IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received December 15, 2019, accepted January 1, 2020, date of publication January 8, 2020, date of current version February 14, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2964904

Improved Multiple Bit-Flipping Fast-SSC Decoding of Polar Codes

XIUMIN WANG[1], TING WANG[1], JUN LI[2], AND YUE ZHANG[3]

[1]College of Information Engineering, China Jiliang University, Hangzhou 310018, China
[2]Binjiang College, Nanjing University of Information Science and Technology, Wuxi 214105, China
[3]Aerospace and Computational Engineering (ACE) Research Group, Department of Engineering, University of Leicester, Leicester LE1 7RH, U.K.

Corresponding authors: Jun Li (07a0303105@cjlu.edu.cn) and Yue Zhang (yue.zhang@leicester.ac.uk)

This work was supported in part by the Zhejiang Provincial Natural Science Foundation of China under Grant Y20F010069, in part by the National Natural Science Foundation of China (NSFC) under Grant 51874264, in part by the European Union (EU) Horizon 2020 5GPPP project on ''Internet of Radio-Light'' (IoRL) under Grant 761992, and in part by the Key Laboratory of Electromagnetic Wave Information Technology and Metrology of Zhejiang Province, College of Information Engineering, China Jiliang University, Hangzhou, China.

ABSTRACT Polar codes can provably achieve channel capacity and have been used in standards of the 5th generation wireless communication. As the first decoding algorithm of polar codes, successive cancellation (SC) suffers a significant error-correction performance loss at the moderate to short code length. The SC-Flip decoding aims to correct the first error in the SC decoding, and has the competitive error-correction performance compared with the SC decoding at the cost of high decoding latency. However, there may exist more than one errors in the SC decoding. Therefore, this paper proposes to enhance the fast-simplified SC (Fast-SSC) decoding to incorporate multiple bit-flipping, and further proposes two enhanced Fast-SSC-Flip decoding algorithms. One is a two-bit-flipping Fast-SSC (Fast-SSC-2Flip-E_2) decoding algorithm based on the distribution of the second error (E_2) in the Fast-SSC decoding, and is further expanded to flip multiple bits based on the distribution of multiple errors. The other is a partitioned Fast-SSC-Flip (PA-Fast-SSC-Flip) decoding algorithm. The Fast-SSC decoder tree is divided into several partitions, on which the Fast-SSC-Flip decoding is performed. Compared with the traditional Fast-SSC-Flip decoding, the proposed Fast-SSC-2Flip-E_2 has an error-correction performance gain up to 0.2 dB while keeping the average complexity close to that of the traditional Fast-SSC-Flip. The decoding speed of the PA-Fast-SSC-Flip is up to 6 times faster than that of traditional Fast-SSC-Flip, and has an error-correction performance gain up to 0.16dB.

INDEX TERMS Polar codes, error-correction performance, fast-simplified SC flip decoding, partitioned decoding, multiple bit-flipping.

I. INTRODUCTION

Polar codes proposed by Arikan [1] are proved to achieve channel capacity for different channels. They have been used in the control channel of the 5th generation wireless systems standards (5G) [2]. The successive cancellation (SC) decoding algorithm was firstly proposed in [1]. But it suffers a significant error-correction performance loss at moderate to short code length. Thus, two enhanced SC decoding methods, successive cancellation list (SCL) [3] and SC-Flip [4], were introduced to greatly improve the error-correction performance. Then a cyclic-redundancy-check (CRC) aided SCL decoding was proposed in [5] to further improve the decoding performance. The SCL decoding improves the performance at the cost of high computational complexity and resource consumption. Relatively, SC-Flip decoding can achieve the similar computational complexity of the SC decoding, while providing the error-correction performance of the SCL decoding at a small list size.

The SC-Flip decoding is a decoding scheme that significantly improves the performance by flipping one erroneous bit of invalid codewords, while keeping similar complexity as the SC decoding [4]. In [6], an enhanced SC-Flip decoding uses the optimized metric to find the first error, which produces better gains than [4]. Chandesris et al. in [7] put forward a dynamic SC-Flip (D-SCFlip) which is more

The associate editor coordinating the review of this manuscript and approving it for publication was Zesong Fei.

