The Effect of Error Propagation on the Performance of Polar Codes Utilizing Successive Cancellation Decoding Algorithm

Alia A. Andi[1], **Orhan Gazi**[2]

[1] Electronic and Communication Engineering Department, Cankaya University, Ankara, Turkey
[2] Electronic and Communication Engineering Department, Cankaya University, Ankara, Turkey
*corresponding author, E-mail: alia_eletri@yahoo.com

Abstract

In this paper, we discuss and analyze the effect of error propagation on the performance polar codes decoded using the successive cancellation algorithm. We show that error propagation due to erroneous bit decision is a catastrophic issue for the successive cancellation decoding of polar codes. Even a wrong decision on a single bit may cause an abundance of successor bits to be wrongly decoded. Furthermore, we observe that the performance of polar codes is significantly improved if even single bit errors are detected and corrected before the decoding of successor bits.

1. Introduction

Polar codes can achieve the capacity discrete memoryless channels, such as binary erasure channel (BEC), and binary symmetric channel. Polar codes are decoded using the successive cancellation (SC) algorithm which is introduced in [1]. The complexity of the SC decoding algorithm is on the order of $O(NlogN)$ where N is the code-word length such that $N = 2^n, n \in \mathbb{N}$. Hardware architectures for successive cancellation list decoding of polar codes in the log-likelihood ratio domain have been presented in [2]. In [3], the improved versions of the successive cancellation decoding algorithm, the successive cancellation list (SCL) and the successive cancellation stack (SCS) decoding methods, which show better performance for polar codes without increasing the code lengths, are introduced. A new SC-based decoding algorithm called SC flip, which preserves the low memory requirements of original SC decoder is studied in [4]. In [5], the authors focus on short length polar codes and present a method which can enhance the performance of the successive cancellation decoder. Systematic polar codes, which are used in channel estimation in [6], are introduced in [7].

In [1], Arıkan introduced the belief propagation (BP) decoding of polar codes. It is pointed out in [8] that soft information exchange in BP decoders play an important role for channels other than binary erasure channels. In [9], early stopping criteria are utilized for BP decoders to reduce the computation complexity. In [10], authors proposed a hybrid decoding approach which makes use of both BP decoding and SC decoding, and in this approach, when BP decoder fails SC algorithm is utilized. Error floor problem of polar codes employing BP decoder is alleviated using the multi-trellis approach in [11]. Early stopping criteria is employed in [12] for the decoding of polar codes employing permuted factor graphs. Deep learning based methods for the decoding of polar codes is used in [13]. Complexity reduction of BP decoders eliminating unnecessary node calculations is studied in [14].

Recently it is shown in [15]-[16] that polar codes utilizing SC list decoders concatenated with cyclic redundancy check (CRC) codes outperforms both turbo and low density parity check codes. In [17], the authors proposed a concatenated structure consisting of SC stack (SCS) and CRC codes, and it is shown in [17] that SCS with CRC achieves almost the same performance of turbo and LDPC codes. In [18] a method to find the optimal CRC polynomial is proposed, and it is shown that the optimal CRC outperforms the standard CRC given in [19]. To improve the code performance, the use of CRC is constrained to only those bits [20] which generate the minimum Hamming weight polar code-words.

From hardware implementation point of view, to decrease the decoding latency of the SC decoders which arises from the sequential decoding operations, a semi-parallel decoding architecture with low latency is proposed in [21] where resource sharing is employed. In [22], an overlapped computation with a precomputation approach is proposed to reduce the decoding latency and improve the throughput. It is reported in [22] that the throughput is doubled compared to the throughput of [21]. Another low latency SC decoder is proposed in [23] where it is shown that a decoding latency of 88% for a rate of 0.7 with block length 218 is achieved for polar codes.

In [24]-[25], concatenated code structures involving polar codes as inner codes and Reed-Solomon codes as outer codes are proposed. Convolutional codes are concatenated with polar codes in [26] where it is shown that slightly better performance is achieved sacrificed decoding complexity. In [27]-[28], low-density parity-check codes (LDPC) and polar codes are concatenated, and it is shown that the resulting code remedies the error floor problem of the LDPC codes.

