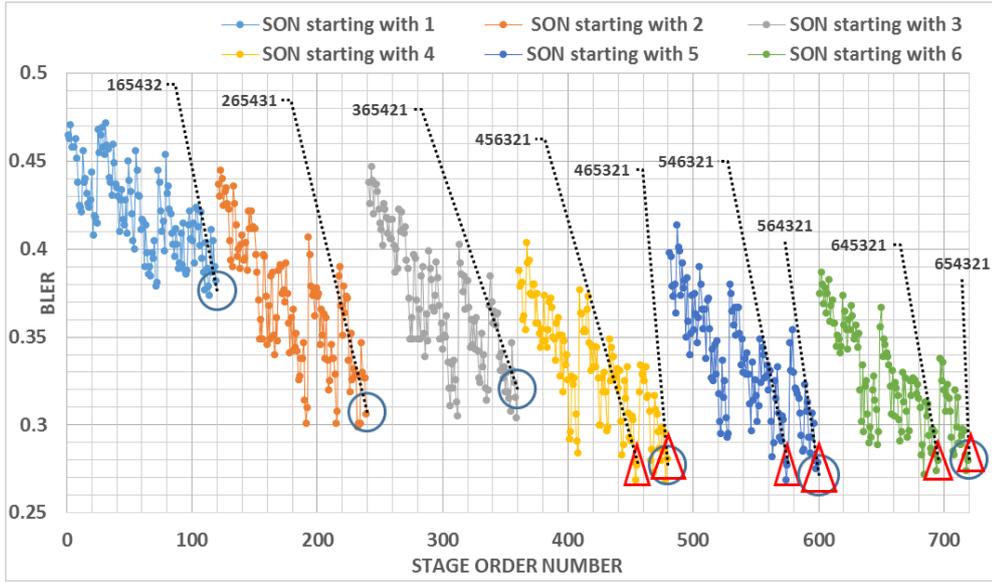


Figure 3.11. BLERs of single-FG BP decoders for  $P(64, 32)$  over the AWGN with  $\text{SNR} = 1.5$  dB, versus their SONs sorted in ascending order (where circles indicate the selection of MaxSON, and triangles show the selection of MaxofMax sets).

In Figure 3.11, we differentiate the SONs starting with different numbers by different colors. Therefore, in each group starting with the same stage number, the leftmost part of the color region corresponds to the minimum SON and the rightmost part corresponds to the maximum SON. Since  $n = 6$ , in each color group there are  $5! = 120$  SONs. When the SONs are sorted from the smallest to the largest; one obtains the BLER performances shown in Figure 3.11 over the AWGN with  $\text{SNR} = 1.5$  dB.

According to this graph, the FGs with maximum SONs in each 120-element group seem to have relatively good BLER performances. They are marked with circles in Figure 3.11. We base our first proposal for the selection of  $n$  FGs of the  $n$ -FG BP decoder upon this observation; so, we pick the maximum SONs in each group (starting with a different stage order) to form an  $n$ -element set. For  $N = 64$ , this rule corresponds to the 6-element set:  $\{654321, 564321, 465321, 365421, 265431, 165432\}$  as the FGs of the  $n$ -FG BP decoder, which is going to be referred to as the

MaxSON decoder choice from now on. One may notice that the first and the last elements of the MaxSON set are shared by the cyclic set as well.

Another good performing group is indicated with triangles in Figure 3.11, which has the SONs ending with 321; i.e., 654321, 645321, 564321, 546321, 465321 and 456321. In other words, they are the ones constructed by shuffling the first 3 digits of the SON of the RFG decoder. For  $n > 6$ , similar  $n$ -element sets can be generated by keeping the last  $(n - 4)$  digits of the SON fixed, while permuting the first 4 digits of the RFG, so that  $4! = 24$  different SON values are obtained. For the  $n$ -FG decoder, one can choose the  $n$  biggest values out of these 24 SONs if  $n \leq 24$ . We name the FG set chosen this way as the MaxofMax set. So, for  $7 \leq n \leq 14$ , we compare the performance of the  $n$ -FG sets chosen as

- i)* all cyclic rotations of the RFG,
- ii)* MaxSON set; i.e., the maximum SONs starting with each one of the  $n$  stages,
- iii)* MaxofMax set; i.e., the  $n$  maximum SONs in the set of permuted first 4 digits.

Notice that all these set choices contain the RFG as their principal element. Moreover, the first and second rules share a second FG as well, which is the FG starting with stage 1. The third rule is defined for  $n \geq 7$  and applicable only to  $n \leq 24$ , or  $N \leq 16,777,216$ ; which seems sufficient for practical purposes. A slightly modulated form of the MaxofMax rule can also be used for  $n = 6$ ; by permuting the first 3 digits of the 6-digit SONs instead of their first 4 digits.

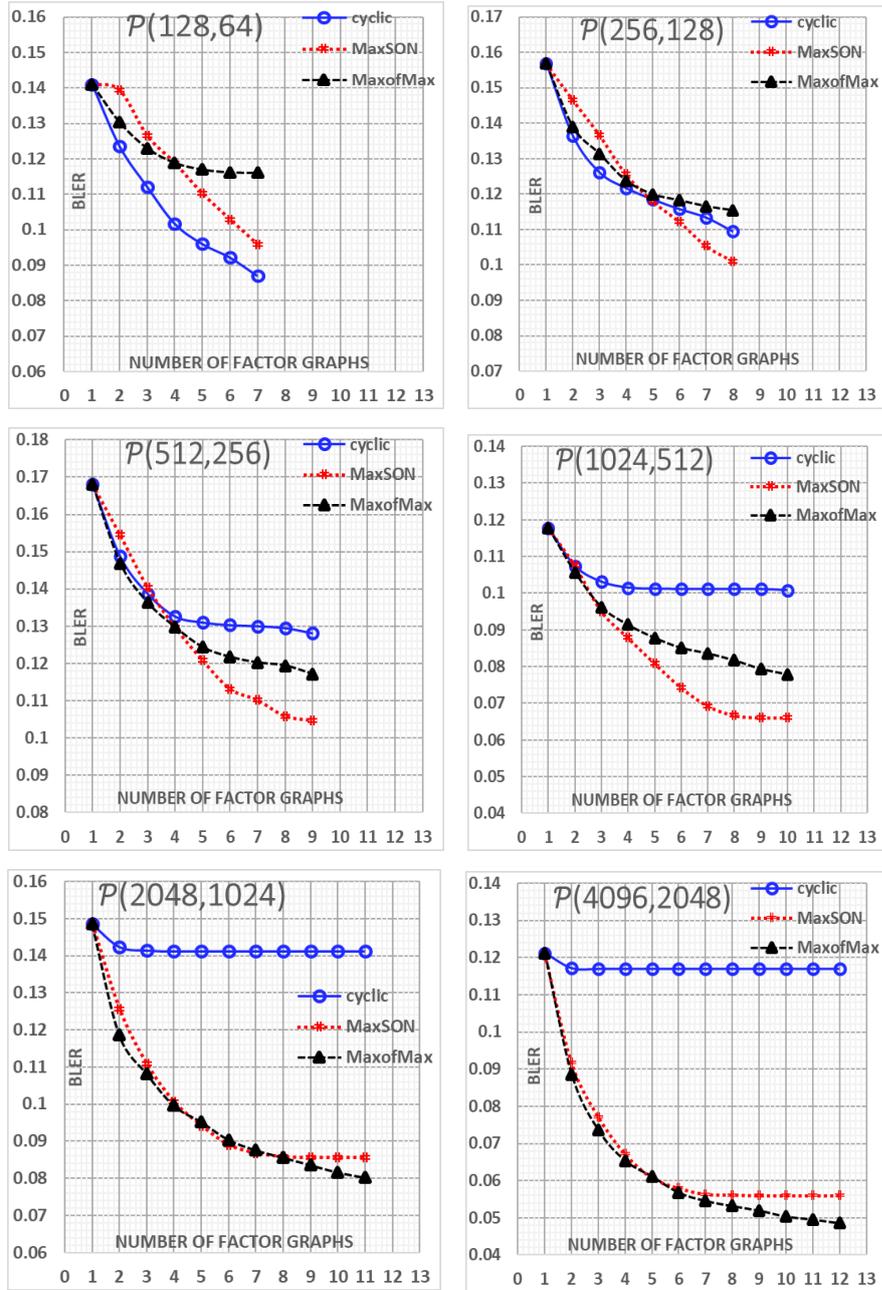


Figure 3.12. BLER performances of the cyclic shift, MaxSON and MaxofMax choices for the FG set of the  $n$ -FG BP decoder for various code lengths  $N = 2^n = 128, 256, 512, 1024, 2048$  and  $4096$ ; where the AWGN channel SNRs are adjusted as 2, 1.8, 1.75, 1.7, 1.4 and 1.35 dB respectively.

In Figure 3.12, the BLER performances of the three set-choice methods are plotted versus the number of factor graphs for  $7 \leq n \leq 12$ . For each code length, we choose

a specific SNR value so that the RFG decoders of all lengths display approximately the same performances.  $10^4$  codewords are used in all simulations, and randomizing seeds of noise samples are kept the same in all simulations to compare the performances of the three set-choice rules fairly.

According to the results presented in Figure 3.12, although the performance of the cyclic decoder for  $P(128, 64)$  is better than those of the MaxSON and MaxofMax choices; as  $N$  grows, the cyclic choice of FGs quickly loses this advantage and becomes the worst among the three choices. On the other hand, the MaxSON choice seems to be the best for  $N = 256, 512$  and  $1024$ , and the MaxofMax choice starts to slightly outperform it for  $N = 2048$  and  $4096$ . Both suggested methods become more profitable than the cyclic decoder structure for  $N > 128$  and it seems worthwhile to prefer them. Appendix B focuses on polar codes  $P(1024, 512)$  and  $P(2048, 1024)$  to analyze the individual effects of single-FG performances of the  $n$  factor graphs in MaxofMax and MaxSON decoders.

To have a better understanding of the performance of the suggested choices of  $n$ -FG sets, we focus on higher code lengths with  $10 \leq n \leq 14$ , and find the BLER performance versus a wider SNR range for the cyclic, MaxSON and MaxofMax  $n$ -FG belief propagation decoders (together with a randomly chosen set for  $10 \leq n \leq 12$ ). Figure 3.13 summarizes the results of this experiment, where we compare the block error ratio performances of the single RFG and multiple  $n$ -FG BP decoders for  $P(1024, 512)$ ,  $P(2048, 1024)$ ,  $P(4096, 2048)$ ,  $P(8192, 4096)$  and  $P(16384, 8192)$ . The number of maximum iterations is chosen as 200, and all noise realizations are kept the same for fair comparison. The BP decoder having the best performance among single-FG decoders; i.e., the RFG is included as a reference in Figure 3.13, to observe the  $n$ -FG decoder gain over single-FG performance. For each polar code  $P(2^n, 2^{n-1})$ , the FG sets formed either by cyclic shifts or by our MaxSON and MaxofMax rules, are of size  $n$ .