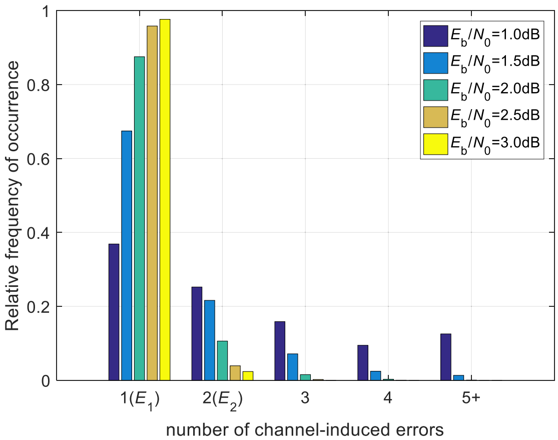


FIGURE 1. Frequency of occurrence of channel-induced errors at various E_b/N_0 for $P(512, 256)$, 5×10^5 simulated frames.

likely to correct the incorrect trajectory of the SC decoding. Later, two SC-Flip techniques based on the error distribution to improve the decoding performance for low-rate polar codes were proposed in [8]. A threshold SC-Flip (TSCF) decoding algorithm introduced in [9] investigates the distribution of the average log-likelihood ratio (LLR) related to the erroneous bit-channels. In [10], the partitioned SC-Flip (PA-SC-Flip) decoding targets at the correction of at least one single wrong decision with lower computational complexity than SC-Flip in [4]. Later in [11], [12], the segmentation method is further introduced in SC-Flip decoding. The progressive SC-Flip decoding proposed in [13] constructs a critical set (CS) including the first error bit and then develops a multiple-layer-flip decoding. The Fast-SSC-Flip decoding, which merges the SCF [4] with the fast-simplified SC (Fast-SSC) [14], [16], was proposed in [17]. The Fast-SSC-Flip decoding extends the flip idea in [4] to four types of special nodes. Then, two enhanced Fast-SSC-Flip decoding algorithms proposed in [18] and [19] use these new nodes [20] to further reduce the number of iterations in decoding.

The results in the literatures about Fast-SSC-Flip decoding target at a single bit error-correction [14], [16]. However, if there are more errors to be corrected, the exhaustive search in the critical set brings to a higher decoding complexity [13]. Fig. 1 depicts the frequency about channel-induced errors for $P(512, 256)$ with different E_b/N_0 and 5×10^5 simulated frames [4]. In Fig. 1, though most of decoding failures are due to a single error (E_1), there exist two errors that cannot be negligible. Therefore, in order to correct the second error (E_2), this paper proposes the Fast-SSC-2Flip-E_2 based on the distribution of E_2 occurrence. It has a better error-correction performance than one error flipping while keeping the average computational complexity close to one error flipping. This paper first proves the superiority of CS through a large number of statistical experiments. Then this paper investigates the distribution of E_2 with respect to E_1 occurrence index in CS, and proposes the Fast-SSC-2flip-E_2 decoding to flip two bits. The performance of the

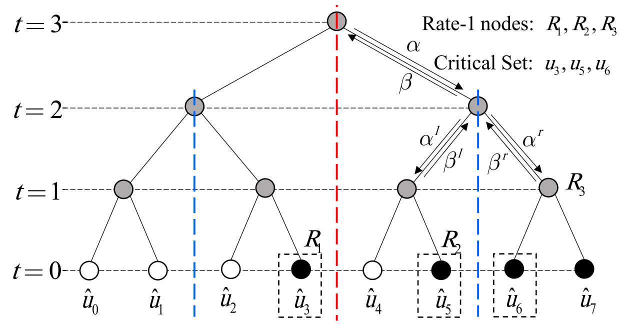


FIGURE 2. SC decoder tree representation for a $P(8, 4)$ polar code.

proposed Fast-SSC-2flip-E_2 outperforms that of Fast-SSC-Flip [17] while keeping the average complexity close to that of Fast-SSC-Flip. In addition, this paper also presents the partitioned Fast-SSC-Flip (PA-Fast-SSC-Flip) decoding. The Fast-SSC decoder tree is divided into S partitions, on which the Fast-SSC-Flip is run. The PA-Fast-SSC-Flip targets at the correction of at least one single error, with lower number of iterations and better performance than the Fast-SSC-Flip [17].