In this paper, we analysis the effect of error propagation on the performance of successive cancellation (SC) decoding of polar codes. The remainder of this paper is organized as follows. In Section II, the concept of polar encoding and SC decoding is briefly visited. In Section III, we discuss and analysis the effect of error propagation on the performance of SC decoding. Simulation results are presented in Section IV. Finally, the conclusions are drawn in Section V.

2. Polar Codes and Successive Cancelation Decoding Algorithm

2.1. Polar code construction

A polar code is constructed considering N independent copies of a discrete binary memoryless channel W. These channels are combined in a recursive manner, followed by channel spliting stage [1], [4]. The split channels with large capacities are used for the transmission of information bits whereas low capacity split channels are used for frozen bits, i.e., parity bits.

For the construction of a polar code of rate

$$R \triangleq \frac{K}{N}, 0 < K < N$$

we choose the K split channels out of N split channels with large capacities. These K channels are called non-frozen channels, and they are used for the transmission of information bits. The remaining channels are not used for carrying information and these channels are called frozen channels. A polar code-word is obtained using $x_1^N = u_1^N G_N$ where G_N is the generator matrix and u_1^N is an N-bit vector consisting of K information bits and $N - K$ frozen bits. We denote the set of frozen channel indices by A^c and the set of non-frozen channel indices by A [1], [4].

2.1.1. The Kernel Encoding and Decoding Units

Kernel Encoding Unit:

The kernel encoding unit of the polar code is depicted in Figure 1.

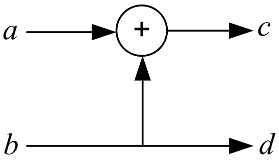


Figure 1: The Kernel encoding unit.

From Figure 1, we can write that

$$c = a \oplus b \qquad d = b.$$

where $\bar{u} = [a\ b]$ is the data vector and $\bar{x} = [c\ d]$ is the polar code-word. The polar encoding operation can be described using

$$\bar{x} = \bar{u}G \qquad (1)$$

where G is the generator matrix, and its value for $N = 2$ equals to

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

The generator matrix for an arbitrary N value is constructed using the formula

$$G_N = B_N F^{\otimes n}$$

where we have

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

and B_N is calculated in a recursive manner using

$$B_N = R_N (I_2 \otimes B_{N/2})$$

where

$$B_2 = I_2$$

and R_N is the odd even permutation matrix.

R_N denotes the $N \times N$ reverse shuffle permutation matrix defined by

$$(s_1, s_2, s_3, \cdots, s_N) R_N = (s_1, s_3, \cdots s_{N-1}, s_2, s_4, \cdots s_N).$$

For instance, for (s_1, s_2, s_3, s_4) the matrix R_4 is found from

$$(s_1, s_2, s_3, s_4) R_4 = (s_1, s_3, s_2, s_4)$$

as

$$R_4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Since

$$(s_1, s_2, s_3, s_4) \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = (s_1, s_3, s_2, s_4).$$

Kernel Decoding Unit:

The kernel decoding unit of Figure 1 is depicted in Figure 2 where $\hat{c}$ and $\hat{d}$ denote the received signals.

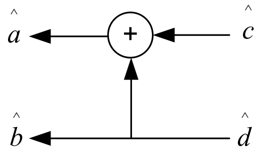


Figure 2: The Kernel decoding unit.

Using Figure 2, we can write that

$$\hat{a} = \hat{c} \oplus \hat{d} \qquad \hat{b} = \hat{d}$$

from which we can write the probabilities

$$P(\hat{a}=0)=P(\hat{c}=0\,)P(\hat{d}=0)\\ +P(\hat{c}=1\,)P(\hat{d}=1) \tag{2}$$

$$P(\hat{a}=1)=P(\hat{c}=0\,)P(\hat{d}=1)\\ +P(\hat{c}=1\,)P(\hat{d}=0). \tag{3}$$

$LR(\hat{a})$ is defined as

$$LR(\hat{a})=\frac{P(\hat{a}=0)}{P(\hat{a}=1)} \tag{4}$$

in which employing (2) and (3), we get

$$LR(\hat{a})=\frac{1+LR(\hat{c})LR(\hat{d})}{LR(\hat{c})+LR(\hat{d})}. \tag{5}$$

Following a similar approach, we can express the likelihood $LR(\hat{b})$ as

$$LR(\hat{b})=[LR(\hat{c})]^{1-2\hat{a}}LR(\hat{d}). \tag{6}$$

The formulas in (5) and (6) can be recursively expressed for those polar decoders constructed for $N>2$ in [1].

