Academy journal for Basic and Applied Sciences (AJBAS) Volume 7, issue 1, 2025

Complexity Reduction with High Throughput M-Parallel Successive Cancellation Decoding

Abdelkareim Alrtaimi
Electrical and Electronic Engineering
The higher institute of science and technology
Alzawia-Libya
abd.alrtaimi@gmail.com
***ORCID iD** 0000-0002-1714-5818*

Abstract--- According to the strategy of SC decoding where its performance is limited by the bit by bit decoding, correcting the bit that wrongly determined in the future decoding procedure becomes impossible. In this paper we proposed a new technique to correct wrongly determined bits in direct through decoding operations; the error propagation can also be directly corrected. The enhanced version of successive cancellation decoding is proposed to improve its performance for any code-length. This technique called multi-parallel SCD can provide a significant complexity reduction by avoiding unnecessary path searching operations. Furthermore, only undetectable bits (lost bits) over successive cancellation decoder are taken into consideration for path searching operations. The proposed algorithm that is applied into each decoder arranged directly over M-parallel SCD through processing operation by taking both possibilities in decision of wrongly determined bits (zero or one) . Multi-parallel SCD technique is a self-correcting path searching track that illustrate a significant decoding performance with low complexity over Binary Erasure Channel (BEC).

***Index Terms*— Multi-Parallel SCD, polar coding, BEC.**

I. INTRODUCTION

Polar codes invented by Arikan [1] is the primary class of error correcting code that can provably achieve the capacity for any binary discrete memoryless channel (B-DMC) when the code length approach to infinity. The channel polarization on independent copies of a given B-DMC can achieve the capacity by performing channel splitting and channel combining operations. The channel reliabilities for constructing polar codes can be calculated efficiently using Bhattacharyya parameters over binary-input erasure channels (BECs) [1]. By transmitting free bits (called information bits) over these noiseless channels and by transmitting fixed bits (called frozen bits) over the others, polar codes can achieve the symmetric capacity under a successive cancellation (SC) decoder with both encoding and decoding complexities O(NlogN). Because subchannels are not completely polarized for finite length of polar codes, the error-correcting performance of successive cancellation (SC) decoding algorithm is poor at short and moderate block lengths. Much attention was especially given to efforts toward improving the throughput. The successive cancellation list (SCL) introduced in [2] is developed and shows significant performance improvement compared to SC decoding.

In an SCL decoder, both 0 and 1 are considered as estimated bits and two decoding paths are generated at each decoding stage. The cyclic redundancy check (CRC) is used in [3, 4] to select the correct decoding path in the SCL algorithm. However, SCL decoding has much higher decoding complexity [5, 6]. Although much work has been done in the area of polar decoding in recent years [7–11], it is still an open issue to find a decoding algorithm with both good frame error rate (FER) performance and low complexity, especially with finite-length polar codes.

This proposed algorithm is a generic SC decoding scheme with enhanced decision functions that applied in multi-SC decoders constructed in parallel to correct the lost bits that in turn leads to increase the error propagation. This technique can provide a flexible configuration. Further, self-correcting path searching track leads to Pruning unnecessary path searching operations, which reduce the decoding complexity. Multi-Parallel SC decoding show a significant performance improvement compared with the original SC decoding.

II. PRELIMINARIES AND NOTATIONS

Binary discrete memoryless channels (B-DMC) are an important class of channels studied in information theory and an important example of this kind of channels is the BEC, which is considered for illustrative purposes in this paper. The main idea of polar codes is to construct from N independent copies of a (B-DMC) W, a new set of N channels $W_N^{(i)}$ with $1 \le i \le N$ using a linear transformation. The more N increases, the more these new channels $W_N^{(i)}$ are polarized. In this paper we write $W: X \to Y$ to denote a generic binary discrete memoryless channel (B-DMC) with input alphabet X, output alphabet Y and transition probabilities $W(y|x)$, $x \in X$, $y \in Y$. Considering a BEC, the input alphabet X will always be a binary input {0, 1} while Y and the transition probabilities may be arbitrary. We write W_N to denote the channel corresponding to N independent uses of W; therefore, $W^N: X^N \to Y^N$ with $W^N(y_1^N|x_1^N) = \prod_{i=1}^{N} W(y_i|x_i)$. Let $y_1^N = (y_1, y_2, \dots, y_N)$ be the observations of