The remainder of this work is organized as follows: in Section II, polar codes and SC-based decoding algorithms are introduced. In Section III, the Fast-SSC-2Flip-E_2 and PA-Fast-SSC-Flip are detailed. The simulation results, together with comparison with the state of the art, are presented in Section IV. Conclusions are drawn in Section V.

II. PRELIMINARIES

A. POLAR CODES AND SC DECODING

A polar code $P(N, K)$ of code length N and rate $R = K/N$, constructed by concatenating two polar codes of length $N/2$, is a linear block code. The recursive construction can be represented as $\mathbf{x} = \mathbf{u}\mathbf{G}^{\otimes n}$, where $\mathbf{u} = \{u_0, u_1, \cdots, u_{N-1}\}$ is the input vector, $\mathbf{x} = \{x_1, x_2 \cdots, x_{N-1}\}$ is the encoded vector, and $\mathbf{G}^{\otimes n}$ is the generator matrix [1]. The input vector $\mathbf{u}$ is composed of K information bits transmitted in the reliable channels and $N - K$ frozen bits transmitted in the unreliable channels. The frozen bits with set F are set to a predefined value (usually 0) known by decoder.

The SC decoding algorithm can be represented as a binary decoder tree as shown in Fig. 2 for $P(8, 4)$ [1], where the white and black leaves correspond to the frozen bits and the information bits, respectively. The SC decoding process is from top to bottom, and left to right, sequentially estimating the $\hat{u}_i$ at the leaf nodes. For a node v of length 2^t, it receives the log-likelihood ratio (LLR) $\boldsymbol{\alpha} = \{\alpha_0, \alpha_1, \cdots, \alpha_{2^t-1}\}$ from its parent nodes, and computes the left $\boldsymbol{\alpha}^l = \left\{\alpha_0^l, \alpha_1^l, \cdots, \alpha_{2^{t-1}-1}^l\right\}$ and right $\boldsymbol{\alpha}^r = \left\{\alpha_0^r, \alpha_1^r, \cdots, \alpha_{2^{t-1}-1}^r\right\}$ to its child nodes as [9]

$$\alpha_i^l = \mathrm{sgn}(\alpha_i)\mathrm{sgn}(\alpha_{i+2^{t-1}})\min(\alpha_i, \alpha_{i+2^{t-1}}), \quad (1)$$

$$\alpha_i^r = \alpha_{i+2^{t-1}} + (1 - 2\beta_i^l)\alpha_i. \quad (2)$$

Node v computes the partial sum β_i according to the received left $\beta^l = \left\{\beta_0^l, \beta_1^l, \cdots, \beta_{2^{t-1}-1}^l\right\}$ and right $\beta^r = \left\{\beta_0^r, \beta_1^r, \cdots, \beta_{2^{t-1}-1}^r\right\}$ from the child nodes as follows

$$\beta_i = \begin{cases} \beta_i^l \oplus \beta_i^r, & \text{if } i \leq 2^{t-1}, \\ \beta_{i-2^{t-1}}^r, & \text{otherwise.} \end{cases} \tag{3}$$

where $\oplus$ is the bitwise XOR operation. At leaf nodes, the i-th bit $\hat{u}_i$ can be estimated as

$$\hat{u}_i = \begin{cases} 0, & \text{if } i \in F \text{ or } \alpha_i > 0, \\ 1, & \text{otherwise.} \end{cases} \tag{4}$$

B. SC-FLIP AND PARTITIONED SC-FLIP DECODING

In the SC decoding, once an error occurs, multiple bits following it may be incorrectly estimated due to the error propagation. In order to improve the error-correction performance of the SC decoding, the SC-Flip was proposed in [4]. The SC-Flip attempts the identification and correction of the first error caused by channel noise. It starts by executing a conventional SC decoding, during which an index set of T smallest absolute decision LLRs of information bits is built. When the SC decoding is completed, a C-bit CRC code is used to check the validity of the codeword. If the codewords pass the CRC, the decoding is finished and the estimated codewords are output. If the CRC fails, T additional decoding attempts are performed successively until the codewords pass the CRC or T attempts are reached.