Polar Encoding Unit for $N=4$:

The polar encoding unit for $N=4$ can be constructed using the Kernel encoding units as shown in Figure 3.

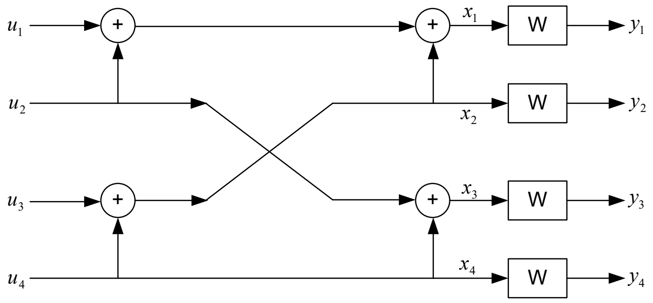


Figure 3: Polar encoding unit for $N=4$.

The decoding unit of Figure 3 can be drawn as in Figure 4.

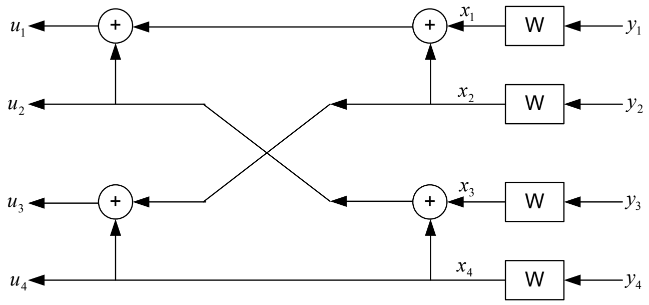


Figure 4: Polar decoding unit for $N=4$.

The decoding path of the bit u_1 using Figure 4 can be drawn as in Figure 5 where it is seen that the decoding path shown by bold lines can be drawn as a tree as shown in the lower part of Figure 5.

In Figure 5, the likelihood ratios for g_1 and g_2 are calculated as

$$LR(g_1)=\frac{1+LR(x_1)LR(x_2)}{LR(x_1)+LR(x_2)}$$

$$LR(g_2)=\frac{1+LR(x_3)LR(x_4)}{LR(x_3)+LR(x_4)}$$

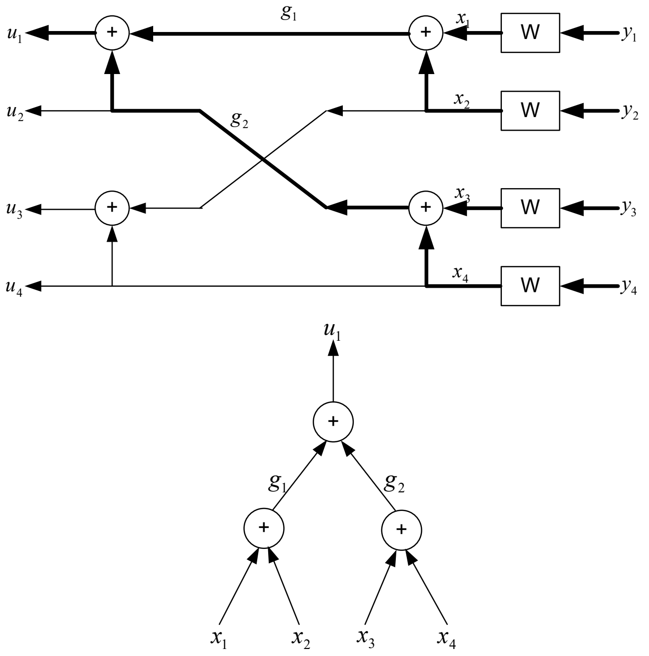


Figure 5: Decoding path of u_1 for $N=4$ and its tree representation.

