# An Improved Multiple Bit-Flipping Successive Cancellation Decoding Algorithm for Polar Codes

Sha Sha, Li Zhang, and Yejun He*
Guangdong Engineering Research Center of Base Station Antennas and Propagation
Shenzhen Key Laboratory of Antennas and Propagation
College of Electronics and Information Engineering, Shenzhen University, 518060, China
Email: 1791784951@qq.com, wzhangli@szu.edu.cn, heyejun@126.com*

***Abstract*—Although original successive cancellation flip (SCF) decoding of the polar codes can identify the first error position and flip the error, decoding still fails when other errors occur after the corrected error bit. Therefore, the improvement of the block error rate (BLER) performance is limited. In this paper, we present an improved algorithm that can correct multiple error bits, named as improved multiple bit-flipping successive cancellation decoding algorithm. First, multiple sets are constructed by calculating the log likelihood ratio (LLR) of each information bit. Then, a bit is selected and corrected from the first set when original decoding fails; two bits are selected and flipped from the second set when all bits in the first set cannot be successfully decoded; when all bits combinations in the second set are flipped but the output vector still does not match the input vector, three bits are selected and corrected from the third set. Simulation results show that the BLER performance of the improved algorithm is better than that of the successive cancellation list decoding (SCL) decoding algorithm with a list size of $L = 8$ for $N = 256$. When $N = 1024$ and *SNR* $= 3$ dB, the BLER performance of improved algorithm is slightly better than that of SCL decoding algorithm with a list size of $L = 4$.**

***Index Terms*—Polar codes, successive cancellation decoding, multiple bit-flipping decoding**

## I. Introduction

Arikan proposed polar codes that provably achieve the channel capacity with infinite code length [1], where polar codes are the major breakthrough in coding theory. In addition, successive cancellation (SC) decoding proposed in [1] [2], which is the first decoding algorithm, has low computational complexity. However, decoding performance is affected when errors occur during the decoding process. Due to the linear characteristics of SC decoding, previous error bits decrease the accuracy of subsequent decoding. In severe cases, it may even cause error propagation which seriously affects decoding performance. Accordingly, people proposed two methods to improve this problem. Method 1: [3] proposed successive cancellation list decoding (SCL) algorithm to solve the error caused by only one decoding path in SC decoding algorithm. Path metric (PM) is introduced in [3]. Multiple paths are reserved to improve the survival rate of correct paths during decoding. [4] introduced the cyclic redundancy check (CRC) to further improve the BLER performance, which has become a common practice in the study of polar codes. Method 2: [5] proposed SCF decoding, which constructs a set containing error-prone bits based on LLR and selects the bits in a set to correct the errors. However, this method can only correct single-errors, which limits the improvement of BLER performance. In order to effectively improve the BLER performance, critical set (CS) is constructed in [6]. After SC decoding fails, multiple information bits can be selected from the CS to be flipped, but this method also introduces higher complexity. [7] considered the word error rate (WER) performance of the ideal Generalized SCFlip decoding. The proposed decoding algorithm has maximum bit-flip order $\omega$, revealing that the use of higher order bit flip can achieve significant improvements. Another method called the partition decoding algorithm is mentioned in [8]. This method divides the decoding tree of polar codes into several regions. Then, the algorithm is implemented separately for each area. It can significantly reduce the average number of iterations. In [9], special nodes are introduced into the flip algorithm to achieve fast simplified successive cancellation (Fast-SSC) decoding, which reduces decoding latency. Thresholded SC-Flip (TSCF) decoding algorithm discussed the relationship between the average LLRs value and the distribution of the errors bit-channels [10].

In this paper, our main contributions are as follows.

- Multiple flipped bit sets are established by counting the minimum and average value of LLRs before decoding.
- In order to locate the position of the error as quickly as possible, the LLRs of the bits in the set are sorted in ascending order during decoding, so that the bit with the minimum LLR is first flipped.
- Using three sets for different error conditions can correct multiple errors to improve decoding performance.