the outputs bits $x_1^N = (x_1, x_2, \dots, x_N)$ through N copies of the channel W where the input bits are $u_1^N = (u_1, u_2, \dots, u_N)$.
The mutual information of a B-DMC with input alphabet $X = \{0,1\}$ is defined as.

$$I(W) \triangleq I(X;Y) = \sum_{y\in Y}\sum_{x\in X}\frac{1}{2}W(y|x)log\frac{W(y|x)}{\frac{1}{2}W(y|0)+\frac{1}{2}W(y|1)} \quad (1)$$

Where X and Y are two discrete random variables corresponding to input and output, respectively, and W(y|x) is the channel transition probability for $x \in X$ and $y \in Y$.
The general mapping $u_1^N \to x_1^N$ can be written by induction and it represented by G_N so that $x_1^N = u_1^N G_N$ and the transition probabilities of the two channels W_N and W^N are related by $W_N(y_1^N|u_1^N) = W^N(y_1^N|u_1^N G_N)$ for all $y_1^N \in Y^N, u_1^N \in X^N$ where $W^N(y_1^N|u_1^N G_N)$ is the vector channel which contains the transformation.

A. Polar Encoding

For (N, K) polar code of K information bits and N encoded bits$(N = 2^n)$, an invertible matrix G_N is introduced to describe channel polarization. Here, $G_N = B_N F^{\otimes n}$ for $N = 2^n$ and $n \geq 1$, where B_N is the bit-reversal matrix, $\otimes n$ denotes the nth Kronecker product and F is defined as.

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

And B_N can be found by using $B_N = R_N\left(I_2 \otimes B_{\frac{N}{2}}\right)$
Where I_2 is the 2-D identity matrix, B_2 is initialized as $B_2 = I_2$, R_N is the permutation operation which maps the input sequence $\{1,2,3,4, \dots\dots, N\}$ to $\{1,3, \dots, N-1, 2,4, \dots, N\}$ and $n = \log_2 N$. The generator matrix in such structure of the encoder shown in figure (1) for N=4 will be

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

The main idea of the transformation in the polar encoder is to create a set of channels with capacity $C \to 1$ for N goes to infinity. The information bits send to the channels that are almost free of noise and the remaining $(N-K)$ is frozen bits that are transmitted in the noisy channels. This process can be accomplished using Bhattacharyya parameter which is always takes the values between 0 and 1 and it denoted by Z(W). Whereas the channels with Z(W) close to zero are almost noiseless, while channels with Z(W) close to one are almost noisy channels. As shown in figure.1, the Bhattacharyya parameters of individual bit-channels in the polar transformation can be calculated by the following recursive formulas, when W is a BEC with erasure probability ϵ, where $Z\left(W_1^{(1)}\right) = \epsilon$.

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

$$Z\left(W_N^{(2i)}\right) = Z\left(W_{\frac{N}{2}}^{(i)}\right)^2 \quad (3)$$

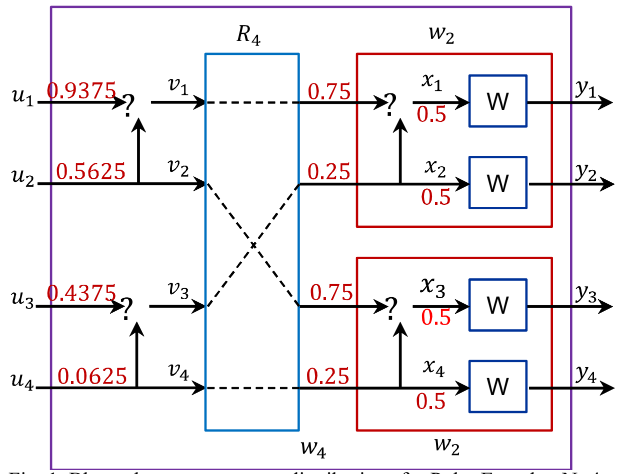

Fig. 1. Bhattacharyya parameter distributions for Polar Encoder, N=4.