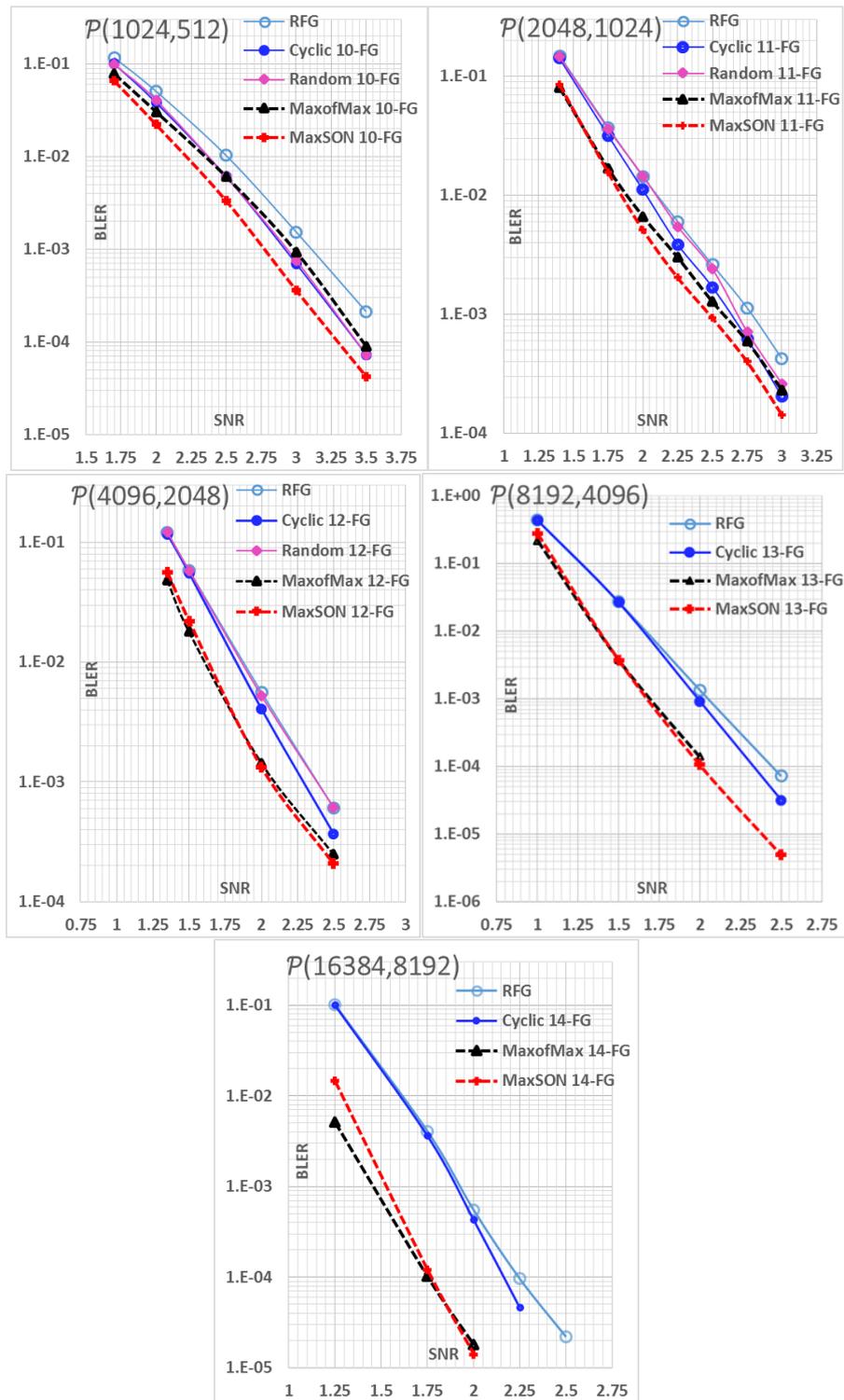


Figure 3.13. BLER performances of RFG as compared to  $n$ -FG BP decoders chosen randomly or deterministically by cyclic, MaxSON and MaxofMax rules for  $P(1024, 512)$ ,  $P(2048, 1024)$ ,  $P(4096, 2048)$ ; and by cyclic, MaxSON, MaxofMax rules for  $P(8192, 4096)$  and  $P(16384, 8192)$ .

We observe that the BLER performance of the RFG (blue curve) is the worst in all five figures of Figure 3.13 as expected, because the other BP decoders are multiple-FG decoders. However, as the code length  $N$  increases, both the cyclic  $n$ -FG (dark blue curve) and the randomly chosen (pink curve)  $n$ -FG decoders start to approach the RFG curve; and somewhat astonishingly, a randomly chosen 12-FG decoder for  $P(4096, 2048)$  performs exactly the same as the single RFG decoder. On the other hand, performances of both of our set choices (dashed black and red curves) seem quite satisfactory. They are the best ones in all cases, the MaxofMax rule of set choice being somewhat beaten by the MaxSON rule; but the inner-competition between them seems to become more driving as  $N$  grows. The gain of the MaxSON decoder (dashed red curve) over the cyclic decoder (dark blue curve) at  $\text{BLER} = 10^{-3}$  is 0.13, 0.16, 0.23, 0.32 and 0.38 dB respectively for the rate-0.5 polar codes  $P(1024, 512)$ ,  $P(2048, 1024)$ ,  $P(4096, 2048)$ ,  $P(8192, 4096)$  and  $P(16384, 8192)$ . Even though the BP decoder whose FGs are chosen by the MaxofMax rule doesn't work as effectively as the MaxSON decoder for  $n = 10$  and 11; it performs as well as the MaxSON decoder for  $n = 12, 13$  and 14. The SNR gain of the MaxofMax decoder over the cyclic decoder at  $\text{BLER} = 10^{-3}$  is 0.2, 0.29 and 0.45 dB respectively for the codes  $P(4096, 2048)$ ,  $P(8192, 4096)$  and  $P(16384, 8192)$ .

In Figure 3.14, we plot the SNR gains of the MaxSON and MaxofMax decoders over the cyclic decoder versus  $n$  for a fixed  $\text{BLER} = 10^{-3}$ . When the performance for all the code lengths is taken into account, the MaxSON decoder seems to be more beneficial than the MaxofMax decoder in general.

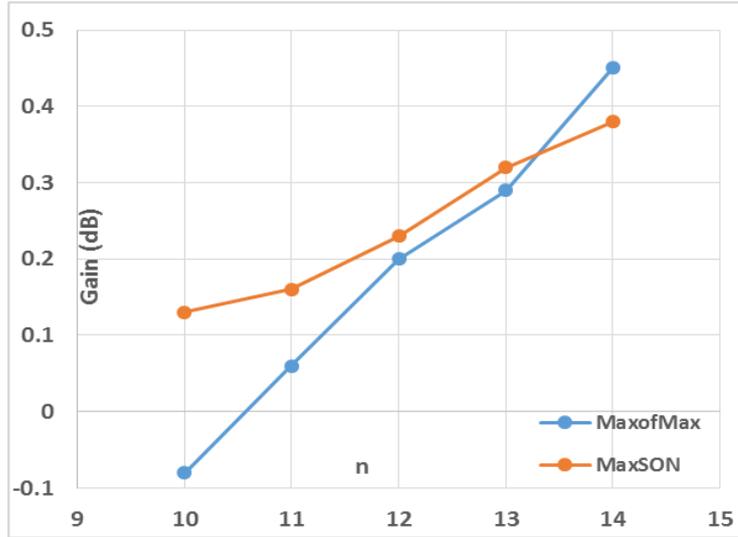


Figure 3.14. SNR gains at BLER =  $10^{-3}$  of the MaxSON and MaxofMax  $n$ -FG BP decoders over the cyclic  $n$ -FG BP decoder versus  $n$ .

### 3.4. Performance Comparison of Multiple Factor Graph BP Decoding for Polar Codes and Reed-Muller Codes

It is a well-known fact that polar codes are close cousins of the Reed-Muller (RM) codes; and they only differ by the selection rule of the rows of the  $G_2^{\otimes n}$  matrix mentioned in Section 2.1: “Polar codes select the rows of  $G_2^{\otimes n}$  so as to minimize the Bhattacharyya parameters and RM codes select them to maximize their Hamming weights.” The aim of this section is to present some experimental results showing the differences between the BP decoder performances of polar and RM codes. Since the code rate is chosen as 0.5 throughout the thesis, we simulate RM codes of rate 0.5.

First, the performances of the  $(2^n, 2^{n-1})$  RM and polar codes for different code lengths are compared in Figure 3.15 and quite surprisingly, it is observed that for the RM codes, if one increases the code length, single-FG (RFG in this case) BP decoder performance decreases. Hence, while for  $n = 7$ , RFG BP decoders for polar and RM codes have similar BLER performances; for  $n = 9$  and 11, polar RFG decoders have considerably better performances than the RM decoders.

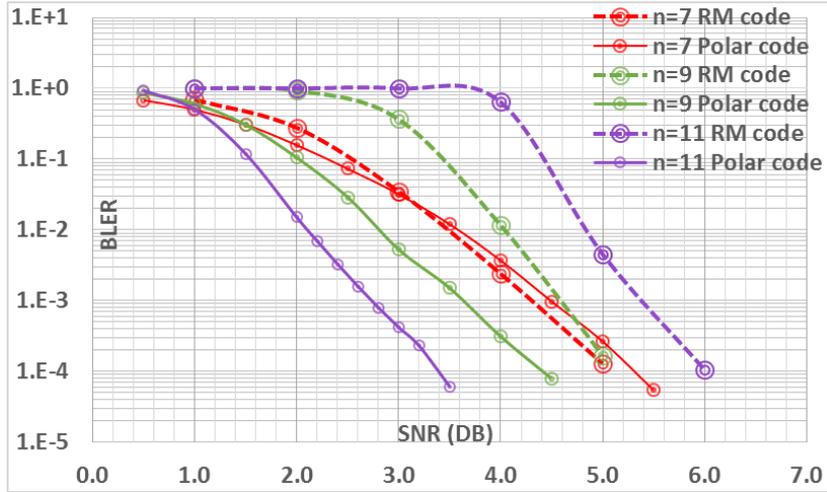


Figure 3.15. BLER of the RFG decoder for polar (solid lines) and RM codes (dashed lines), versus the channel SNR, where red, green and purple curves are corresponding respectively to  $(N, K) = (128, 64)$ ,  $(512, 256)$ , and  $(2048, 1024)$  codes.

In order to explore the contribution of multiple-FG decoding, we plot the 120-FG performance of RM codes for  $n = 9$ , in Figure 3.16. Set choice of the 120-FG BP decoder is done by slightly modifying the MaxofMax algorithm; and the first 5 stages are permuted instead of 4, so as to generate  $5! = 120$  factor graphs.