The remainder of this paper is organized as follows. In Section II, the principle of polar codes, encoding and basic decoding algorithm are introduced. In Section III, the improved SCF algorithm is described in detail. In Section IV, the simulation results are given. In Section V, conclusions are drawn.

## II. Related Work

### A. Polar Codes

Polar coding is a novel channel coding method constructed by channel polarization. Channel polarization includes two operations: channel combining and channel splitting. We denote by $W$ the Binary-Discrete Memoryless Channel (B-DMC). In the channel combining operation: the known $N$ B-DMC channels are combined to generate a new channel $W_N$, stipulating that $N = 2^n$ ($n \geq 0$), when $n = 0$, $W_1 = W$; if $n = 1$, then two W channels are combined into $W_2$; when $n = 3$, the two $W_2$ channels are combined into $W_4$, etc. Finally, the $N$ channels are combined into the channel $W_N$ through iterative operations. Channel splitting is to split the combined channel $W_N$ into $N$ channels and the $N$ B-DMC channels after splitting are different from the previous $N$ channels. This discrepancy is called the channel polarization. The channel polarization appears in $N$ channels: some channels have higher channel reliability and the channel capacity is close to 1, while other channels have lower reliability and the channel capacity is close to 0. In communication transmission, We can use the poor channels to transmit the known information of the receiving end and the sending end, and the excellent channels are used to transmit important information, thereby improving the reliability of the transmission.

The encoding of polar codes includes two important steps: constructing the generator matrix and selecting the information bits. Let $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$ be the encoded vector, it is defined by

$$\boldsymbol{x} = \boldsymbol{u}\boldsymbol{G_N}. \tag{1}$$

where $N = 2^n$ is the code length, $\boldsymbol{G_N}$ is the $N \times N$ generator matrix, and $\boldsymbol{u}$ is the $1 \times N$ vector containing $K$ information bits and $N$-$K$ frozen bits, where the information bits are set to 1 and the frozen bits are set to 0.

### B. SC Decoding of Polar Codes

The decoding of the polar codes uses the SC decoding algorithm [1] [2]. Let $\hat{u}$ be the estimated bit of the $u_i$, it is defined as

$$\hat{u} = \begin{cases} h_i(y_1^N, \hat{u}_1^{i-1}), & if\ i \in A; \\ u_i, & if\ i \in A_c. \end{cases} \tag{2}$$

where $y_1^N$ denotes the output of the decoding, $A$ is called information bits sets, and $A_c$ is called frozen bits sets. If the currently decoded bit is a frozen bit, the result can be directly decided as $u_i$. When the decoded bit is an information bit, it is necessary to further judge according to the decision function. The decision function $h_i$ is defined as

$$h_i(y_1^N, \hat{u}_1^{i-1}) = \begin{cases} 0, & if\ L_N^{(i)}(y_1^N, \hat{u}_1^{i-1}) \geq 0; \\ 1, & if\ L_N^{(i)}(y_1^N, \hat{u}_1^{i-1}) < 0. \end{cases} \tag{3}$$

where $L_N^{(i)}(y_1^N, \hat{u}_1^{i-1})$ is LLR. When $L_N^{(i)}(y_1^N, \hat{u}_1^{i-1})$ is less than 0, the decision is 1, otherwise the decision is 0. The decision LLR can be obtained by

$$\begin{aligned} &L_N^{(2i-1)}(y_1^N, \hat{u}_1^{2i-2}) = \\ &f(L_{N/2}^{(i)}(y_1^{N/2}, \hat{u}_{1,o}^{2i-2} \oplus \hat{u}_{1,e}^{2i-2}), L_{N/2}^{(i)}(y_{N/2+1}^N, \hat{u}_{1,e}^{2i-2})); \end{aligned} \tag{4}$$

$$\begin{aligned} &L_N^{(2i)}(y_1^N, \hat{u}_1^{2i-1}) = \\ &g(L_{N/2}^{(i)}(y_1^{N/2}, \hat{u}_{1,o}^{2i-2} \oplus \hat{u}_{1,e}^{2i-2}), L_{N/2}^{(i)}(y_{N/2+1}^N, \hat{u}_{1,e}^{2i-2}), \hat{u}_{2i-1}). \end{aligned} \tag{5}$$