Due to the recursive structure of polar encoding and decoding, the complexity of polar codes is the most important issue. The complexity of the SC decoder is the same as encoder namely $O(N \log N)$. Let us first consider the encoding complexity. The encoding is done layer by layer for $n = \log N$ within the recursive channel combining operation. Let $E\chi(N)$ denote the encoding complexity for blocklength N.

$$E\chi(N) = \frac{N}{2} + 2E\chi(N/2) \quad (4)$$

For instance when blocklength $(N = 2)$, the encoding complexity is $E\chi(2) = 1$ because of just one XOR operation is needed to compute $u_0^1\, G_2$. The above recursive relation implies

$$\begin{aligned} E\chi(N) &= \frac{N}{2} + 2E\chi(N/2) = \frac{N}{2} + 2(N/4 + 2E\chi(N/4)) \\ &= \frac{N}{2} + \frac{N}{2} + 4\left(N/8 + 2E\chi(N/8)\right) = \frac{N}{2}logN \end{aligned} \quad (5)$$

B. Successive Cancellation Decoding

At the receiver side, the original transmitted codeword x_i is corrupted because of noise interference to the received codeword $y_1^N = (y_1, y_2, \dots, y_N)$. After receiving y_1^N the bits $\hat{u}_i$ are determined successively with index i from 1 to N by using the likelihood ratio (LR) of y_i. In this paper Binary Erasure Channel model is depicted, where information may be lost but is never corrupted. A transmitted bit is either received correctly with probability $(1-\epsilon)$ or known to be lost with probability ϵ. The likelihood ratio (LR) of y_i according to original transmitted codeword $x_i = \{0 \text{ or } 1 \text{ or } -1\}$ will be $y_i = \{\text{high or low or } 1\}$ respectively. As shown in fig.2, the LR-values from the f-function in (6) depend only on LR-values of nodes a , b and LR values calculated with the g-function in (7) are in addition to that also influenced by the result from the f-function.

DOI: 10.5281/zenodo.15507494

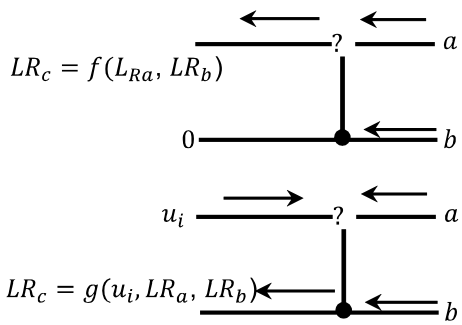

Fig..2. Likelihood ratio (LR) calculation over Kernel decoder unit

$$f(LR_a, LR_b) = \frac{LR_a LR_b + 1}{L_{Ra} + LR_b} \quad (6)$$

$$g(u_i, LR_a, LR_b) = LR_b . LR_a^{(1-2u_i)} \quad (7)$$

Where LR_a, LR_b are the likelihood ratios of received codeword y_1^N.
The formulas in (6) and (7) are expressed in a recursive manner in [1] as

$$L_N^{(2i-1)}\left(y_1^N, \hat{u}_1^{2i-2}\right) = \frac{L_{\frac{N}{2}}^{(i)}\left(y_1^{\frac{N}{2}}, \hat{u}_{1,o}^{2i-2} \oplus \hat{u}_{1,e}^{2i-2}\right) L_{\frac{N}{2}}^{(i)}\left(y_{\frac{N}{2}+1}^{N}, \hat{u}_{1,e}^{2i-2}\right) + 1}{L_{\frac{N}{2}}^{(i)}\left(y_1^{\frac{N}{2}}, \hat{u}_{1,o}^{2i-2} \oplus \hat{u}_{1,e}^{2i-2}\right) + L_{\frac{N}{2}}^{(i)}\left(y_{\frac{N}{2}+1}^{N}, \hat{u}_{1,e}^{2i-2}\right)} \quad (8)$$