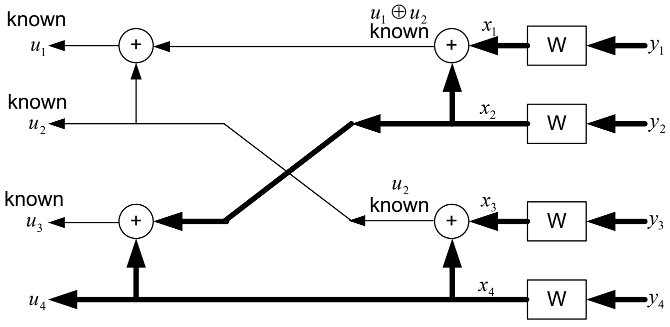


Figure 6 Decoding path of u_4 for $N=4$.

The decoding path of u_4 shown by bold lines in Figure 6 can be redrawn in a tree structure as shown in Figure 7.

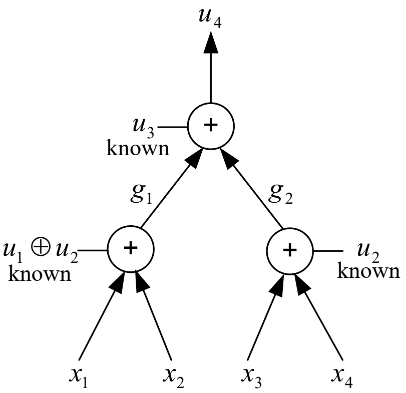


Figure 7 Decoding tree of u_4 for $N = 4$.

In Figure 7, the likelihood ratios for g_1, g_2 and u_4 are calculated as

$$LR(g_1) = [LR(x_1)]^{1-2(u_1 \oplus u_2)} \times LR(x_2)$$

$$LR(g_2) = [LR(x_3)]^{1-2u_2} \times LR(x_4)$$

$$LR(u_4) = [LR(g_1)]^{1-2u_3} \times LR(g_2).$$

When the tree structure in Figure 7 is inspected, we see that u_2 appears more in node likelihood calculations than the other data bits. In fact, we can generalize this as follows. The even indexed data bits appear more in number than the odd indexed data bits in node likelihood calculations. For instance, the distribution of the even and odd indexed previously decoded data bits to the nodes of the decoding three for the decoding of current bit is depicted in Figure 8 where it is seen that even indexed data bits appear in node labels more than the odd indexed data bits.

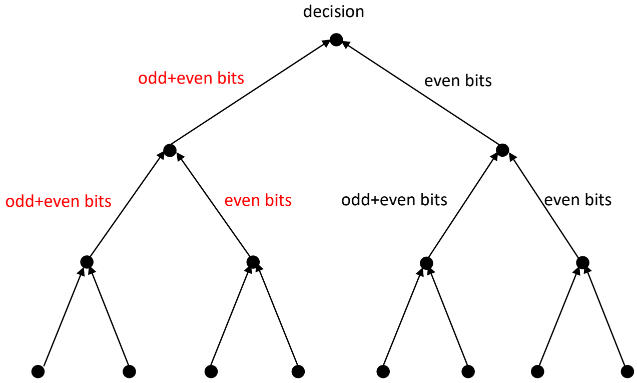


Figure 8 Decoding tree for $N = 8$.

2.2. Successive Cancellation (SC) Decoding of Polar Codes

The encoder maps the input bits $\boldsymbol{u}_1^N$ into the code-word bits $\boldsymbol{x}_1^N$, which are transmitted through the split channels $\boldsymbol{W}_N^i, \boldsymbol{1 \leq i \leq N}$, and $\boldsymbol{y}_1^N$ is the received signal. The task of the decoder is to estimate information bits $\hat{\boldsymbol{u}}_1^N$ from the received signal $\boldsymbol{y}_1^N$. The duty of the decoder is to estimate the data bits according to

$$\hat{u}_i \triangleq \begin{cases} u_i, & if\ i \in\ A^c \\ h_i(y_1^N, u_1^{i-1}), & if\ i\ \in A \end{cases} \qquad (7)$$