where $f$, $g$ are defined as

$$f(a,b) = ln\left(\frac{1+e^{a+b}}{e^a+e^b}\right); \tag{6}$$

$$g(a,b,u_s) = (-1)^{u_s}a + b. \tag{7}$$

where $a$ and $b$ denote LLRs of input bits. $u_s$ ($u_s \in 0,1$) denotes the extra input of $g$, called partial sum. Since there are exponential and logarithmic operations in (6), it is usually approximated as

$$f(a,b) = sign(a)sign(b)min\{|a|,|b|\}. \tag{8}$$

## III. Improved Multiple Bit-Flipping Decoding Algorithm

In this section, we describe two ways for constructing flip sets and propose a new method for building flip sets. In this way, flip sets are established for PC (256, 128) and PC (1024, 512), where PC ($N$, $K$ + $r$) represents a polar code of length $N$ and $K$ information bits. The first flip set is established by calculating the LLR of each information bit, and then the other two sets are constructed on the basis of the first set. Finally, we can use the created set to try to flip multiple errors.

### A. Creating Filp Sets

The purpose of establishing a set is to collect error-prone bit as efficiently as possible. In [5], what we need to correct is the bit with the smallest LLR during decoding. The smaller the LLR, the worse the reliability of the channel.

Reference [5] established a flip set and introduced a threshold. When the CRC detects decoding errors, it will start to flip from the bit with the minimum LLR until the decoding is successful or reaches the flip threshold. This method of creating a flip set requires sorting the LLRs of all bits at each decoding. Reference [6] proposed that a flip set can be established before decoding.

What both methods have in common is to find information bits with smaller LLRs. They believe that channels with smaller LLRs are unreliable, and the transmission of information in these channels is error-prone, so we need to pay attention to these channels.

Following this line of thought, we first use the original SC decoding algorithm for decoding. The LLR of each bit is recorded frame by frame (only the LLR of the information bit is recorded) during decoding. The average LLR of each information bit is calculated to obtain Fig. 1. It can be seen from the Fig. 1 that the LLRs of some bits are significantly lower than others. After turning the LLRs of all bits of the error frames into a positive number, the minimum value of each bit is taken to obtain Fig. 2. By comparing Fig. 1 and Fig. 2, we can find that the average LLRs and the minimum LLRs are similar. The bit with the smaller LLRs in Fig. 2

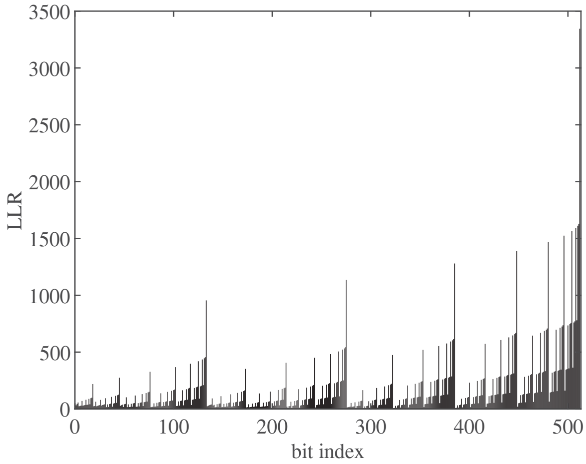


Fig. 1. The average LLR of each information bit for PC (1024, 512).