And

$$L_N^{(2i)}\left(y_1^N, \hat{u}_1^{2i-1}\right) = \left[L_{\frac{N}{2}}^{(i)}\left(y_1^{\frac{N}{2}}, \hat{u}_{1,o}^{2i-2} \oplus \hat{u}_{1,e}^{2i-2}\right)\right]^{1-\hat{u}_{2i-1}} . L_{\frac{N}{2}}^{(i)}\left(y_{\frac{N}{2}+1}^{N}, \hat{u}_{1,e}^{2i-2}\right) \quad (9)$$

The decision made using

$$\hat{u}_i = \begin{cases} 0, if \ \ LR(u_i) \geq 1 \\ 1, otherwise \end{cases} \quad (10)$$

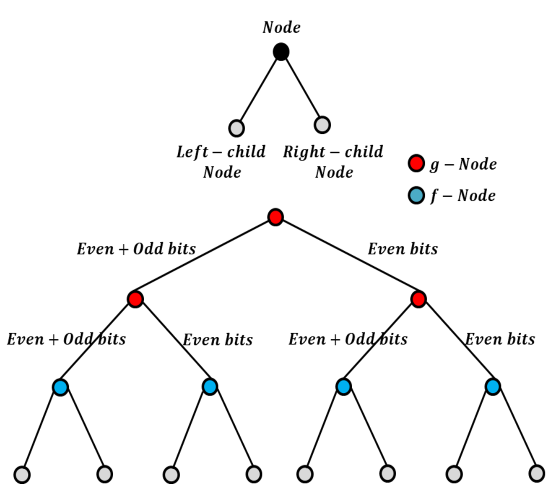

Fig. 3. Bits distribution over SC decoding procedure for N=8.

It can be seen that in Figure (3) the tree structure of SC decoder consists of two types of computation units with bit distribution over SC decoding procedure for N=8 polar codes. Here the decoder consists of two basic nodes, namely f node and g node, note that g node can be (0 or 1) according to the previous decoded bit, the operations of these two nodes are distributed only for estimate the first data bit where the g-node in level (1 and 2) are activated.

III. Proposed M-Parallel SC Decoding Algorithms

According to the strategy of SC decoding where its performance is limited by the bit by bit decoding, correcting the bit that wrongly determined in the future decoding procedure becomes impossible. In this paper, we proposed a technique to correct the bits that are wrongly determined in direct through decoding operation. The enhanced version of successive cancellation decoding is proposed to improve its performance for any code-length. The block diagram shown in figures (4.a and 5.a) called multi-parallel SCD where M=(2 and 4) respectively can provide a significant complexity reduction by avoiding unnecessary path searching operations.

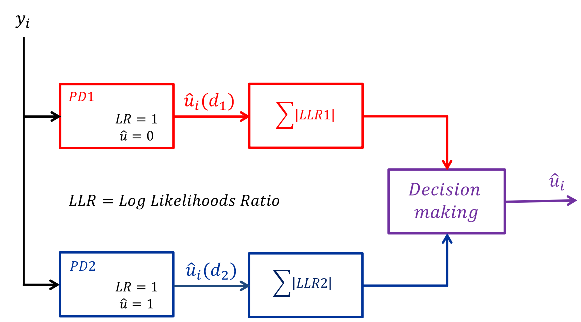

Fig.4.a. Block diagram for M=2 parallel SC decoding operation.