In order to reduce the computational complexity of SC-Flip decoding, a partitioned SC-Flip (PA-SC-Flip) was presented [10]. Unlike SC-Flip decoding, the PA-SC-Flip is possible to correct more than one error since the codewords are divided into several partitions. The SC-Flip decoding is running on each partition and reduces the number of iterations. This paper extends the idea of partition to the Fast-SSC-Flip decoding, and outperforms the Fast-SSC-Flip decoding in terms of the error-correction performance and average computational complexity.

C. FAST-SSC AND FAST-SSC-FLIP DECODING

Due to the sequential nature of the SC decoding, the decoding latency is very high. A simplified SC (SSC) decoding was proposed in [14] to decode the Rate-1 nodes simultaneously. The Fast-SSC decoding adds the repetition (REP) nodes and single-parity-check (SPC) nodes in [15]. Later in [16], five types merge nodes are introduced to Fast-SSC decoding to further enhance the decoding speed of polar codes.

Since the Fast-SSC decoding has lower decoding latency, the Fast-SSC-Flip decoding [17] was presented by merging the Fast-SSC decoding [15] and the SC-Flip decoding [4], to improve the decoding speed of the SC-Flip decoding. The Fast-SSC-Flip uses the decision LLRs absolute values to measure the reliability of each information bit like the SC-Flip decoding. The difference is that the decision LLRs values are applied to Rate-1, REP, SPC and Type I nodes respectively. The specific modifications are described as follows [17]

1) Rate-1 Nodes
   Rate-1 nodes have length $N_v = 2^{t-1}$ and contain $k_v = 2^{t-1}$ information bits. The decision LLRs of each information bit is $\beta_s = |\alpha_s|$, where $0 \leq s < k_v$. If the first decoding trial fails and the index to be flipped falls in the Rate-1 nodes, the bit that corresponds to this index is flipped.
2) REP nodes
   REP nodes have only one single information bit on the right most index. After the first trial, in case the index to be flipped is the single information bit index, which is flipped.
3) SPC nodes
   SPC nodes only have one single frozen bit in the first location and others are all information bits with $k_v = N_v - 1$. Due to the parity constraint of SPC nodes, there are two cases to be distinguished. Let i_{flip} denote the index of initial bit to be flipped, and $i_{\min\ 1}$ and $i_{\min\ 2}$ denote the indices of the smallest and sub-smallest input LLRs respectively. If $i_{\text{flip}} = i_{\min\ 1}$, both the bits corresponding to i_{flip} and $i_{\min\ 2}$ are flipped, otherwise, both the bits corresponding to i_{flip} and $i_{\min\ 1}$ are flipped.
4) Type-I nodes
   Type-I nodes have two information bits at the right two positions and others are all frozen bits. Like the REP nodes, if the index to be flipped falls in one of the two information bits of the Type-I nodes, the corresponding bit is flipped after the first trial fails.

   Based on the Fast-SSC-flip decoding, this paper proposes Fast-SSC-2Flip-E$_2$ decoding to flip two bits in the Fast-SSC-flip decoding, and outperforms the Fast-SSC-flip in error-correction performance. And this method can be further expanded to flip multiple bits based on the distribution of multiple errors.

III. IMPROVED FAST-SSC-FLIP DECODING

Most of bit-flipping decoding algorithms target at a single error flipping. Fig. 1 shows that most of decoding failures are due to one single error, however, there still exist two errors which cannot be negligible. In addition, although the multiple layers bit-flipping decoding based on CS in [13] was proposed to flip more than one errors, the exhaustive search brings higher computational complexity. In this section, we first evaluate the superiority of CS through the statistical investigation and simulation results. Then, based on CS, we propose a Fast-SSC-2Flip-E$_2$ to flip two bits in the Fast-SSC-Flip decoding, and a PA-Fast-SSC-Flip to flip at least a single bit in the Fast-SSC-Flip decoding.