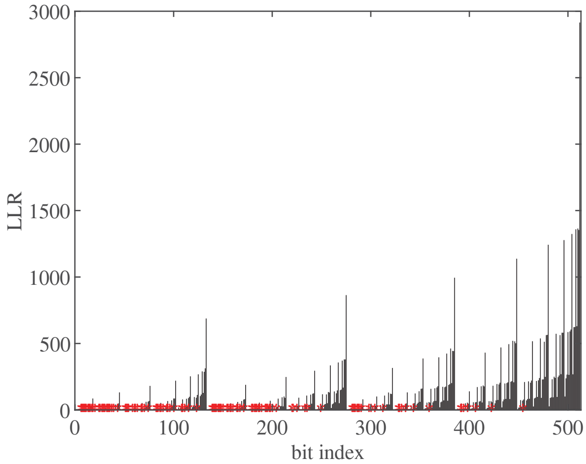


Fig. 2. The minimum LLR of each information bit for PC (1024, 512).

is considered as an error-prone bit. In the statistical process, comparing the LLRs of all information bits for PC (1024, 512), the smallest LLR = 0 can be found. Therefore, in order to highlight these special bits, the information bits with LLR = 0 are marked as * in Fig. 2. Finally, the information bits with * are collected to obtain a set of 108 error-prone bits. It should be noted that the bit as CRC has been deleted. For PC (256, 128), we use the same method to select information bits with LLR = 0 to obtain a set of 40 error-prone bits. Let $\boldsymbol{S_1}$ be the collection created by this method.

### *B. Improved Decoding Algorithm*

In the original SCF decoding method, one bit is selected and flipped from the set when the CRC detects an error after SC decoding.

In this paper, three flip sets are constructed for the algorithm. The first flip set is the aforementioned 108-bits set and 40-bits set, called $\boldsymbol{S_1}$. The remaining two flip sets are subsets of the first flip set, called $\boldsymbol{S_2}$ and $\boldsymbol{S_3}$. In actual decoding, when $\boldsymbol{S_2}$ and $\boldsymbol{S_3}$ contain too many bits, the decoding performance is not significantly improved and the decoding complexity has increased.

In addition, in order to reduce the decoding complexity, it is necessary to further optimize the method of constructing the flip set $\boldsymbol{S_1}$. When decoding a frame of the information vector, we reorder the error-prone bits to obtain a new set $\boldsymbol{S_1}$. Since bits with smaller LLRs are more error-prone, these bits are placed first. Then sets $\boldsymbol{S_2}$ and $\boldsymbol{S_3}$ are updated in the same way to obtain new sets $\boldsymbol{S_2'}$ and $\boldsymbol{S_3'}$.

The improved decoding algorithm can locate 1 to 3 error positions. As shown in Algorithm 1, one bit is selected and flipped from the first set $\boldsymbol{S_1'}$ when SC decoding fails. If the number of inversions reaches the size of the set $\boldsymbol{S_1'}$, it means that there may be other error bit after corrected bits during decoding, i.e., decoding errors may need to be reversed twice to correct them. Therefore, two possible bits can be selected from $\boldsymbol{S_2'}$ (regardless of order) as the target of error correction. Define $s$ as $\boldsymbol{S_2'}$ set size, then the maximum number of flips is $C_s^2$.

**Algorithm 1** Multiple Bit-Flipping Decoding of Polar Codes

**Input:** $y_1^N$, $N$, $L$, $S_1'$, $S_2'$, $S_3'$, $n$um
**Output:** $x_1^N$
**function** POLARDECODER$(y, N, L)$
  $Data \leftarrow SC(y_1^N)$
  **if** $CRC(Data) = success$ **then**
    $x_1^N \leftarrow Data$
  **else**
    $Data \leftarrow BitFlip1(y_1^N, S_1', num)$
    **if** $CRC(Data) = success$ **then**
      $x_1^N \leftarrow Data$
    **else**
      $Data \leftarrow BitFlip2(y_1^N, S_2', num)$
      **if** $CRC(Data) = success$ **then**
        $x_1^N \leftarrow Data$
      **else**
        $Data \leftarrow BitFlip3(y_1^N, S_3', num)$
        **if** $CRC(Data) = success$ **then**
          $x_1^N \leftarrow Data$
        **else**
          $x_1^N \leftarrow MaxLlr(y_1^N)$
        **end if**
      **end if**
    **end if**
  **end if**
  **return** $x_1^N$
**end function**

If there are still errors in the CRC detection after trying all combinations of $\boldsymbol{S_2'}$, this means that there are errors after the two corrected bits. At this time, three bits can be selected and flipped (regardless of order) from the $\boldsymbol{S_3'}$ set, and the number of flips can be up to $C_s^3$. If the decoding is still unsuccessful, the decoding declares failure.