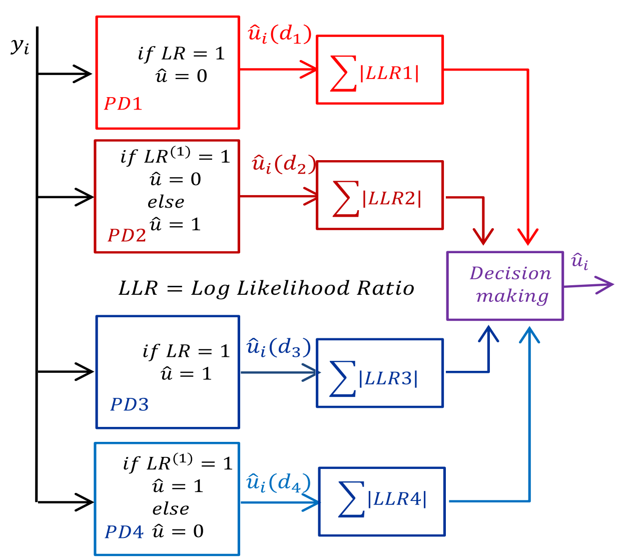

Fig.5.a. Block diagram for M=4 parallel SC decoding operation.

The algorithm that is applied into each SCD arranged directly over M-parallel SCD through processing operation, by taking all possibilities decision of undetected bits (zero or one). For M=2, the decision operation for the decoder (1,2) are depicted as in (11.a,11.b) respectively, and this procedure are applied for all bits that are wrongly determined which have LR = 1.

$$\hat{u}_i(D1) = \begin{cases} 0, if \ \ LR(u_i) \geq 1 \\ 1, otherwise \end{cases} \quad (11.a)$$

$$\hat{u}_i(D2) = \begin{cases} 1, if \ \ LR(u_i) \leq 1 \\ 0, otherwise \end{cases} \quad (11.b)$$

Unlike to the SC decoder where only one path is reserved after processing at each level, the m-parallel algorithm utilizes M different searching paths as shown in figures (4.b and 5.b). Therefore, it is more likely for the m-parallel algorithm to find the desired path than the SC algorithm. Instead of waiting to find out all the M candidate paths at every level, we can thus keep on searching along the single candidate path in each decoder. For M=4, the decision operation for the decoder (1,3) are depicted as in (11.a,11.b) respectively, therefore for M=4, two more decoders are added with specific decision operation in each decoder as in (11.c,11.d) and this procedure are also applied for all bits that are wrongly determined which have LR = 1.

$$\hat{u}_i(D3) = \begin{cases} 0, if \ \ LR^{1st}(u_i) = 1 \\ 1, otherwise \end{cases} . \quad (11.c)$$

$$\hat{u}_i(D4) = \begin{cases} 1, if \ \ LR^{1st}(u_i) = 1 \\ 0, otherwise \end{cases} . \quad (11.d)$$

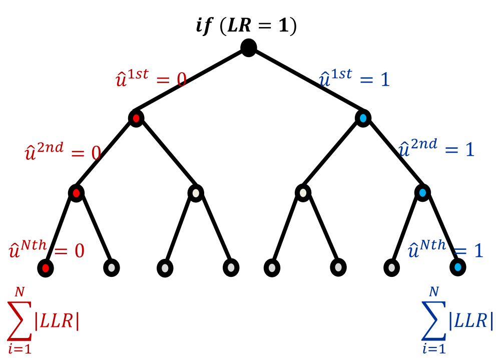

Fig.4.b. Tree structure for M=2 parallel SC decoding operation.

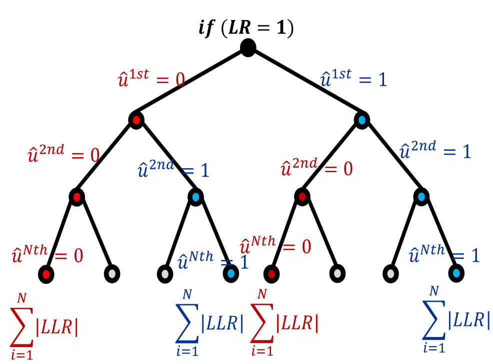

Fig.5.b. Tree structure for M=4 parallel SC decoding operation.