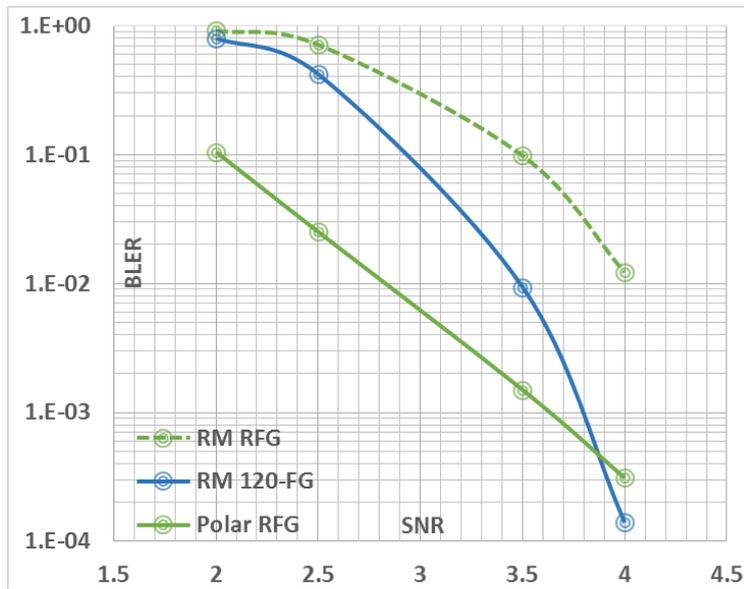


Figure 3.16. BLER performance comparison between the RFG and 120-FG BP decoders of RM codes and the RFG decoder of polar codes for  $(N, K) = (512, 256)$ .

Results of Figure 3.16 show that the RFG BP decoder for the (512, 256) polar codes over-performs the 120-FG BP decoder for the (512, 256) RM codes at low SNRs. However, the performance of the 120-FG decoder gets better in the high SNR region.

Figure 3.16 leads one to investigate the required number of decoders for the RM code to achieve the polar code's performance. Since the number of multiple-FGs is very high for longer code lengths, a practical value of  $n = 7$  is chosen in order to cover all possible FGs in the multiple-FG decoder set. So,  $7! = 5040$  FGs are used for comparison with the earlier studies of Peker, performed for a BEC, where multiple-FG BP decoders are selected among the maximum equi-FV sets up to 152 FGs [Peker, 2018]. In Figure 3.17, we compare the performances of 5040-FG decoders (ranked in decreasing order of SON), for the RM and adaptive polar codes designed over a  $\text{BEC}(\epsilon)$  with the erasure probability  $\epsilon = 0.35$ . (Note that the adaptive polar code designed for  $\epsilon = 0.35$  turns out to be the same as the polar code designed for  $\epsilon = 0.5$  in this case.)

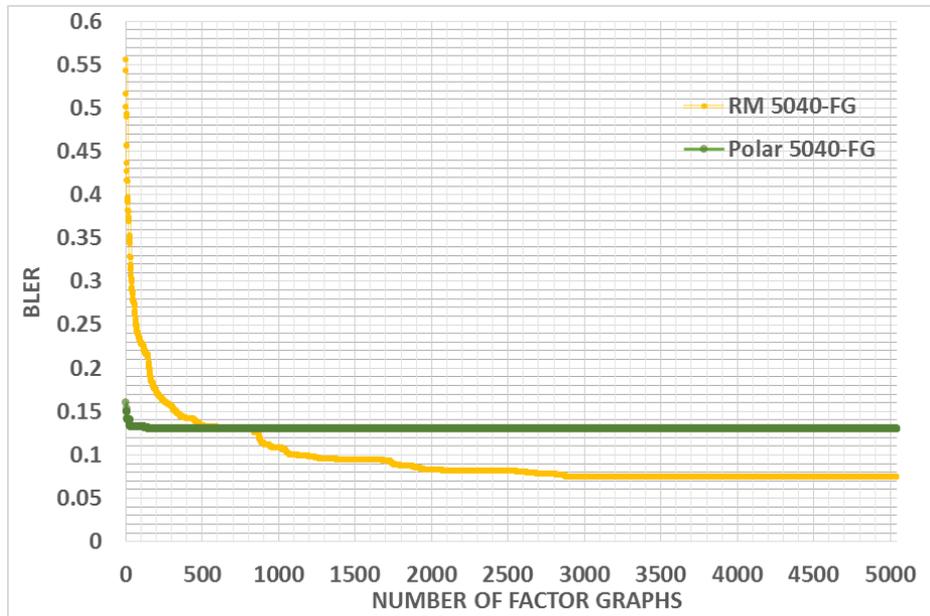


Figure 3.17. BLER performance comparison of multiple-FG BP decoders for the RM codes and adaptive polar codes for  $(N, K) = (128, 64)$  over a  $\text{BEC}(\epsilon)$  with  $\epsilon = 0.35$ .

In Figure 3.17, which shows the simulation results over a BEC( $\epsilon$ ) with erasure rate  $\epsilon = 0.35$ , the multiple-FG BP decoder for the RM code starts to perform better than the polar multiple-FG BP decoder, if the number of FGs is greater than 850. However, the trend with increasing  $n$  (shown in Figure 3.15) of deteriorating BP decoding performance for the RM codes should also be taken into consideration.

We also perform a similar simulation over an AWGN at SNR = 2 dB. In Figure 3.18, simulation results plotted for multiple-FG BP decoding of the adaptive polar code (designed at 2 dB) seem to be better than that of the RM code, if the number of FGs is less than approximately 2500; but both performances reach the same point at the end of the experiment. The performance of the multiple-FG BP decoder of the polar code designed at 0 dB is better than that of RM code till  $M$  is approximately 250 but for higher values of  $M$ , it is quite similar to that of the RM code as well.

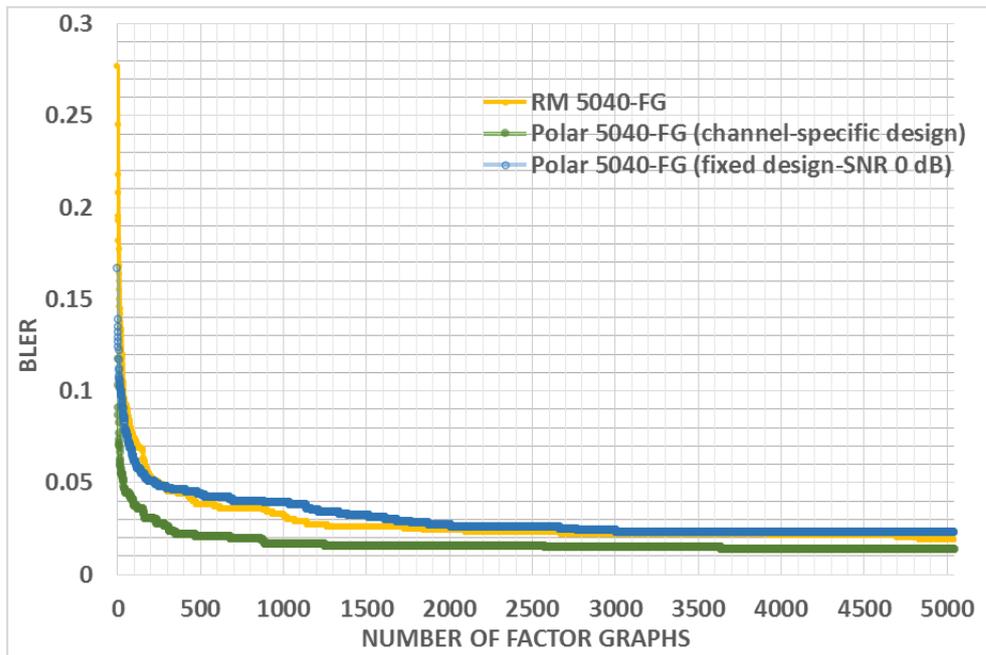


Figure 3.18. BLER performance comparison of multiple-FG BP decoders for the RM codes and polar codes with parameters  $(N, K) = (128, 64)$  over an AWGN at SNR = 2dB.

### 3.5. Performance Comparison with Other Multiple Factor Graph BP Decoders over the AWGN

Belief propagation decoding using factor graphs with permuted stages is extensively discussed in the literature [Doan et al., 2018], [Elkelesh et al., 2018a], [Elkelesh et al., 2018b], [Hashemi et al., 2018]. In this section, we compare the performance of our set-choice proposals (MaxSON and MaxofMax) with some of them, by using polar codes constructed both at some constant design-SNRs and according to the specific channel SNR.

One of these studies carried out for  $P(1024, 512)$  by Doan, Hashemi, Mondelli and Gross compares the performances of multiple-FG decoders such as PBP-CS (meaning “permuted BP decoder with cyclic shift” that we call “cyclic 10-FG” in our work), and PBP-B10 (meaning “permuted BP decoder with the best 10 FGs”, whose 10 FGs are chosen by Doan et al. beforehand to maximize the probability of successful decoding), and SCL32 (successive cancellation-list decoder of list size 32), with the single-FG decoders such as BP (that we call the RFG in our work) and SC (successive cancellation) decoders [Doan et al., 2018].

In Figure 3.19, we plot our performance results (dashed curves) obtained by the RFG, cyclic, MaxofMax and MaxSON decoders, together with Doan et al.’s results (solid curves) for the polar code  $P(1024, 512)$ , for which the design SNR is not stated. So, we compare their results with four choices of the design-SNR, three of them being constant and the last one being variable design SNRs. One of the constant design SNRs that we pick is Arikan’s suggestion ( $Z_0 = 0.5$  corresponding to  $\text{SNR} = -1.59$  dB) [Arikan, 2008]; the second one is Vangala’s proposal (0 dB SNR) [Vangala et al., 2015]; and the third constant design SNR is chosen as 0.5 dB. The last performance curve in Figure 3.19 is obtained for channel-specifically designed polar codes, where we vary the design-SNR of the polar code in each simulation point, with steps of 0.5

dB (or 0.2 dB for our last cyclic, MaxSON and MaxofMax points in the last sub-figure of Figure 3.19).