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

$$h_i(y_1^N, \hat{u}_1^{i-1}) \triangleq \begin{cases} 0, if\ \dfrac{W_N^{(i)}(y_1^N, \hat{u}_1^{i-1}|0)}{W_N^{(i)}(y_1^N, \hat{u}_1^{i-1}|1)} \geq 1. \\ 1, otherwise. \end{cases} \qquad (8)$$

In (8), the rational term can be defined as

$$L_N^{(i)}(y_1^N, \hat{u}_1^{i-1}) \triangleq \frac{W_N^{(i)}(y_1^N, \hat{u}_1^{i-1}|0)}{W_N^{(i)}(y_1^N, \hat{u}_1^{i-1}|1)} \qquad (9)$$

which is named as likelihood ratio (LR) [1], [4], [6]. The LRs can be recursively computed using

$$L_N^{(2i-1)}(y_1^N, \hat{u}_1^{2i-2}) = \frac{L_{N/2}^{(i)}(y_1^{N/2}, \hat{u}_{1,o}^{2i-2} \oplus \hat{u}_{1,e}^{2i-2}) L_{N/2}^{(i)}\left(y_{\frac{N}{2}+1}^N, \hat{u}_{1,e}^{2i-2}\right) + 1}{L_{N/2}^{(i)}(y_1^{N/2}, \hat{u}_{1,o}^{2i-2} \oplus \hat{u}_{1,e}^{2i-2}) + L_{N/2}^{(i)}\left(y_{\frac{N}{2}+1}^N, \hat{u}_{1,e}^{2i-2}\right)} \qquad (10)$$

and

$$L_N^{(2i)}(y_1^N, \hat{u}_1^{2i-1}) = L_{N/2}^{(i)}(y_1^{N/2}, \hat{u}_{1,o}^{2i-2} \oplus \hat{u}_{1,e}^{2i-2})^{1-2\hat{u}_{2i-1}} L_{N/2}^{(i)}(y_{N/2+1}^N, \hat{u}_{1,e}^{2i-2}) \qquad (11)$$

The formulas (10)-(11) are run for the decoding of every information bit, and as it is seen from the exponential part of (11), for the decoding of the current bit, we use the decision result of the previous decoding stage. If the previously decoded bit value is wrong, then this wrong information is used by the stages where successor bits are decoded.

Besides, when the formulas (10) and (11) are inspected, we see that the even indexed data bits appear more frequently in these formulas than the odd indexed data bits. For instance, in (10) even indexed data bits appear 4 times whereas odd indexed data bits appear 2 times. Similarly, in (11) even indexed data bits appear 2 times whereas odd indexed data bits appears once. This gives us an intuition that even indexed data bits may have more effect on the code performance.

3. Error propagation

Erroneous bit decisions in SC decoding may happen due to two factors, one is the channel noise and the other is the error propagation due to previous erroneous bit decisions. However, for the decoding of the first information bit, only channel and noise effects take role on the wrong decision.

3.1. Effect of Error propagation

In order to examine the effect of error propagation on the performance of polar codes under successive cancellation decoder over a BEC, we artificially introduced single and double bit errors and inspect the performance of the decoder considering the decoding of all the other bits.

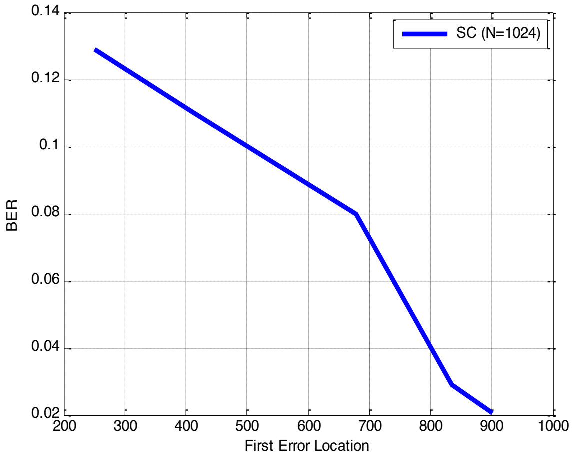
We examine error propagation in SC decoding in two ways. In the first approach, we consider the effect of single bit error location on the code performance.