A multi-parallel decoder keeps track of several decoding results instead of just one, in fact for M = 1we obtain the SC again. Instead of deciding to set the value of u_i, it takes both options. Since for each information bit it splits the decoding path into two new paths (one ends with "0" and the other ends in "1"), in figure (4.b and 5.b) we prune unnecessary path searching and the maximum number of paths allowed is M. In order to keep the best paths at each stage, the pruning criterion will be to keep the most likely paths. Let us try now to make an example just for the first tree bits and M = 2 into two cases, in case(a), where the 1^{st}, 2^{nd} and 3^{rd} bits are lost as shown in figure (6.a) and case (b), where the $1^{st}$and 3^{rd} bits are lost but the 2^{nd} bit is normally detected as 1, as shown in figure (6.b).

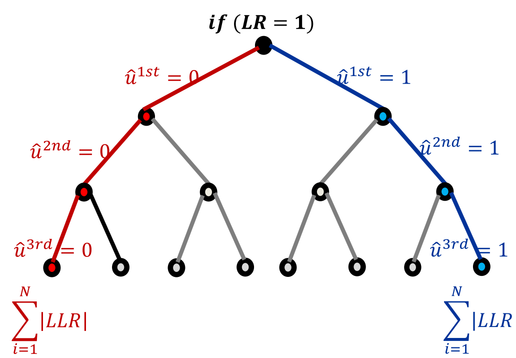

Fig.6.a. Decoding procedure in case (a), where the 1^{st}, 2^{nd} and 3^{rd} bits are lost

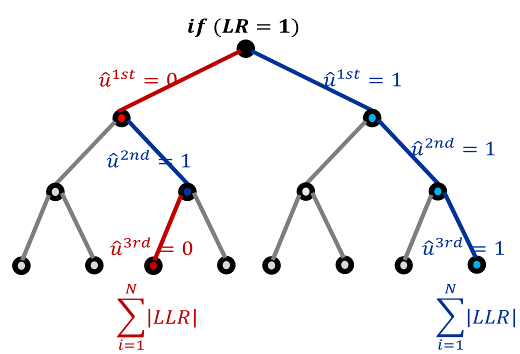

Fig.6.b. Decoding procedure in case (b), where the $1^{st}$and 3^{rd} bits are lost but the 2^{nd} bit is normally detected as 1.

We assume N = 8. First, the decoding algorithm starts and the first bit can be either 0 or 1. In the case (a) the 1^{st}, 2^{nd} and 3^{rd} bits assumes to be either zeros or ones thus the possible words are {000,111} but the number of paths is not greater than M = 2 , in this case just the first bit can be exactly correct and this is the worst case. However in case (b), M=2 where the $1^{st}$and 3^{rd} bits are lost but the 2^{nd} bit is normally detected as 1. The possible words will be {010,111}, hence the first and second bits are true, but the third bit can be true with probability =0.5. As seen in this case the path searching is changed with self-correcting track. Finally the most likely path which has the highest logarithmic likelihood decoding will be chosen.

DOI: 10.5281/zenodo.15507494

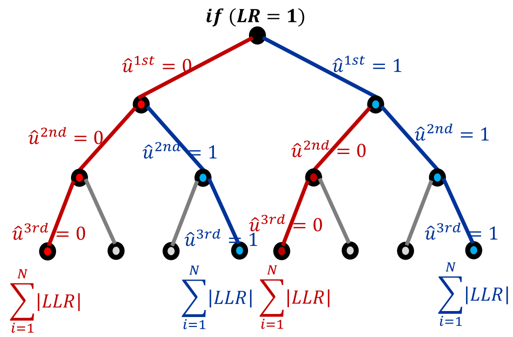

Fig.7.a. Decoding procedure in case (a), where the 1^{st}, 2^{nd} and 3^{rd} bits are lost.