A. THE SUPERIORITY OF CS

The initial SC-Flip decoding needs to sort the LLRs of all the information bits and select T_{max} least reliable bits to construct

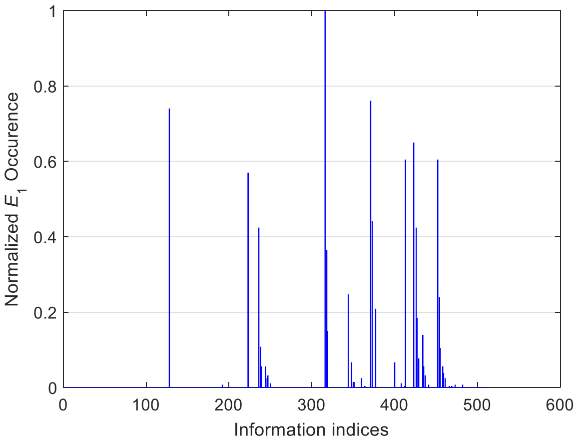


FIGURE 3. Normalized distribution of E_1 occurrence for $P(1024, 512)$, $C = 16$, $E_b/N_0 = 2.5$dB, 5×10^5 simulated frames.

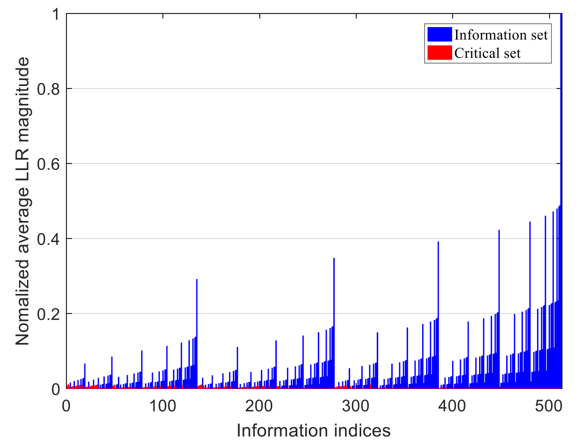


FIGURE 4. Normalized average LLR magnitude of each information bit index for $P(1024, 512)$, with $C = 16$ and $E_b/N_0 = 2.5$dB. Indices highlighted in red correspond to the CS.

a flipping set, which leads to substantial implementation cost. The CS in [13] selects the index of the first information bit in each Rate-1 node as its element. For instance, in Fig. 2, the Rate-1 nodes are R_1, R_2, and R_3, thus the critical set $CS = \{u_3, u_5, u_6\}$. According to the proposition [13], the CS has high probability including the first error that causes error propagation in the SC decoding, which reduces the sorting range of LLRs. The number of elements in CS is denoted as l.

Besides, l increases with the SNR according to the simulation results in [13]. In order to construct a fixed CS before decoding and obtain better error-correction performance, the CS containing more error-prone bit indices is selected at $E_b/N_0 = 3.0$dB which is the maximum SNR in the following simulation.

The normalized distribution of E_1 over all the information indices is depicted in Fig. 3 for $P(1024, 512)$, with $C = 16$ and 5×10^5 simulated frames at $E_b/N_0 = 2.5$dB. It can be noticed that only a few bits have higher probability of E_1, and no error happens in the majority of indices.

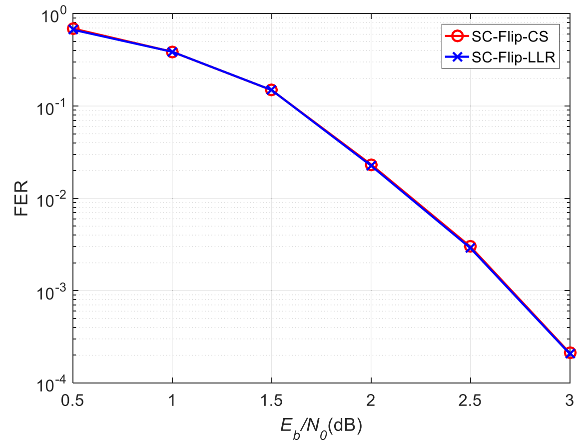


FIGURE 5. Error-correction performance for SC-Flip-LLR with $P(1024, 512)$, $T_{max} = 8$ and $C = 16$.