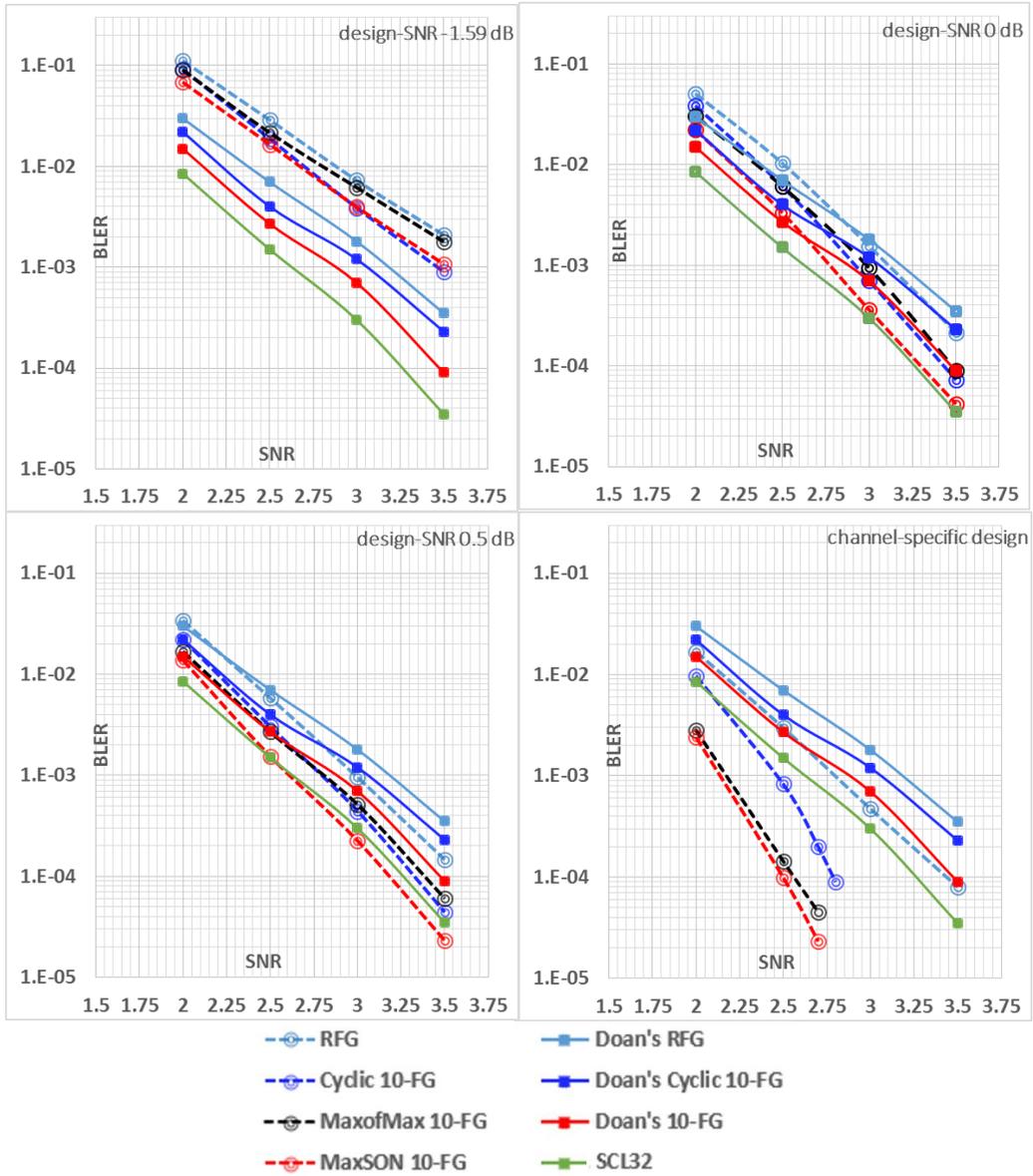


Figure 3.19. BLER performance comparison between the BP decoders of our study (shown by dashed lines) and another study in the literature reproduced from [Doan et al., 2018]) (shown by solid lines); where our first three figures are obtained for  $\mathcal{P}(1024, 512)$  constructed at fixed design-SNRs ( $-1.59$ ,  $0$ ,  $0.5$  dB) and last one is constructed with variable, channel-specific design-SNR.

One observes from the dashed curves of Figure 3.19 that the BP decoding performance of polar codes is highly dependent on the design SNR. Among the chosen four cases, our dashed simulation curves have the worst performance for the fixed design-SNR of  $-1.59$  dB ( $Z_0 = 0.5$ ) and the best performance for the channel-specific design with variable design-SNR. When the polar codes are designed adaptively at each channel SNR, it can be seen from the last sub-figure of Figure 3.19 that all of our dashed curves become better. However, the fact that our BP decoding algorithm uses perfect knowledge based (PKB) decisions for early termination causes our BLER curves be better than would be obtained in practice. Since Doan et al. use a practical decision criterion, a fair comparison is not possible at this point. Therefore, we delay our comments on performance comparison to the end of this chapter, where we add a practical criterion for early termination.

Two other studies on multiple-FG BP decoders are carried out for  $P(2048, 1024)$  codes by Elkelesh, Ebada, Cammerer and Brink in 2018[Elkelesh et al., 2018a], [Elkelesh et al., 2018b]. The design of polar codes used by Elkelesh et al. is according to Arıkan’s suggested method for AWGNs [Arıkan, 2009]. In order to find a comparable design-SNR with Elkelesh et al.’s work, we plot the BLER performances polar codes we’ve designed at different design-SNRs in Figure 3.20.

Elkelesh et al.’s single BP decoder is the same as our RFG BP decoder, and their BP list decoder, BPL [Elkelesh et al., 2018a] with a list size of  $L = 10$  is a multiple FG decoder, whose FGs are created with random permutations; so, while comparing its performance with that of our cyclic 11-FG decoder, one may expect a worse BLER curve similar to  $P(2048, 1024)$  curves of Figure 3.13. On the other hand, their BPL [Elkelesh et al., 2018b] decoder with a list size of  $L = 32$  includes randomly generated ( $32 - 11 = 21$ ) FGs, in addition to the 11 cyclic permutations of the RFG. So, it is expected to perform better than the cyclic 11-FG decoder and our MaxSON 11-FG decoder.

Since our decoders use perfect knowledge based (PKB) decisions for early termination, we avoid comparison with the results in the literature, until we add a practical stopping criterion to our BP decoding as in Figures 3.21, 3.22 and 3.23.

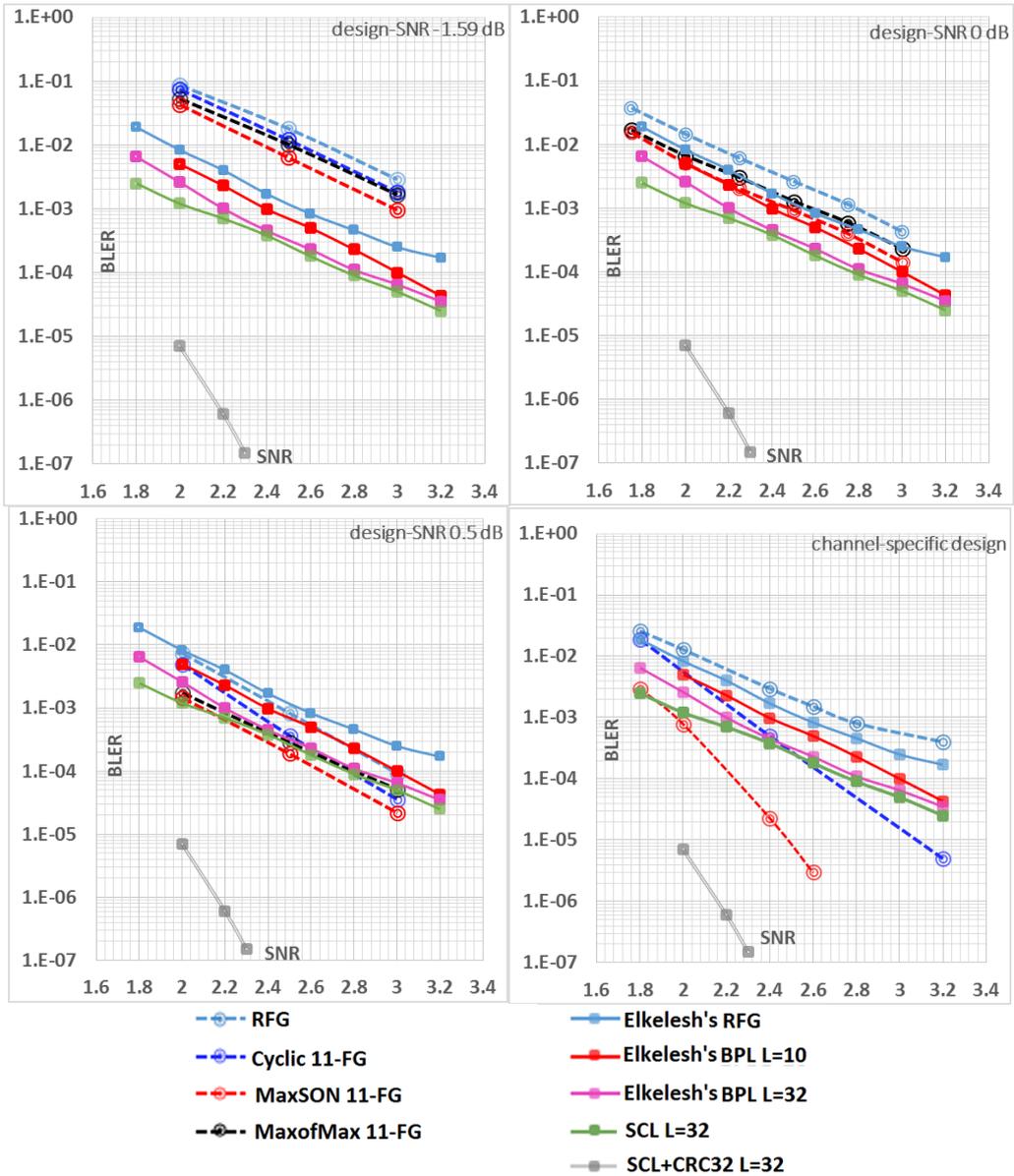


Figure 3.20. BLER performance comparison for  $P(2048, 1024)$ , constructed at different design-SNRs, between the decoders of our study (shown by dashed lines) and another one in the literature (shown by solid lines, reproduced from [Elkelesh et al., 2018a] and [Elkelesh et al., 2018b]).

The last sub-figure of Figure 3.20 corresponds to the channel-specific design that is expected to work the best; and it seems to result in quite satisfactory performance for the MaxSON 11-FG decoder. However, rather unpredictably, the RFG decoder of the channel-specifically designed  $P(2048, 1024)$  code has worse performance than that of the fixed 0.5 dB design. This is also verified in Appendix C, in which the performance comparison between our RFG and MaxSON decoders is given for different design SNRs.

As emphasized above, our simulation results presented up to this point demonstrate the BLER performances of the proposed BP decoders, stopped by early decisions based on the perfect knowledge of the transmitted bits. Since in practical applications the receiver doesn't know the transmitted bits; the promising performance of the MaxSON decoder should be tested under a practical decision criterion as well. For this purpose, we repeat some simulations, and instead of the PKB decisions of the previous simulations, we stop the BP decoding algorithm by a cyclic redundancy check (CRC) decision. We generate  $r$  bits of CRC for  $K - r$  input bits, at the cost of rate loss  $= r/N$ . At each step of 10 BP iterations, we check whether the CRC of input bits is satisfied. Decoding stops when either the CRC is fulfilled, or the maximum number of 200 iterations is reached. The CRC polynomial of length 16 that we use is  $x^{16} + x^{12} + x^5 + 1$ , and the corresponding rate loss is reflected to the SNR in the horizontal axis of the related figures.

Figure 3.21 and Figure 3.22 show the BLER performance comparison of the polar codes under perfect knowledge based (PKB) and CRC based early termination algorithms for polar codes of lengths  $N = 1024$  and  $2048$ . Dashed lines show the PKB performances of the RFG, cyclic and MaxSON decoders and the solid lines of corresponding colors show the CRC16 based practical decisions. Green lines display the SCL32 (successive cancellation-list decoder of list size 32) decoder performances. We select the design-SNR of 0.5 dB in both cases, in order to keep the performances

of the RFGs in ours (found by CRC16 based decisions) and others [Doan et al., 2018], [Elkelesh et al., 2018a], [Elkelesh et al., 2018b] comparable, so that a base of reference is provided for the multiple-FG performance comparison of the proposed BP decoders.