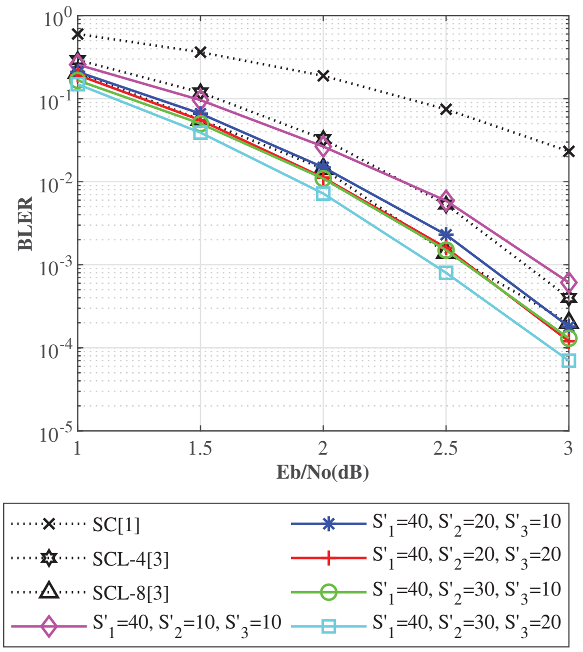


Fig. 3. Improved decoding BLER performance compared with SC and SCL for PC (256, 128) polar codes, and CRC length is 24.

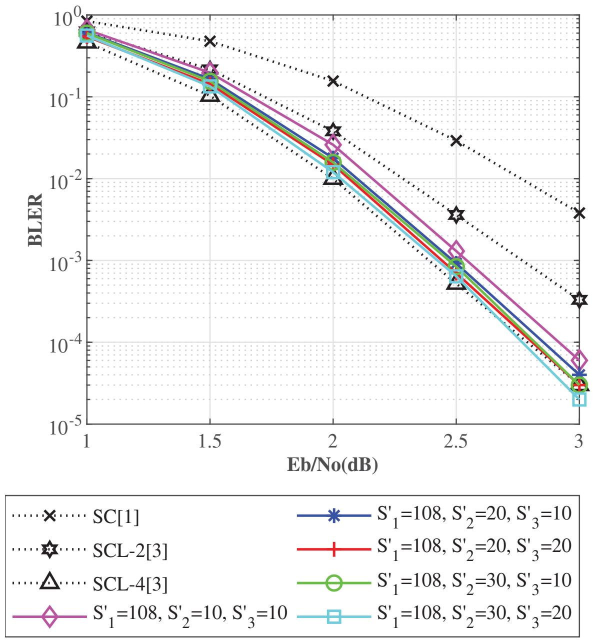


Fig. 4. Improved decoding BLER performance compared with SC and SCL for PC (1024, 512) polar codes, and CRC length is 24.

## IV. SIMULATION

In this section, we compare the improved bit-flipping decoding algorithm with the algorithm in [1] [3]. We choose PC (256, 128) and PC (1024, 512) in experiments where the number of valid information bits is 96 and 488. The generator polynomial of the 24-bits CRC is $g(x) = x^{24} + x^{23} + x^6 + x^5 + x + 1$. All experimental signals are transmitted under the B-DMC channel with additive white Gaussian noise. First, we analyze the decoding performance at different values by setting the size of the flip set. Second, the average complexity of each algorithm is given in this paper.

### A. BLER of Decoding

Fig. 3 compares the BLER performance of proposed algorithm with that of algorithms in [1] [3] for PC (256, 128). It can be seen that the performance of improved algorithm is gradually optimized as the sizes of sets $S_2'$ and $S_3'$ increase when performing multiple bit flips.