In the second approach, we introduce single bit errors for the information bits having odd indices, such as $(\hat{u}_1, \hat{u}_3, \hat{u}_5, \dots\dots, \hat{u}_{N-1})$ or introduce single bit errors for the information bits having even indices such as $(\hat{u}_2, \hat{u}_4, \hat{u}_6, \dots\dots, \hat{u}_N)$ and examine its effect on the code performance. We repeated this procedure for the case of double bit errors. Besides, we also inspected the effect of single bit errors on the performance of SC list decoding of polar codes.

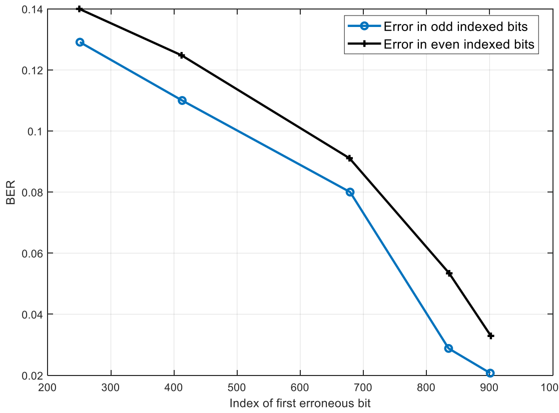
4. Simulation Results

In Figure 9A, simulation results for the effect of error propagation considering the location of single error on the performance of polar codes under successive cancellation decoder at block length 2^{10} over a BEC with erasure probability zero and rate 0.5 are shown. It is clear from Figure 9A that as the location number gets smaller values, the degrading effect of the error propagation becomes more serious.

In Figure 9B, we compare the effect of error propagation considering the location number falling into odd or even numbers. It is clear from Figure 9B that single bit errors occurring in even locations have more degrading effect on the code performance.



(A)



(B)

Figure 9 (A)The effect of error propagation on performance of polar codes under SC decoder at block length 2^{10} over a BEC with erasure probability zero and rate 0.5. First error location is the index of the data for which wrong decision is made for the first time. Small error locations have more degrading effect. (B) The effect of first error location on performance of polar codes under SC decoder at block length 2^{10} over a BEC with erasure probability zero and rate 0.5. Even locations have more degrading effects.

We also considered the effects of double errors. The double error locations are chosen in a such a way that the one location is the successor of the other, for instance, two even locations can be selected as 400 and 4002, and two odd locations can be selected as 401 and 403. In Figure 10, the effects of double errors on even locations only, on odd locations only, and one-bit error on even and one-bit error on location are considered. It is clear from Figure 10 that double errors at even locations have the most degrading effect, and double errors at odd locations have less degrading effect on the code performance.

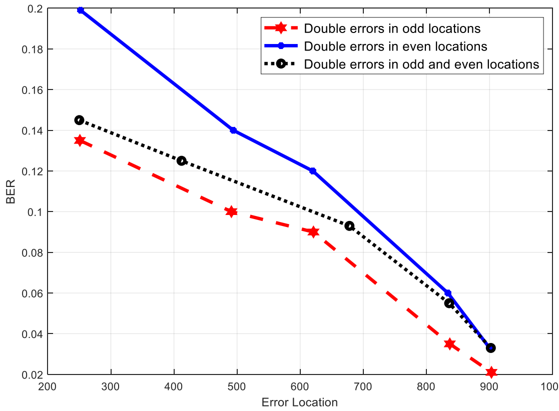


Figure 10 The effects of double errors on even locations only, on odd locations only, and one at even and one at odd locations.

In Figure 11, we compare the effects of single and double errors at even and odd locations. It is clear from Figure 10 that the occurrence of a single error at an odd location almost has the same effect as the occurrence of double errors at odd locations. However, this is not the case for errors occurring at even locations. Double errors occurring at even locations have more degrading effect than a single error occurring at an even location.