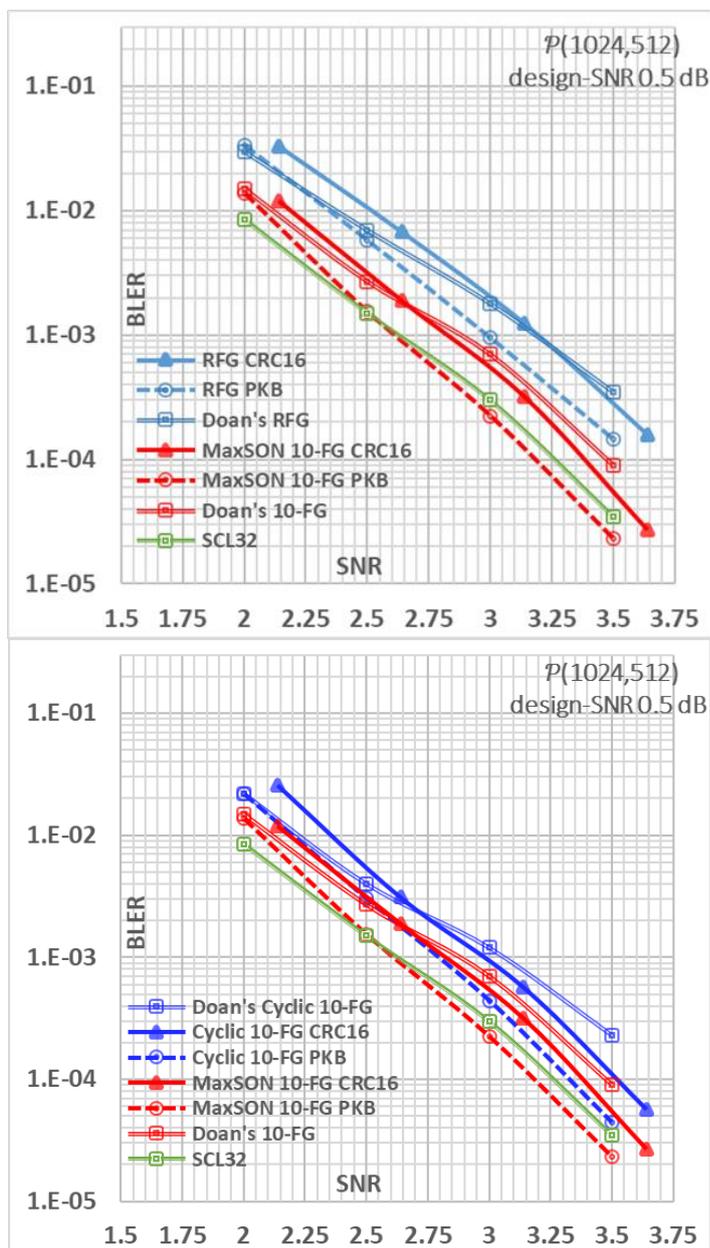


Figure 3.21. BLER performance for  $P(1024, 512)$ , constructed at 0.5 dB design-SNR, of our RFG, cyclic and MaxSON decoders using perfect knowledge based (PKB) and CRC based early termination, (Doan's curves are reproduced from [Doan et al., 2018]).

One observes from Figure 3.21 that Doan’s RFG (blue) and cyclic 10-FG (dark blue) decoders have very similar performance to ours obtained with CRC16 decisions, which forms a fair comparison base. Although Doan’s 10-FG and our MaxSON 10-FG (with CRC16) decoders also have similar performance, ours seems more preferable because of its deterministic MaxSON rule of choosing 10 FGs, instead of Doan et al.’s experimental method of selecting 10 FGs to maximize the probability of successful decoding. The performance loss of the CRC16 decisions over PKB decisions is around 0.18 dB; approximately 0.15 dB of which corresponding to the rate loss ( $r/N = 16/1024$ ) penalty of CRC16 that is reflected to the horizontal axis. So, with CRC16 based decisions, the MaxSON 10-FG decoder is observed to no longer compete with the SCL32 decoder, which is only possible with PKB decisions for  $N = 1024$ .

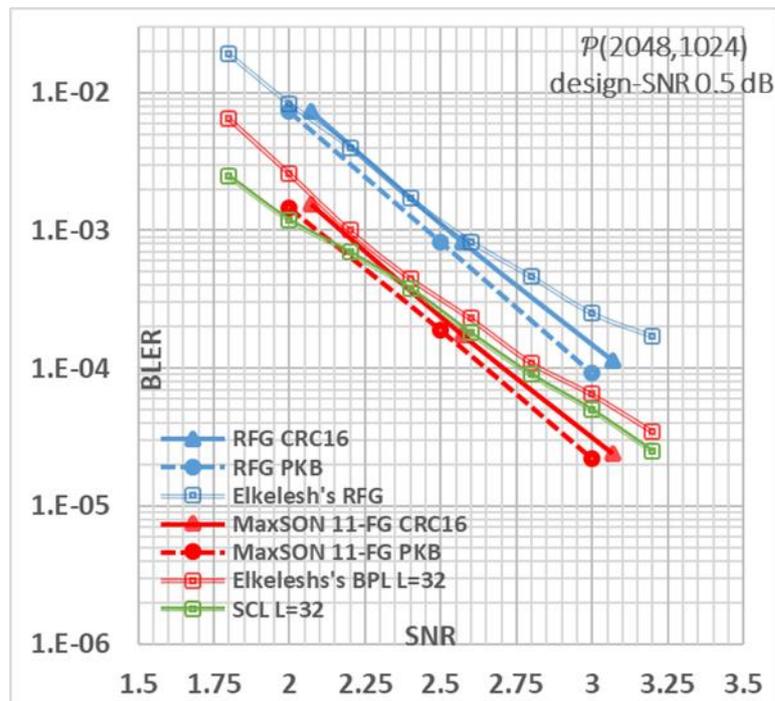


Figure 3.22. BLER performance for  $P(2048, 1024)$ , constructed at 0.5 dB design-SNR, of our RFG and MaxSON decoders using perfect knowledge based (PKB) and CRC based termination, (Elkelesh’s curves are reproduced from [Elkelesh et al., 2018a] and [Elkelesh et al., 2018b]).

In Figure 3.22 that is plotted for  $N = 2048$ , again a fair comparison base is achieved because of the matching performances of Elkelesh's RFG and our RFG (with CRC16 decisions). Since the CRC rate loss ( $r/N = 16/2048$ ) penalty that is reflected to the horizontal axis ( $\sim 0.08$  dB) is smaller than the  $N = 1024$  case, the performance of the MaxSON 11-FG (CRC16) decoder is now comparable to that of the SCL32 decoder for  $N = 2048$ . This performance looks quite promising since the multiple-FG BP decoder has the potential of reaching much higher speeds, because of the parallel implementation of multiple factor graphs.

Finally, we plot the BLER and BER performances of the RFG and MaxSON decoders for  $P(1024, 512)$  and  $P(2048, 1024)$  in Figure 3.23. Solid lines display the BLER and dashed lines show the BER performances. We plot the BER curves of the MaxSON decoders by considering two different FGs in the multiple FG set. Whenever a received block remains undecoded, we either reflect it to the BER of the  $K - r$  information bits found by 200 iterations of the first FG in the set, that is the RFG, or to the information bits found by the last FG within the set of the MaxSON decoder. Of course, the second BER is expected to be higher, because the individual performance of the last FG is quite bad; but we wonder how much this difference is. It can be seen from Figure 3.23 that the BER performance of the last FG is so bad that it is almost equal to the BER performance of the RFG decoder. Although for the MaxSON decoder, the number of undecoded blocks is less than that of the RFG decoder, the BER found for the last FG approaches the BER of the single-FG performance of the RFG. On the other hand, the BER performance of the MaxSON decoder calculated by the first FG (i.e., the RFG in the  $n$ -FG set) is better than the single-FG performance of the RFG decoder by up to 0.5 dB at a BER value of  $10^{-5}$ .

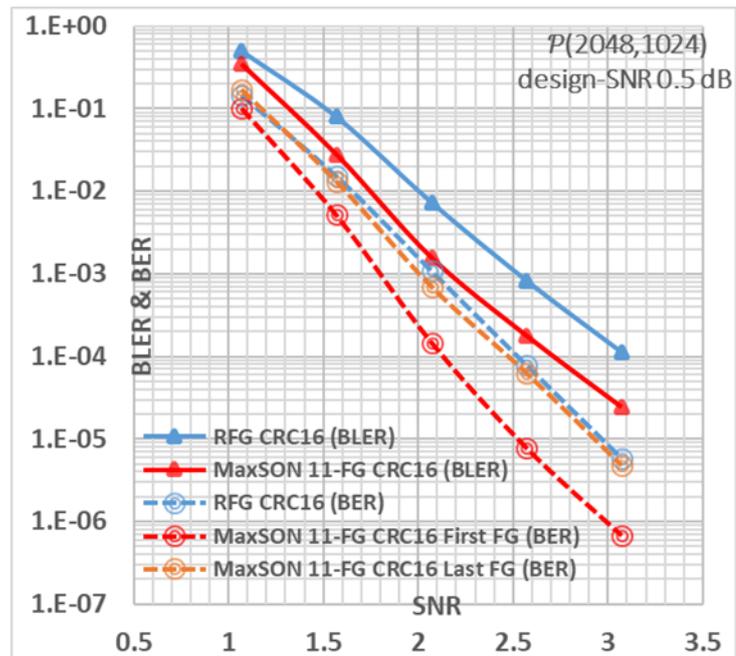
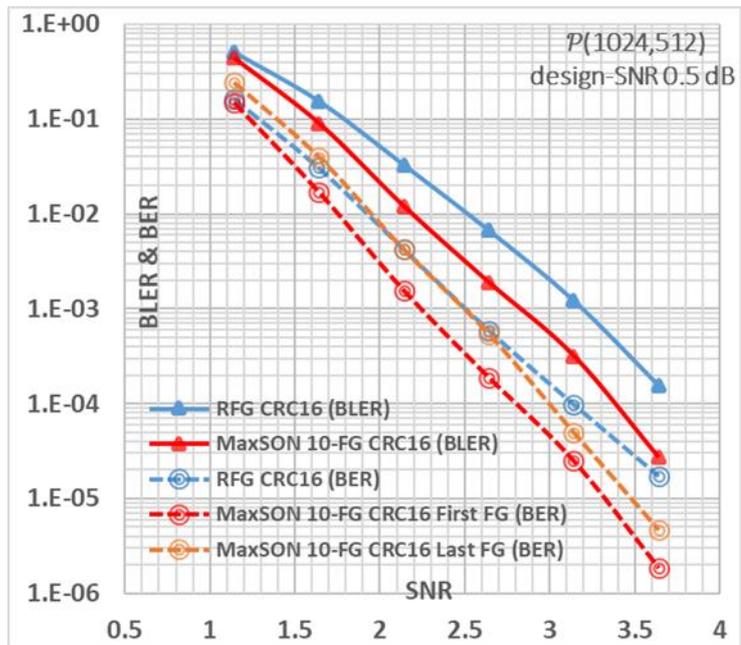


Figure 3.23. BLER (solid lines) and BER(dashed lines) performances for  $P(1024, 512)$  and  $P(2048, 1024)$ , constructed at 0.5 dB design-SNR, between our RFG and MaxSON decoders using CRC16 based early termination.



## CHAPTER 4

### CONCLUSION

In this thesis, belief propagation (BP) decoders for rate-0.5 polar codes designed over binary-input additive white Gaussian noise channels are studied, also glancing at the BP decoder performance of the Reed-Muller codes in a few cases. For a polar code  $P(N, K) = P(2^n, 2^{n-1})$ , there are  $n!$  different factor graph (FG) representations whose decoding performances may differ; so, we explore the problem of finding better performing FG sets for the multiple-FG belief propagation decoder, by simulations performed for  $n \leq 14$  and  $N \leq 16384$ .