Besides, the normalized average LLR magnitude of each information bit index can be observed in Fig. 4 for $P(1024, 512)$, with $C = 16$ and $E_b/N_0 = 2.5$dB. Indices highlighted in red are the LLRs of the indices in CS. It can be seen that all the bits in CS have the lower LLRs, in other words, the bits in CS are unreliable enough to be flipped. The error-correction performance of the SC-Flip based on CS (SC-Flip-CS) and the SC-Flip based on the LLRs of all information bits (SC-Flip-LLR) is shown in Fig. 5 with $P(1024, 512)$, $T_{max} = 8$ and $C = 16$. It can be observed that the two curves are almost overlapped. It indicates that the SC-Flip-CS decoding can avoid the sorting of the LLRs of all information bits, while keeping the performance with the conventional SC-Flip decoding. In the rest of this work, we will introduce the CS to select the T_{max} least reliable bits in the proposed improved Fast-SSC-Flip decoding algorithms.

B. FAST-SSC-2FLIP-E_2 DECODING

Fig. 1 indicates that the second error cannot be ignored in the SC decoding. In the Fast-SSC-Flip decoding [17], if the codewords don't pass the CRC after flipping a bit, we are not sure whether the decoding failure is caused by the flipped bit or an erroneous bit which follows the flipped bit. In order to make it clear, based on CS, the SC-Oracle decoder with prior information is used to investigate the distribution of E_2 occurrence after flipping the first error whose index is in CS. The results are displayed in TABLE 1 for $P(1024, 512)$ with $C = 16$ and 5×10^5 simulated frames at $E_b/N_0 = 2.0$dB. The first line in TABLE 1 is the indices of E_1 in CS constructed previously according to the construction method in [13]. When the first error appears, we flip it and then wait for the second error following it to record the index of E_2. The $E_2(P)$ in TABLE 1 represents the sorted normalized probability P of E_2 occurrence. For a specific flipped index E_1 in CS, the distribution of E_2 is constructed in TABLE I. It can be seen from P which bit trends to become the second error after flipping a specific bit. Based on the

TABLE 1. The normalized distribution of E_2 correspond to E_1 in CS.

E_1	128	192	223	$\cdots$
$E_2(P)$	236(1)	239(1)	599(1)	$\cdots$
	371(0.64)	223(0.75)	236(0.94)	$\cdots$
	223(0.59)	238(0.75)	413(0.94)	$\cdots$
	344(0.55)	371(0.75)	238(0.69)	$\cdots$
	602(0.54)	452(0.75)	239(0.63)	$\cdots$
	599(0.50)	434(0.50)	452(0.63)	$\cdots$
	$\vdots$	$\vdots$	$\vdots$	$\ddots$

distribution of E_2, we propose the Fast-SSC-2Flip-E$_2$ decoding algorithm.

The SPC nodes in Fast-SSC-Flip decoding [17] are computed approximately, which will lead to some performance loss, thus only Rate-1, REP, and Type-I nodes are added in this work. The Fast-SSC-2Flip-E$_2$ decoding process is described in algorithm 1. The parameters in algorithm 1 are predefined before decoding, where A, T_1 and T_2 are the information bit indices, the first and the second maximum iterations numbers, respectively. Before decoding, the CS with l elements is first constructed as the Section III-A describes, and E_1 in CS and the corresponding E_2 with T_2 highest probability in TABLE 1 make up the index matrix IS. The dimension of IS is $(T_2+1)\times l$. Then the Fast-SSC-2Flip-E$_2$ decoding starts by performing the Fast-SSC with three kinds of simplified nodes, during which the LLRs set $L(CS)$ with the indices in CS is constructed. If the codewords pass the CRC, the decoding is finished. If the CRC fails, T_1 least reliable LLRs are first selected from the sorted $L(CS)$, and then the T_1 additional flipping attempts will flip the bit of the special nodes or leaf nodes as the Section II-C discusses. In the every T_1 attempt, the LLRs set $L(IS)$ of T_2 predefined indices is constructed. Once the CRC fails again in every T_1 attempts, the $L(IS)$ is sorted and T_2 additional attempts will correct the second error, until both T_1 and T_2 are reached or the CRC passes. With the help of IS, the Fast-SSC-2Flip-E$_2$ decoding can correct two errors, which further improves the error-correction performance.