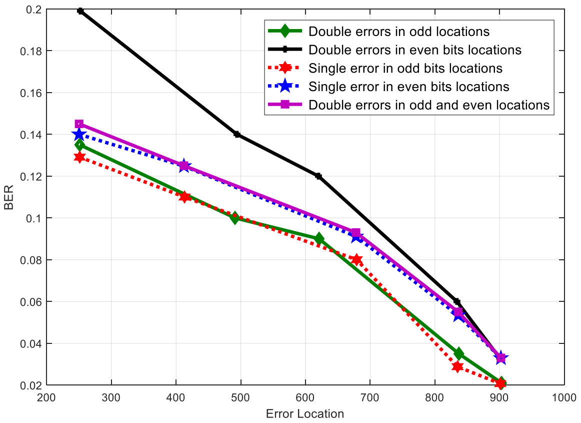
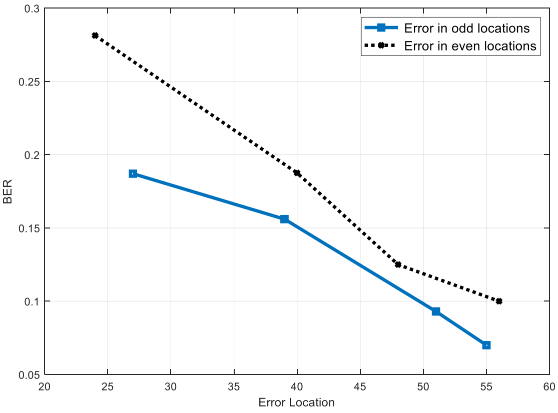


Figure 11 Comparison of the effects of single and double errors on even and odd locations.

We also inspected the effects of error propagation for SC list decoding of polar codes. The simulation results are depicted in Figure 12 where it is clear that for SC list decoders, errors occurring at even locations have more degrading effects than the errors occurring at odd locations.



(C)

Figure 12 Single error propagation in SCL, $Rate = 0.5$, $N = 64$.

5. Conclusion

In this paper, we inspected the effect of error propagation on the performance of SC and SC list decoding of polar codes. The results demonstrate that, the performances of SC and SC list decoding depend on the locations of the bit errors, and error locations with low indices have more degrading effect. This is due to the propagation of the error; since more successors bits are affected, worse performance is obtained. Besides, the single or double bit errors occurring in even locations have more degrading effect on the code performance compared to the bit single and double bit errors at odd locations. Besides, single and double bit errors occurring on the odd locations almost have the same degrading effect on SC decoder's performance.

Acknowledgements

The authors are thankful to URSI-TÜRKİYE’2018 IX. Scientific Committee.

References

[1] E. Arıkan, Channel polarization: A method for constructing capacity achieving codes for symmetric binary- input memoryless channels, IEEE Trans. Inform. Theory, vol. 55, no. 7, pp. 3051-3073, July 2009.

[2] A. Balatsoukas, LLR-Based Successive Cancellation List Decoding of Polar Codes, IEEE Transactions on signal processing, vol. 63, no. 19, 2015.

[3] K. Chen, Improved Successive Cancellation Decoding of Polar Code, IEEE Transactions on Communications, vol. 61, No. 8, August 2013.

[4] O. Afisiadis, A. Balatsoukas-Stimming and A. Burg, A Low-Complexity Improved Successive Cancellation Decoder for Polar Codes, IEEE 48th Asilomar conference on signal, systems and computers, pp. 2116-2120, 2014.

[5] A. Hadi, E. Alsusa and K. M. Rabie, A Method to Enhance the Performance of Successive Cancellation Decoding in Polar Codes, IEEE 10th International Symposium on communication systems, Network and Digital Signal Processing (CSNDSP), pp. 1-5, 2016.

[6] L. Liping, X. Zuzheng, H. Yanjun, "Channel estimation with systematic polar codes," IEEE Transactions on vehicular technology, vol. 67, no. 6, pp. 4880-4889, 2018.

[7] E. Arikan, "Systematic Polar Coding", IEEE Communications Letters, vol. 15, no. 8, pp. 860-862, August 2011.

[8] N. Hussami, S. B. Korada, and R. Urbanke, "Performance of polar codes for channel and source coding," in Proc. 2009 IEEE Int. Symp. Inform. Theory, pp. 1488–1492.

[9] B. Yuan and K. K. Parhi, "Early stopping criteria for energy-efficient low-latency belief-propagation polar code decoders," IEEE Trans. Signal Process., 2014.