Early studies have shown that among single-FG BP decoders, the reference factor graph (RFG) BP decoder performance is one of the best and the inverse-RFG (IRFG) performance is the worst [Doğan, 2015], [Peker, 2018], [Akdoğan, 2018]. Exploring the performance variation of the RFG and the IRFG BP decoders for polar codes designed over the AWGN channels at 0 dB, one of our noteworthy observations is that, as the code length increases from 64 to 4096, the RFG decoder performance improves whereas the IRFG performance degrades. Hence the polar code  $P(4096, 2048)$  performs as the best single-FG BP decoder; but its IRFG decoder, quite unpredictably, performs as the worst one. On the other hand, while the polar code  $P(64, 32)$  gives the anticipated worst RFG performance, it has the best IRFG performance. Our simulation results show that the gain of the RFG decoder over the IRFG decoder increases linearly with the increasing code length  $n$ , from 0.6 dB for  $n = 6$  to 7.6 dB for  $n = 12$ , at the BLER =  $10^{-2}$ .

Since the gap between the RFG and the IRFG performance increases with length, multiple-FG BP decoder sets combined of FGs with random stage permutations don't

function well for long code lengths. In other words, when the code length  $N = 2^n$  increases, a random FG chosen among  $n!$  different factor graphs has a smaller chance of performing effectively. Hence, the total performance of a randomly chosen multiple-FG set decreases with a high probability.

Noticing that random selection is not a good option for generating multiple-FG sets of BP decoders, we aim at a wiser method to create  $n$ -FG decoders. Observing the performance of all  $n!$  single-FG BP decoders for  $P(64, 32)$  over an AWGN channel at 1.5 dB versus their stage order numbers (SONs) (introduced by Akdoğan [Akdoğan, 2018]); we propose two different  $n$ -FG decoder selection algorithms, namely the MaxSON and MaxofMax rules explained on pages 53 and 54.

For polar codes  $P(2^n, 2^{n-1})$  with  $6 \leq n \leq 14$ , designed over an AWGN at 0 dB, the performances of the MaxSON and MaxofMax decoders are compared to the cyclic and randomly constructed  $n$ -FG decoders, which are frequently mentioned in the literature [Korada, 2009], [Hussami et al., 2009], [Elkelesh et al., 2018a], [Elkelesh et al., 2018b], [Doan, Hashemi, Mondelli, Gross, 2018]. It is observed that both rules of choosing  $n$  FGs that we propose give much better results than those of the cyclic  $n$ -FG decoder. The SNR gain of our MaxSON  $n$ -FG decoder over the cyclic decoder increases with the increasing code length  $n$ , and approaches to 0.38 dB at BLER =  $10^{-3}$ , for  $P(16384, 8192)$  and  $n = 14$ .

When the BP decoding performances of the polar codes and Reed-Muller (RM) codes are compared; our results show that the single-FG (RFG) performance of a polar code is much better than that of the corresponding RM code, and the gap between their performances increases with the code length. However, with multiple-FG BP decoders, polar and RM code performances are observed to approach each other as the number of utilized FGs increases.

Set choice methods of MaxSON and MaxofMax that we present in this study are observed to compete with similar multiple-FG belief propagation decoders in the literature [Doan et al., 2018], [Elkelesh et al., 2018a] and [Elkelesh et al., 2018b], in terms of not only performance, but also the ease in their formation that uses only the SONs of the FGs.

Since we mainly use perfect knowledge based (PKB) decisions for early termination, the BLER curves found in most simulations of this study form lower bounds to those which would be obtained in practice. In order to compare the performance of our decoders with the results in the literature, we modify the early termination decisions of the BP decoding algorithm and use a 16-bit cyclic redundancy check (CRC16) instead of the PKB decisions, in the last three figures of Chapter 3. Our simulations for  $P(1024, 512)$ , with CRC16 yielding a rate loss of  $r/N = 16/1024$  that is reflected to the SNR axis as a 0.15 dB-penalty, show that the MaxSON 10-FG decoder has equivalent performance to Doan's 10-FG decoder that uses the best 10 FGs chosen by performing pre-simulations of single-FG decoding to find the most successful FGs. We reason that the MaxSON decoder, whose FG set is chosen deterministically depending on the stage order numbers (SONs), should be more preferable since optimization by pre-simulations may complicate the initialization of the BP decoding practice. However, our MaxSON 10-FG decoder cannot reach the block error ratio performance of SCL decoder with list size of 32 for  $P(1024, 512)$  under CRC based decoding algorithm, which looks reasonable considering the more complicated decoding of SCL32.

On the other hand, for  $P(2048, 1024)$ , the smaller SNR penalty of 0.08 dB resulting from the use of CRC16 (with rate loss  $r/N = 16/2048$ ) allows our  $n$ -FG BP decoders be more competitive. The MaxSON 11-FG decoder has the same performance with Elkelesh et al.'s BPL decoder that uses 32 FGs, 11 of them taken from the cyclic set and the remaining 21 are chosen randomly. More importantly, Figure 3.22 on page 69 emphasizes that the performance of our MaxSON 11-FG BP decoder reaches to that

of SCL32 decoder. This performance looks quite promising since the multiple-FG BP decoder has the potential of reaching much higher speeds, because of the parallel implementation of multiple factor graphs.

As a future work, it is of interest to explore whether the performance of some cyclic redundancy check aided BP list decoders (which will perhaps be abbreviated as CA-BPL decoders) would compete with today's attention-receiving-performance of the successive cancellation list (CA-SCL) decoders.

## REFERENCES

- [Akdoğan, 2018] Akdoğan, Ş. (2018). A study on the set choice of multiple factor graph belief propagation decoders for polar codes, MSc. Thesis.
- [Alamdar-Yazdi & Kschischang, 2011] Alamdar-Yazdi, A., Kschischang, F. R. (2011). A simplified successive-cancellation decoder for polar codes. In *IEEE Communications Letters*, 15(12): 1378–1380.
- [Alsan, 2014] Alsan, M. (2014), Universal polar decoding with channel knowledge at the encoder. In *IEEE Information Theory Workshop (ITW)*, *arXiv preprint arXiv: 1311.7590*.
- [Arıkan, 2008] Arıkan, E. (2008). A performance comparison of polar codes and reed-muller codes. *IEEE Communications Letters*, 12(6): 447–449.
- [Arıkan, 2009] Arıkan, E. (2009). Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels. *Information Theory, IEEE Transactions on*, 55(7):3051–3073.
- [Arıkan, 2010] Arıkan, E. (2010). Polar codes: A pipelined implementation. In *Proc. 4th ISBC*, pages 11–14.
- [Berrou et al., 1993] Berrou, C., Glavieux, A., & Thitimajshima, P. (1993). Near Shannon limit error-correcting coding and decoding: Turbo-codes. *Proceedings of ICC '93 - IEEE International Conference on Communications*, 2(1), 1064–1070.
- [Bonik et al., 2012] Bonik, G., Goreinov, S., Zamarashkin, N. (2012). Construction and analysis of polar and concatenated polar codes: practical approach. *arXiv preprint arXiv: 1207.4343*.
- [Bose and Ray-Chaudhuri, 1960] Bose, R. C. and Ray-Chaudhuri, D. K. (1960). On a class of error correcting binary group codes. *Information and control*, 3(1):68–79.
- [Cammerer et al. 2017] Cammerer, S, Leible, B., Stahl, M., Hoydis, J. and Brink, S. T. (2017). Combining belief propagation and successive cancellation list decoding of polar codes on a GPU platform. in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, pages 3664-3668.

- [Chandesris et al., 2018] Chandesris, L., Savin, V. and Declercq, D. (2018). Dynamic-SCFlip decoding of polar codes. *IEEE Trans. Commun.*, 66(6): 2333-2345.
- [Doan et al., 2018] Doan, N., Hashemi, S., Mondelli, M., Gross W. (2018). On the Decoding of Polar Codes on Permuted Factor Graphs. *arXiv preprint arXiv: 1806.11195*.
- [Doğan, 2015] Doğan, O. (2015). An investigation on belief propagation decoding of polar codes, MSc. Thesis.
- [Elias, 1955] Elias, P. (1955). Coding for noisy channels. In *IRE Conv. Rec. 4*, pages 37-46.
- [Elkelesh et al., 2018a] Elkelesh, A., Ebada, M., Cammerer, S., Brink, S. (2018). Belief propagation decoding of polar codes on permuted factor graphs. *arXiv preprint arXiv: 1801.04299v3*
- [Elkelesh et al., 2018b] Elkelesh, A., Ebada, M., Cammerer, S., Brink, S. (2018). Belief propagation list decoding of polar codes. *arXiv preprint arXiv: 1806.10503v1*.
- [Eslami & Pishro-Nik, 2010] Eslami, A., & Pishro-Nik, H. (2010). On bit error rate performance of polar codes in finite regime. *Communication, Control, and Computing (Allerton), 2010 48th Annual Allerton Conference on*.
- [Eslami & Pishro-Nik, 2013] Eslami, A., Pishro-Nik, H. (2013). On finite-length performance of polar codes: Stopping sets, error floor, concatenated design. In *IEEE Trans. Commun.*, 61(3): 919–929.
- [Forney, 2001] Forney, G., D. (2001). Codes on graphs: normal realizations. In *IEEE Trans. Info. Theory*, 47(2):520-548.
- [Gallager, 1962] Gallager, R. G. (1962). Low-density parity-check codes. *Information Theory, IRE Transactions on*, 8(1):21–28.
- [Giard et al., 2014] Giard, P., Sarkis, G., Thibeault, C., Gross, W. J. (2014). A fast software polar decoder. In *IEEE Int. Conf. on Acoustics, Speech, and Signal Process (ICASSP)*.
- [Golay, 1949] Golay, M. J. (1949). Notes on digital coding.
- [Hamming, 1950] Hamming, R.W. (1950). Error detecting and error correcting codes. *The Bell System Technical Journal*, vol. 29, 147-160.

- [Hashemi et al., 2018] Hashemi, S., Doan, N., Mondelliy, M., Gross W. (2018). Decoding Reed-Muller and polar codes by successive factor graph permutations. *arXiv preprint arXiv: 1807.03912v1*.
- [Hassani and Urbanke, 2014] Hassani, S. H. and Urbanke, R. (2014). Universal polar codes. In *IEEE International Symposium on Information Theory (ISIT)*, pages 1451–1455.
- [Hussami et al., 2009] Hussami, N., Korada, S., Urbanke, R. (2009). Performance of polar codes for channel and source coding. In *Proc. IEEE ISIT*, pages 1488–1492.
- [Kern et al., 2014] Kern, D., Vorkoper, S., Kuhn, V. (2014). A new code construction for polar codes using min-sum density. In *International Symposium on Turbo Codes and Iterative Information Processing (ISTC)*, pages 228–232.
- [Korada, 2009] Korada, S. B. (2009). Polar codes for channel and source coding. *PhD thesis*.
- [Li and Yuan, 2013] Li, H. and Yuan, J. (2013). A practical construction method for polar codes in AWGN channels. In *TENCON Spring Conference*, pages 223–226.
- [MacKay and Neal, 1995] MacKay, D. J. and Neal, R. M. (1995). Good codes based on very sparse matrices. In *Cryptography and Coding*, pages 100–111. Springer.
- [Mori and Tanaka, 2009a] Mori, R. and Tanaka, T. (2009a). Performance of polar codes with the construction using density evolution. *IEEE Communication Letters*, 13(7): 519-521.
- [Mori and Tanaka, 2009b] Mori, R. and Tanaka, T. (2009b). Performance and construction of polar codes on symmetric binary-input memoryless channels. *International Symposium on Information Theory (ISIT)*, pages 1496–1500.
- [Muller, 1954] Muller, D. E. (1954). Application of Boolean algebra to switching circuit design and to error detection. *Electronic Computers, Transactions of the IRE Professional Group on*, (3):6–12.
- [Niu & Chen, 2012] Niu, K., Chen, K. (2012). CRC-aided decoding of polar codes. In *IEEE Communications Letters*, 16(10): 1668–1671.
- [Pamuk, 2011] Pamuk, A. (2011). An FPGA implementation architecture for decoding of polar codes. In *Wireless Communication Systems (ISWCS), 2011 8th International Symposium on*, pages 437–441.