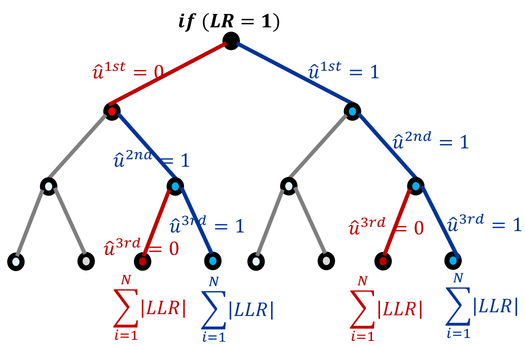

Fig.7.b Decoding procedure in case (b), where the $1^{st}$and 3^{rd} bits are lost but the 2^{nd} bit is normally detected as 1.

.

Now let us increase M=4 as shown in figures (7.a and 7.b) to the same previous example cases. As clearly seen in case (a) the possible words are {000,011,100,111} and the number of paths is not greater than M = 4. In case (b), M=4 where the $1^{st}$and 3^{rd} bits are lost but the 2^{nd} bit is normally detected as 1. The possible words will be {010,011,110,111}. As seen in this case the path searching is changed with self-correcting track. The most likely path that has the highest logarithmic likelihood decoding will be chosen.

III. Simulation Results

As for the implementation aspect, simulation results for binary erasure channels with erasure probability 0.5 in terms of bit error rate BER and frame error rate FER for short and moderate frame lengths are used. The cyclic redundancy chick CRC is also applied, the code rate of all simulations in this paper include the CRC bits. As clearly seen that the decoding performance of improved version of SCD namely M-parallel SCD compared with the original SC decoding algorithm, have a significant improvement as much as M increased.

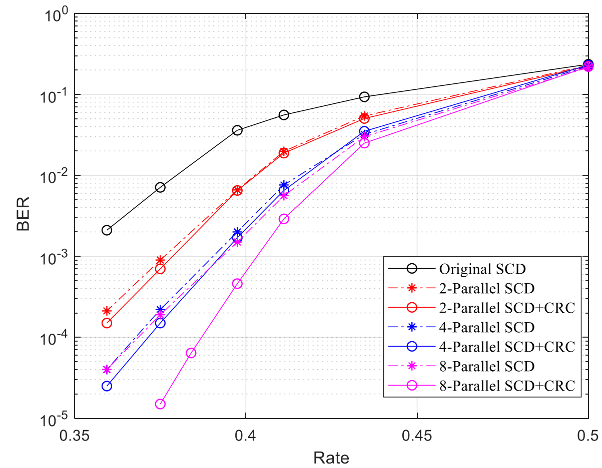

Fig. 8. BER performance under different M size for frame- length P(1024) over Binary Erasure Channel.

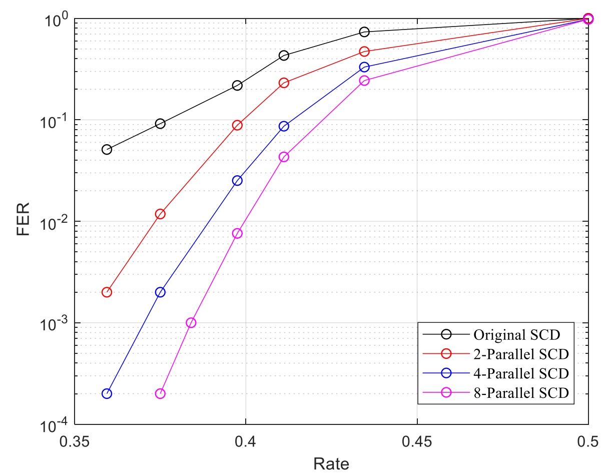

Fig. 9. FER performance under different M size for frame- length P(1024) over Binary Erasure Channel.