[10] B. Yuan and K. K. Parhi, "Algorithm and architecture for hybrid decoding of polar codes," Asilomar Conference on Signals, Systems and Computers, pp. 2050–2053, April, 2014.

[11] A. Elkelesh, S. Cammerer, M. Ebada, and S. Ten Brink, "Mitigating clipping effects on error floors under belief propagation decoding of polar codes," in Proceedings of the International Symposium on Wireless Communication Systems, pp. 384–389, August, 2017.

[12] A. Elkelesh, M. Ebada, S. Cammerer, and S. Ten Brink, "Belief propagation decoding of polar codes on permuted factor graphs," in IEEE Wireless Communications and Networking Conference, WCNC, 2018.

[13] S. Cammerer, T. Gruber, J. Hoydis and S. ten Brink, "Scaling Deep Learning-Based Decoding of Polar Codes via Partitioning," GLOBECOM 2017 - 2017 IEEE Global Communications Conference, Singapore, 2017, pp. 1-6.

[14] J. Xu, T. Che, and G. Choi, "XJ-BP: Express journey belief propagation decoding for polar codes," IEEE Global Communications Conference, GLOBECOM 2015, 2015.

[15] I. Tal and A. Vardy, "List decoding of polar codes," arXiv:1206.0050v1, May 2012.

[16] B. Li, H. Shen, and D. Tse, "An adaptive successive cancellation list decoder for polar codes with cyclic redundancy check," IEEE Commun. Lett., vol. 16, no. 12, pp. 2044–2047, 2012.

[17] K. Niu and K. Chen, "CRC-aided decoding of polar codes," IEEE Commun. Lett., vol. 16, no. 10, pp. 1668–1671, Oct. 2012.

[18] Q. Zhang, A. Liu, X. Pan, and K. Pan, "CRC code design for list decoding of polar codes," IEEE Commun. Lett., vol. 21, no. 6, pp. 1229–1232, Jun. 2017.

[19] P. Koopman and T. Chakravarty, "Cyclic redundancy code (CRC) polynomial selection for embedded networks," in Proc. IEEE Int. Conf. AINA, Jun. 2004, pp. 145–154.

[20] Q. Zhang, A. Liu, and X. Pan, "Efficient CRC concatenation scheme for polar codes," IET Electr. Lett., vol. 53, no. 13, pp. 860–862, Jun. 2017.

[21] C. Leroux, A. J. Raymond, G. Sarkis, and W. J. Gross, "A semi-parallel successive-cancellation decoder for polar codes," IEEE Trans. Signal Process., vol. 61, no. 2, pp. 289–299, Jan. 2013.

[22] C. Zhang and K. K. Parhi, "Low-latency sequential and overlapped architectures for successive cancellation polar decoder," IEEE Trans. Signal Process., vol. 61, no. 10, pp. 2429–2441, May 2013.

[23] A. Alamdar-Yazdi and F. R. Kschischang, "A simplified successive cancellation decoder for polar codes," IEEE Commun. Lett., vol. 15, no. 12, pp. 1378–1380, Dec. 2011.

[24] M. Bakshi, S. Jaggi, and M. Effros, "Concatenated Polar codes," in Proc. IEEE Int. Symp. Inf. Theory (ISIT), Jun. 2010, pp. 918–922.

[25] H. Mahdavifar, M. El-Khamy, J. Lee, and I. Kang, "Performance limits and practical decoding of interleaved Reed-Solomon polar concatenated codes," IEEE Trans. Commun., vol. 62, no. 5, pp. 1406–1417, May 2014.

[26] Y. Wang and K. R. Narayanan, "Concatenations of polar codes with outer BCH codes and convolutional codes," in Proc. 52th Annu. Allerton Conf., Monticello, IL, USA, Sep./Oct. 2014, pp. 813–819.

[27] A. Eslami and H. Pishro-Nik, "On finite-length performance of polar codes: Stopping sets, error floor, and concatenated design," IEEE Trans. Commun., vol. 61, no. 3, pp. 919–929, Mar. 2013.

[28] Y. X. Zhang and A. Liu, "Polar-LDPC concatenated coding for the AWGN wiretap channel," IEEE Commun. Lett., vol. 18, no. 10, pp. 1683–1686, Oct. 2014.