- [Peker, 2018] Peker, G. (2018). Belief propagation decoding of polar codes under factor graph permutations. MSc. Thesis METU.
- [Reed and Solomon, 1960] Reed, I. S. and Solomon, G. (1960). Polynomial codes over certain finite fields. *Journal of the society for industrial and applied mathematics*, 8(2):300–304.
- [Sarkis & Gross, 2013] Sarkis, G., Gross, W. J. (2013). Increasing the throughput of polar decoders. In *IEEE Communications Letters*, 17(4): 725–728.
- [Sarkis et al., 2013] Sarkis, G., Giard, P., Vardy, A., Thibeault, C., Gross W. J. (2013). Fast polar decoders: algorithm and implementation. *arXiv preprint arXiv:1306.6311*.
- [Sasoglu & Wang, 2014] Sasoglu, E., Wang, L. (2014). Universal polarization. In *IEEE International Symposium on Information Theory (ISIT)*, pages 1456–1460.
- [Shannon, 1948] Shannon, C. (1948). A mathematical theory of communication. *The bell system technical journal* 27(3): 379-423 and 623–656.
- [Tal and Vardy, 2011] Tal, I. and Vardy, A. (2011). List decoding of polar codes. In *Information Theory Proceedings (ISIT), 2011 IEEE International Symposium on*, pages 1–5. IEEE.
- [Tal and Vardy, 2013] Tal, I. and Vardy, A. (2013). How to construct polar codes. In *IEEE Transactions on Information Theory*, 59(10): 6562–6582.
- [Tong et al., 2005] Tong, S., Wang, P., Wang, D., Wang, X. (2005). Box-minus operation and application in sum product algorithm. In *Electronic Letters*, (4): 197-198.
- [Trifonov, 2012] Trifonov, P. (2012). Efficient design and decoding of polar codes. *IEEE Transactions on Communications*, 60(11): 3221-3227.
- [Vangala et al., 2015] Vangala, H., Viterbo, E., & Hong, Y. (2015). A comparative study of Polar Code constructions for the AWGN channel. *arXiv preprint arXiv:1501.02473v1*.
- [Wiberg, 1996] Wiberg, N. (1996). Codes and decoding on general graphs Citeseer.
- [Wiberg et al., 1995] Wiberg, N., Loeliger, H.-A., and Kötter, R. (1995). Codes and iterative decoding on general graphs. *European Transactions on telecommunications*, 6(5):513–525.

- [Wu et al., 2014] Wu, D., Li, Y., Sun, Y. (2014). Construction and block error rate analysis of polar codes over AWGN channel based on Gaussian approximation. *IEEE Communication Letters*, 18(7): 1099-1102.
- [Xu et al., 2015] Xu, J., Che, T., & Choi, G. (2015). XJ-BP: Express journey belief propagation decoding for polar codes. *2015 IEEE Global Communications Conference, GLOBECOM 2015*.
- [Yazdi and Kschischang, 2011] Yazdi, A., A., Kschischang, F., R. (2011). A simplified successive-cancellation decoder for polar codes. In *IEEE Communication Letters*, 15(12): 1378-1380.
- [Yu et al., 2018] Yu, Q., P., Shi, Z., P., Deng, L., li, X. (2018). An improved belief propagation decoding of concatenated polar codes with bit mapping. *IEEE Comm. Letters*, 22(6): 1160-1163.
- [Yu et al., 2019] Yu, Y., Pan, Z., Liu, N. and You, X. (2019). Belief propagation bit-flip decoder for polar codes. *IEEE Access*, vol 7:10937-10946.
- [Yuan and Parhi, 2013] Yuan, B. and Parhi, K. (2013). Architecture optimizations for BP polar decoders. *Proc. 38th IEEE ICASSP*, pages 2654–2658.
- [Yuan and Parhi, 2014a] Yuan, B. and Parhi, K. (2014). Early stopping criteria for energy-efficient low-latency belief-propagation polar code decoders. *Signal Processing, IEEE Transactions on*, 62(24):6496–6506.
- [Yuan and Parhi, 2014b] Yuan, B. & Parhi, K. (2014). Algorithm and architecture for hybrid decoding of polar codes. in *Proc. 48th Asilomar Conf. Signals, Syst. Comput.* pages 2050-2053.
- [Zhang et al., 2014] Zhang, Y., Lui, A., Pan, K., Gong, C., and Yang, S. (2014). A practical construction method for polar codes. *IEEE Communication Letters*, 18(11): 1871-1874.
- [Zhang et al., 2017] Zhang, Z., Qin, K., Zhang, L., Zhang, H., and Chen, G., T. (2017). Progressive bit-flipping decoding of polar codes over layered critical sets. *Proc. IEEE Global Comm. Conf. (GLOBECOM)*, pages 1-6.
- [Zhang et al., 2018] Zhang, Z., Qin, K., Zhang, L., and Chen, G. T. (2018). Progressive bit-flipping decoding of polar codes: A critical-set based tree search approach. *IEEE Access*, pages 57738-57750.
- [Zhao et al., 2011] Zhao, S., Shi, P., Wang, B. (2011). Designs of Bhattacharyya parameter in the construction of polar codes. In *Wireless Communications, Networking and Mobile Computing (WiCOM)*, pages 1-4.



## APPENDICES

### A. RFG over IRFG BP Decoder Gain for Polar Code Designed over Binary Erasure Channels

It may be of interest to examine whether the BP decoders of the polar codes designed over BECs behave similarly to those designed over AWGN channels; so we produce Figure A.1 and Figure A.2 for a visual comparison with Figure 3.5 and Figure 3.4 respectively.

One observes from Figure A.1 that the RFG and IRFG decoder performances over BECs are not as separated as in the AWGN channels of Figure 3.5. Moreover, RFGs as well as IRFGs are more clustered in their groups, as opposed to the AWGN channel case. Doubling the code length seems to produce no gain for the passage of RFG decoders from  $n = 6$  to 7, or the passage of IRFG decoders from  $n = 8$  to 7, as a result of this clustering.

Figure A.2 shows that while channel-specific design shows better performance than fixed-design erasure rate of 0.5 for the RFG decoder, performance remains the same for the IRFG BP decoder of  $\mathcal{P}(1024, 512)$  codes.

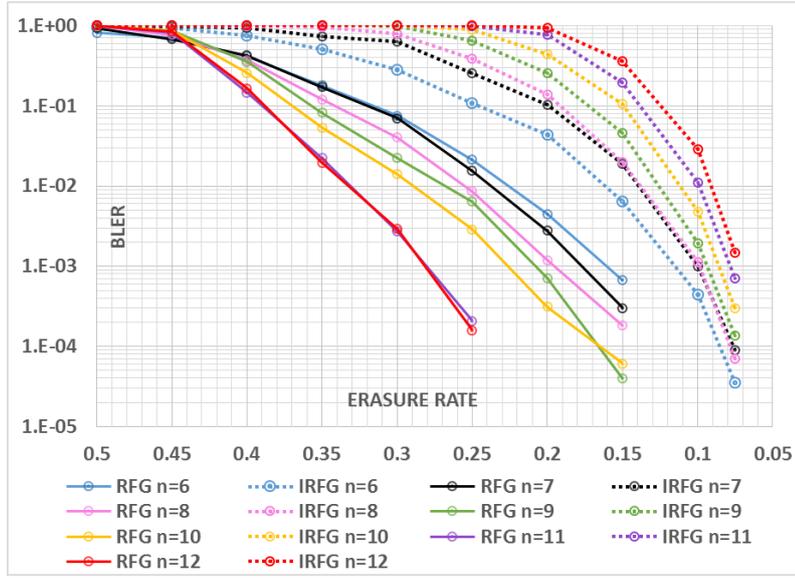


Figure A.1. BLER of the RFG and IRFG decoders versus the channel erasure rate. Solid lines refer to RFG and dotted ones of the same color refer to IRFG performances for the same code lengths. Blue curves:  $n = 6$ ,  $P(64, 32)$ , Black curves:  $n = 7$ ,  $P(128, 64)$ , Pink curves:  $n = 8$ ,  $P(256, 128)$ , Green curves:  $n = 9$ ,  $P(512, 256)$ , Yellow curves:  $n = 10$ ,  $P(1024, 512)$ , Purple curves:  $n = 11$ ,  $P(2048, 1024)$ , Red curves:  $n = 12$ ,  $P(4096, 2048)$ .

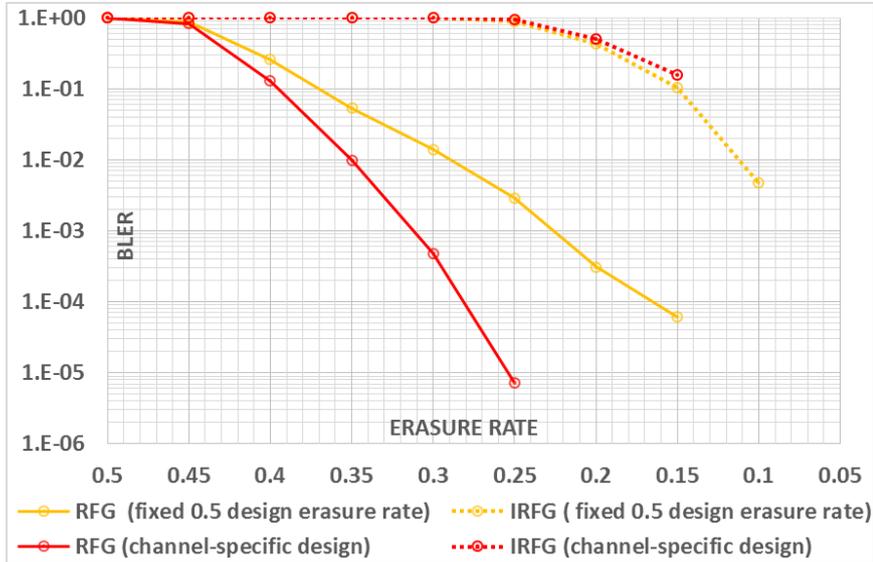


Figure A.2. BLER performances of the RFG and IRFG BP decoders for  $P(1024, 512)$  codes, constructed using i) fixed design-erasure-rate of 0.5, ii) channel-specific design at each erasure-rate.