In order to verify the validity of IS, we also propose a Fast-SSC-2Flip-CS decoding algorithm, in which the flipping of the second error is based on the updated CS. After flipping the first error, all bit indices following it in CS form the updated CS. In the Fast-SSC-2Flip-CS decoding,T_2 least reliable LLRs are selected from the sorted updated CS, which is different from the Fast-SSC-2Flip-E$_2$ decoding.

In this paper, the proposed Fast-SSC-2Flip-E$_2$ decoding is a two-bit-flip decoding, however, it also can be expanded to flip multiple bits. The maximum number of flipping bits is denoted as n, the multiple errors are denoted as $E_1, E_2, \cdots, E_n$, the multiple index matrices are denoted as $IS_1, IS_2, \cdots, IS_{n1}$, and the multiple maximum iteration numbers are denoted as $T_1, T_2, \cdots T_n$. Firstly, the SC-Oracle decoder is used to obtain the index matrix IS_1 constructed by

---

Algorithm 1 Fast-SSC-2Flip-E$_2$ Decoding

Input: $y_0^{N-1}, T_1, T_2, A, CS_0^{l-1}, IS_{(T_2+1)\times l}$
Output: u_0^{N-1}
$(u_0^{N-1}, L(CS_0^{l-1})) \leftarrow$ Fast-SSC$(y_0^{N-1}, A, CS_0^{l-1})$;
if $T_1 > 1$ **and** $CRC(\hat{u}_0^{N-1}) = failure$ **then**
  $U_1 \leftarrow i_1 \in CS_0^{l-1}$ of T_1 smallest $\left|L(CS_0^{l-1})\right|$;
  for $j_1 \leftarrow 1$ **to** T_1 **do**
    $k_1 \leftarrow U_1(j_1)$;
    $(u_0^{N-1}, L(IS[2:, k_1])) \leftarrow$ Fast-SSC-Flip $(y_0^{N-1}, A, IS[2:, k_1], k_1)$;
    if $CRC(\hat{u}_0^{N-1}) = success$ **then**
      break;
    else
      $U_2 \leftarrow i_2 \in IS[2:, k_1]$ of *sorted* $|L(IS[2:, k_1])|$;
      for$j_2 \leftarrow 1$ **to**T_2 **do**
        $k_2 \leftarrow U_2(j_2)$;
        $u_0^{N-1} \leftarrow$ Fast-SSC-Flip$(y_0^{N-1}, A, IS[2:, k_1], k_2)$;
        if $CRC(\hat{u}_0^{N-1}) = success$ **then**
          break;
        end if
      end for
      if $CRC(\hat{u}_0^{N-1}) = success$ **then**
          break;
      end if
    end if
  end for
end if
return u_0^{N-1};

E_1 and E_2, IS_2 constructed by E_2 and $E_3, \ldots, IS_{n1}$ constructed by E_{n-1} and E_n. Then, the flipping method of Fast-SSC-2Flip-E$_2$ in Algorithm 1 can be expanded to flip n bits based on the $E_1, E_2, \cdots, E_n$.

C. PARTITIONED FAST-SSC-FLIP DECODING

In the PA-SC-Flip decoding [10], the codewords are divided into several partitions, which reduces the average number of iterations while keeping the similar error-correction performance with the SC-Flip decoding in [4]. Therefore, this paper introduces the idea of partition to the Fast-SSC-Flip decoding presented in [17], and propose a partitioned Fast-SSC-Flip decoding algorithm, where the decoder tree is broken into S partitions.