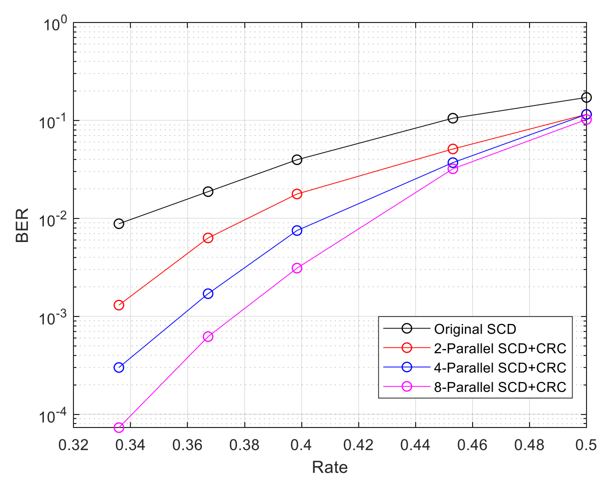

Fig. 10. BER performance under different M size for frame- length P(128) over Binary Erasure Channel.

DOI: 10.5281/zenodo.15507494

IV. Conclusion

The SC decoding algorithm of polar codes and its improved versions, namely, M-parallel SC decoding is restated as path search procedures on the code tree of polar codes. This technique can avoid unnecessary path searching and has an ability of self-correcting path searching track when the wrongly determined bits are not consecutively ordered. The number of searching paths can be greatly reduced based on the proposed scheme. So, the time and space complexities of M-parallel SC decoding are $O(N\,logN)$ and less than or equal to $O(M.N)$, compared with the original SC complexity of $O(N\,logN)$ and $O(N)$ respectively.

References

[1] E. Arıkan, "Channel Polarization: A Method for Constructing Capacity-Achieving Codes for Symmetric Binary-Input Memoryless Channels," *IEEE Trans. Inf. Theory*, vol. 55, no. 7, pp. 3051–3073, Jul. 2009.

[2] I. Tal and A. Vardy, "List Decoding of Polar Codes," *IEEE Trans. Inf. Theory*, vol. 61, no. 5, pp. 2213–2226, May 2015.

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

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

[5] G. Sarkis, P. Giard, A. Vardy, C. Thibeault, and W. Gross, "Fast polar decoders: Algorithm and implementation," *IEEE J. Sel. Areas Commun.*,vol. 32, no. 5, pp. 946–957, May 2014.

[6] G. Sarkis, P. Giard, A. Vardy, C. Thibeault, and W. J. Gross, "Fast list Decoders for polar codes*," IEEE J. Sel. Areas Commun*., vol. 34, no. 2, pp. 318–328, Feb 2016.

[7] K. Chen, K. Niu and J. R. Lin, "Improved successive cancellation decoding of polar codes," *IEEE Trans. Commun*., vol. 61, no. 8, pp. 3100-3107, Aug. 2013.

[8] Daesung Kim, In-Cheol Park, "A Fast Successive Cancellation List Decoder for Polar Codes With an Early Stopping Criterion," *Signal Processing IEEE Transactions on*, vol. 66, no. 18, pp. 4971-4979, 2018.

[9] S. A. Hashemi, C. Condo, and W. J, "Simplified Successive Cancellation List Decoding of Polar Codes," *IEEE International Symposium on Information Theory*, pp.815-819,Aug. 2016.

[10] H. Sun, J. Gao, L. Li, Z. Ma, P. Fan. "Successive Cancellation List Flipping for Short Polar Codes Based on Row Weights of Generator Matrix", IEEE Wireless Communications and Networking Conference (WCNC) 2021, pp. 1-6.

[11] Ercan, Furkan, and Warren J. Gross. "Fast thresholded SC-flip decoding of polar codes." IEEE International Conference on Communications (ICC), 2020, pp. 1-7.