## B. Some Considerations about Multiple-FG Set Choice

Simulation results suggest that the gain accomplished by the multiple-FG decoder is due to the correct combination of different stage permutations in the  $n$ -element sets of FGs. So, we focus on polar codes  $P(1024, 512)$  and  $P(2048, 1024)$  to analyze the individual effects of single-FG performances of the  $n$  factor graphs in MaxofMax and MaxSON decoders.

In Figure B.1 we try to show the effects of individual 1-FG performances of the FGs in the  $n$ -element BP decoder sets, which are formed either by cyclic shifts or by the MaxSON and MaxofMax rules; to the overall  $n$ -FG performance, for  $P(1024, 512)$  at SNR = 1.7 dB and  $P(2048, 1024)$  at SNR = 1.4 dB. Each row in the figure indicates the individual 1-FG BP decoding BLERs of the  $n$  factor graphs, which belong to the sets chosen by the MaxofMax and MaxSON rules (the cyclic set obtained by cyclic shifts of the RFG stages is also included in the last column for  $P(1024, 512)$ ). BLERs are found over the same  $10^4$  AWGN output words at a certain SNR, with the same randomizing noise seed.

The plots on the left of Figure B.1 show the performances of the  $n$ -FG BP decoders. On the tables, rows circled by blue curves have similar 1-FG performances and their contribution to the multiple-FG decoder performance is similar. The rows circled by red curves, show a noteworthy feature, that even a worse 1-FG performing component of the  $n$ -element FG set may contribute to the  $n$ -FG decoder performance (see the 6<sup>th</sup> and 7<sup>th</sup> elements of the MaxSON set contributing to the  $n$ -FG performance, more than the same numbered elements in the MaxofMax set having better 1-FG performances). Therefore, we conclude that the combination of good performing single-FG decoders may not always result in a performance increment of the multiple-FG decoder, which is an observation supported in the literature as well [Akdoğan, 2018]. After the BLER

of the  $n$ -FG decoder reduces to a certain point, the rows circled by black curves suggest that contributions from the last elements of the  $n$ -FG sets are not appreciable and the amount of the contribution seems not depending significantly on the individual 1-FG performances. Besides, bad performing single-FG decoders, which approximately have BLERs of more than 0.5 do not contribute to multiple-FG decoder performance at all. The same observation can also be made by examining the 1-FG performances of the cyclic decoder for  $P(1024, 512)$ , in which the factor graphs used after the 4<sup>th</sup> FG seem to have very small contributions to the performance of the 10-FG decoder.

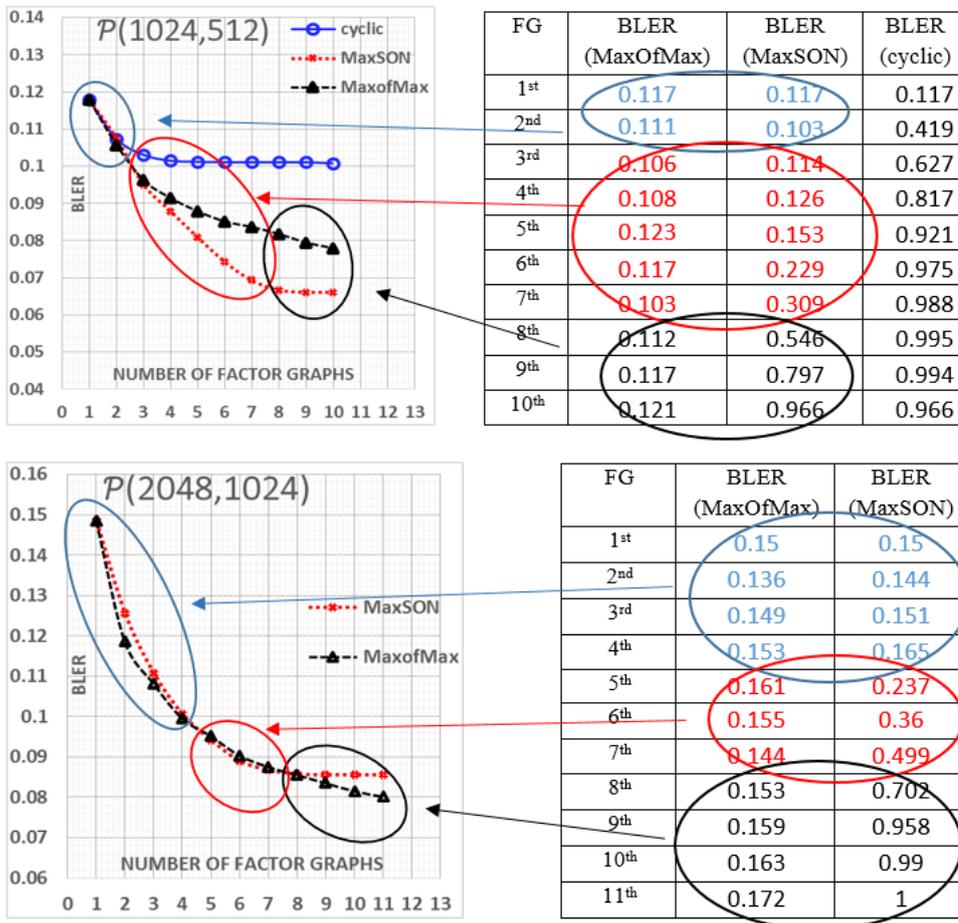


Figure B.1. Effects of individual 1-FG performances of the FGs in the  $n$ -element BP decoder sets (formed either by cyclic shifts or by the MaxSON and MaxofMax rules), to the overall  $n$ -FG performance; for  $P(1024, 512)$  at SNR = 2 dB and  $P(2048, 1024)$  at SNR = 1.7 dB.

### C. BLER Comparison of Some BP Decoders over the AWGN for Polar Code Constructions Using Fixed or Channel-Specific Design-SNRs

We compare the BP decoding performances of the polar codes designed with respect to a fixed design SNR of  $-1.59$ ,  $0$ ,  $0.5$  and  $1$  dB, to those of the adaptive polar codes designed with respect to the specific SNR of the AWGN channel over which they are used. Chosen BP decoders are the single RFG and MaxSON  $n$ -FG, for  $P(1024, 512)$  and  $P(2048, 1024)$  in Figures C.1 and C.2 respectively.

One observes in Figure C.1 that using channel-specific design for  $P(1024, 512)$  improves the MaxSON decoder performance much more than it affects the RFG decoder. The SNR gain of the “MaxSON 10-FG decoder over the RFG” in case of the channel-specific design is more than twice of the “MaxSON over the RFG” gain corresponding to the fixed  $0$  dB-design.

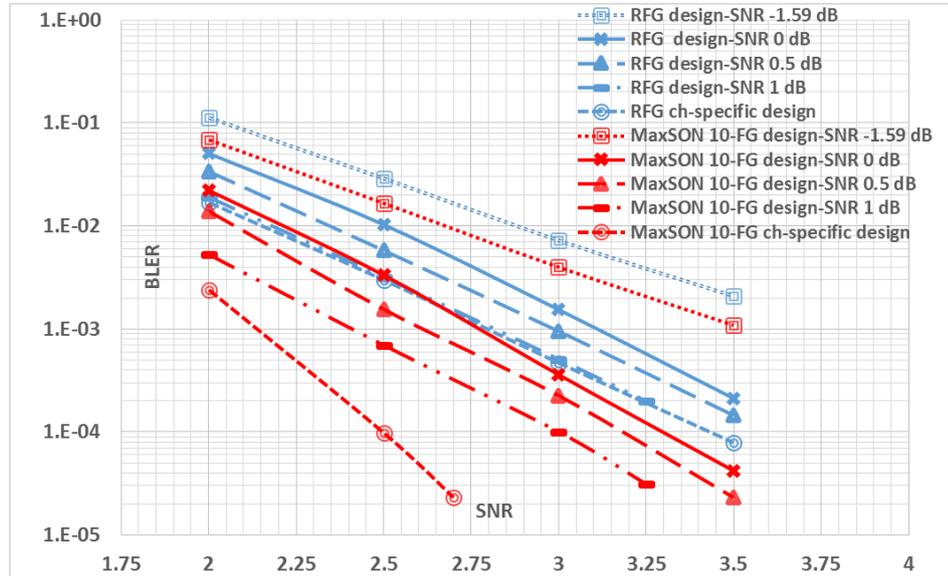


Figure C.1. BLER performances of multiple-FG BP decoder performances for  $P(1024, 512)$  codes, constructed using either fixed design-SNR of  $0$  dB (solid curves), or channel-specific design-SNR (dashed curves).

RFG decoder performance increases with the design-SNR in Figure C.1, and its performance for a fixed design SNR of 1 dB is similar to that of the channel-specific design. However, in case of the MaxSON 10-FG decoder, the performance of the channel-specific design outperforms that of the 1 dB design-SNR by 0.25 dB at BLER =  $10^{-3}$  and by 0.5 dB at BLER =  $10^{-4}$ .

For  $P(2048, 1024)$ , we have confusingly seen that the RFG decoder performance for the channel-specific design is almost the same as that of the fixed design-SNR of 0 dB; whereas the 0.5 and 1 dB designs are better. More expectedly, the MaxSON 11-FG decoder appears to be the best for the channel-specific design; however, the fixed design SNR of 1 dB performs very close to it.

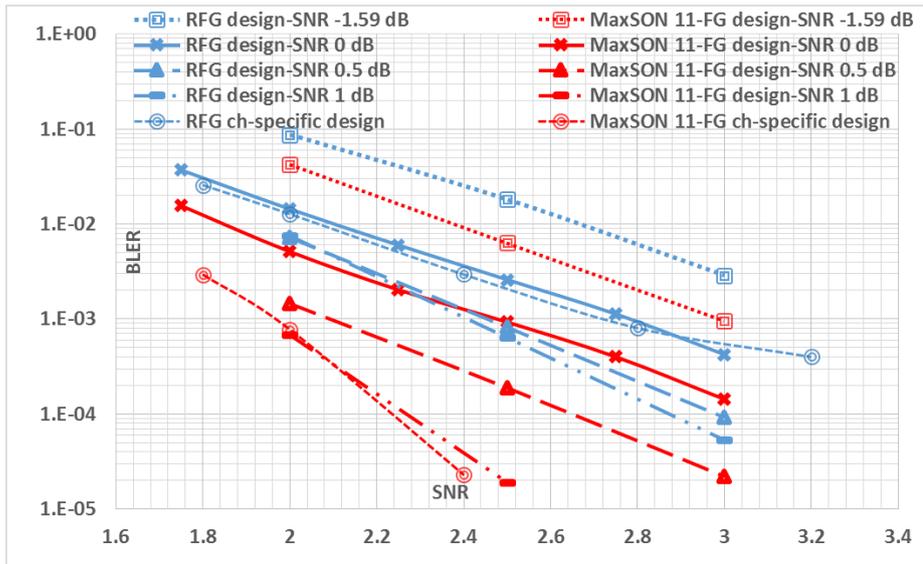


Figure C.2. BLER performances of multiple-FG BP decoder performances for  $P(2048, 1024)$  codes, constructed using either fixed design-SNR of 0 dB (solid curves), or channel-specific design-SNR (dashes curves).