When $S_2' = 10$ and $S_3' = 10$, the performance of this algorithm surpasses that of original SC decoding algorithm and is close to the performance of the SCL decoding algorithm with the list size of $L = 4$. When $S_2' = 30$ and $S_3' = 10$, the BLER performance of the proposed algorithm is equivalent to that of SCL decoding algorithm with $L = 8$. When $S_2' = 30$ and $S_3' = 20$, the BLER performance is better than that of SCL decoding algorithm with the list size of $L = 8$.

Fig. 4 compares the BLER performance of the proposed algorithm with that of algorithms in [1] [3] for PC (1024,512). We can observe that when $S_2' = 10$ and $S_3' = 10$, the BLER performance of our algorithm is better than that of SCL decoding algorithm with the list size of $L = 2$. However, the performance improvement is not obvious with the further increase of $S_2'$ and $S_3'$ and the decoding times for low and medium SNR start to increase rapidly. Until $S_2' = 30$ and $S_3' = 20$, when the $SNR = 3$ dB, the BLER performance starts to be better than that of SCL decoding algorithm with the list size of $L = 4$.

### B. Average Complexity of Decoding

Fig. 5 and Fig. 6 compare the average complexity of the proposed algorithm with that of algorithms in [1] [3] for PC (256, 128) and PC (1024, 512). In this paper, complexity refers to the average list size in [5], and $F$ represents the total number of decoding frames and $T$ represents times. For example, the average complexity of the original SC decoding algorithm is ($T/F$), and the complexity of the SCL decoding algorithm with a list size of $L = 2$ is ($2T/F$). For the bit-flipping decoding algorithm, each additional flip means that the complexity is increased by 1. If the proposed flip decoding algorithm tries to flip additional t times, the complexity is ($T$+$t$)/$F$.

It can be observed from the Fig. 5 and Fig. 6 that the improved bit-flipping algorithm is more suitable for medium and high SNR than the original SCL decoding algorithm for PC (256, 128). The algorithm proposed in this paper has greater advantages in channels with high SNR for PC (1024, 512). In channels with low SNR, the complexity of the proposed algorithm is much higher than that of the original

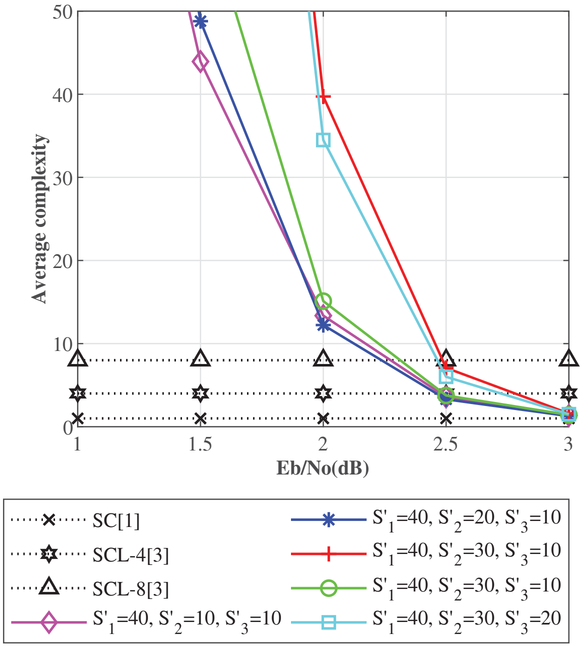


Fig. 5. Average complexity of decoding for PC (256, 128).

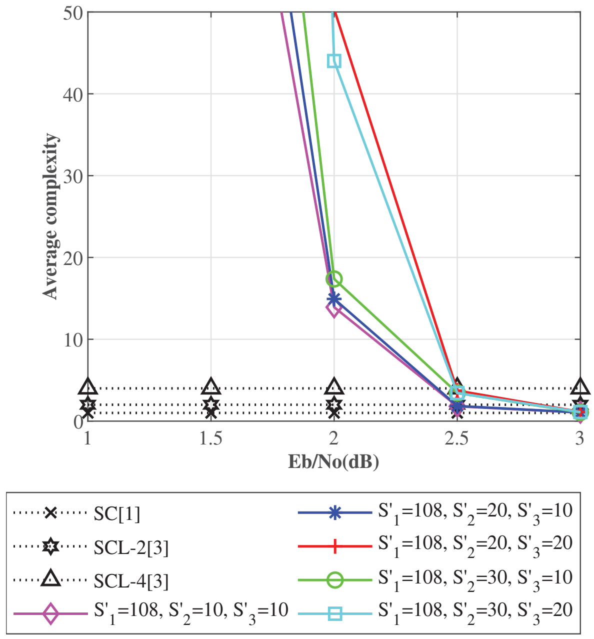


Fig. 6. Average complexity of decoding for PC (1024, 512).

decoding algorithm because the number of expected flips is drastically increased.

## V. Conclusions

In this paper, we propose an algorithm to improve the SCF decoding performance. A set of error-prone bits are obtained by counting the average and minimum value of the LLRs of the information bits. Based on this set, two other flip sets are established, and the flip bits in the three sets are arranged in ascending order of LLR. The improved SCF decoding algorithm can correct multiple errors. If the sizes of these sets are set properly, the BLER performance for PC (256, 128) is better than that of SCL decoding with the list size of $L$ = 8. However, the performance improvement of the proposed algorithm for PC (1024, 512) is not obvious for PC (256, 128), which may be resulted from the omission of some error-prone bits. However, When the bits of $S'_2$ and $S'_3$ increase, the time consumption of decoding under the channel with low or medium SNR will increase sharply, which requires us to trade off between the decoding performance and decoding time.

## Acknowledgement

This work was supported in part by the National Natural Science Foundation of China under Grant 62071306, in part by the Mobility Program for Taiwan Young Scientists under Grant RW2019TW001, and in part by Shenzhen Science and Technology Program under Grants GJHZ20180418190529516 and JSGG20180507183215520.

## References

[1] E. Arikan, "Channel polarization: A method for constructing capacity-achieving codes," in *Proc. 2008 IEEE International Symposium on Information Theory*, Toronto, ON, pp. 1173-1177, 2008.
[2] E. Arikan, "Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels," *IEEE Transactions on Information Theory*, vol. 55, no. 7, pp. 3051-3073, Jul. 2009.
[3] I. Tal and A. Vardy, "List decoding of polar codes," *IEEE Transactions on Information Theory*, vol. 61, no. 5, pp. 2213-2226, May 2015.
[4] K. Niu and K. Chen, "CRC-aided decoding of polar codes," *IEEE Communications Letters*, vol. 16, no. 10, pp. 1668-1671, October 2012.
[5] O. Afisiadis, A. Balatsoukas-Stimming and A. Burg, "A low-complexity improved successive cancellation decoder for polar codes," in *Proc. 2014 48th Asilomar Conference on Signals, Systems and Computers*, pp. 2116-2120, Nov. 2014
[6] Z. Zhang, K. Qin, L. Zhang, H. Zhang and G. T. Chen, "Progressive Bit-Flipping Decoding of Polar Codes over Layered Critical Sets," in *Proc. 2017 IEEE Global Communications Conference (GLOBECOM 2017)*, Singapore, pp. 1-6, 2017.
[7] L. Chandesris, V. Savin and D. Declercq, "Dynamic-SCFlip Decoding of Polar Codes," *IEEE Transactions on Communications*, vol. 66, no. 6, pp. 2333-2345, Jun. 2018.
[8] F. Ercan, C. Condo, S. A. Hashemi and W. J. Gross, "Partitioned Successive-Cancellation Flip Decoding of Polar Codes," in *Proc. 2018 IEEE International Conference on Communications (ICC 2018)*, Kansas City, MO, pp. 1-6, 2018.
[9] Y. Zhou, J. Lin and Z. Wang, "Improved Fast-SSC-Flip Decoding of Polar Codes," *IEEE Communications Letters*, vol. 23, no. 6, pp. 950-953, Jun. 2019.
[10] F. Ercan, C. Condo and W. J. Gross, "Improved Bit-Flipping Algorithm for Successive Cancellation Decoding of Polar Codes," *IEEE Transactions on Communications*, vol. 67, no. 1, pp. 61-72, Jan. 2019.