Since the Fast-SSC decoding is based on the simplified calculation of three kind of nodes identified in Section II-C, the integrity of the special nodes should be considered when selecting the partition location. Therefore, the decoder tree is divided into S partitions uniformly, which is first introduced in [21] to partition the SCL decoding tree. For instance, in Fig. 2, the decoder tree is broken into two partitions by the red dotted line, and four partitions by the blue dotted line. In addition, the number of CRC bits are distributed equally in the end of each partition, and the CS is divided into S partitions according to the partition indices.

Algorithm 2 PA-Fast-SSC-Flip Decoding

Input: $y_0^{N-1}, T_m, S, A, \eta_0^S, \lambda_0^S CS_0^{l-1}$

Output: u_0^{N-1}

1: $\eta[0] = 0, \lambda[0] = 0$

2: **for** $j \leftarrow 1$ **to** S **do**

3: **for** $i \leftarrow \eta[j-1]$ **to** $\eta[j]$ **do**

4: $(u_{\eta[j-1]}^{\eta[j]}, L(CS_{\lambda[j-1]}^{\lambda[j]})) \leftarrow$ Fast-SSC$(y_0^{N-1}, A, CS_{\lambda[j-1]}^{\lambda[j]})$;

5: **if** $T_m > 1$ **and** $CRC(\hat{u}_{\eta[j-1]}^{\eta[j]}) = failure$ **then**

6: $U \leftarrow m \in CS_{\lambda[j-1]}^{\lambda[j]}$ of T_m smallest $\left|L(CS_{\lambda[j-1]}^{\lambda[j]})\right|$;

7: **for** $t \leftarrow 1$ **to** T_m **do**

8: $k \leftarrow U(t)$;

9: $u_{\eta[j-1]}^{\eta[j]} \leftarrow$ Fast-SSC-Flip(y_0^{N-1}, A, k);

10: **if** $CRC(\hat{u}_{\eta[j-1]}^{\eta[j]}) = success$ **then**

11: break;

12: **end if**

13: **end for**

14: **end if**

15: **end for**

16: **end for**

17: **return** u_0^{N-1};

The PA-Fast-SSC-Flip decoding process is described in algorithm 2. The information indices A, the partition indices η in code length N, the maximum iteration number T_m, the partition indices λ in CS, are predetermined by the decoder. For each partition, the Fast-SSC decoding is executed first, during which the LLRs of the indices in CS are calculated (line 4), followed by the computation of CRC remainder (line 5). If the CRC detects an error in the first time decoding, then the index with the least reliable LLRs in the T_m sorted unreliable information bits in CS, is identified (line 6). For the maximum iteration number T_m, the Fast-SSC is carried out and the index with the $t-$th least reliable LLRs is flipped until the CRC succeeds or the T_m iterations are reached (line 7-12). The codeword u_0^{N-1} will be output in the end.

IV. SIMULATION RESULTS

In this section, we evaluate the error-correction performance and the extra number of iterations for the proposed Fast-SSC-2Flip-E$_2$ and PA-Fast-SSC-Flip. Polar codes $P(512, 256)$ and $P(1024, 512)$ are constructed targeting at a SNR of 2.5dB under AWGN channel. Simulations are performed with BPSK modulation and AWGN channel. The simulation results of the Fast-SSC-2Flip-E$_2$ and PA-Fast-SSC-Flip decoding are discussed in the following two parts respectively.

A. THE FAST-SSC-2FLIP-E$_2$ DECODING

The error-correction performance of Fast-SSC-2Flip-E$_2$ is compared with SC-Flip [4], Fast-SSC-Flip [17], D-SCFlip [7], Fast-SSC-2Flip-CS and SC-Oracle. The length of CRC code is 16, and the CRC generator polynomial is $g(x) = x^{16} + x^{15} + x^2 + 1$. The maximum iteration number for contrastive algorithms is $T_{max} = 8$.

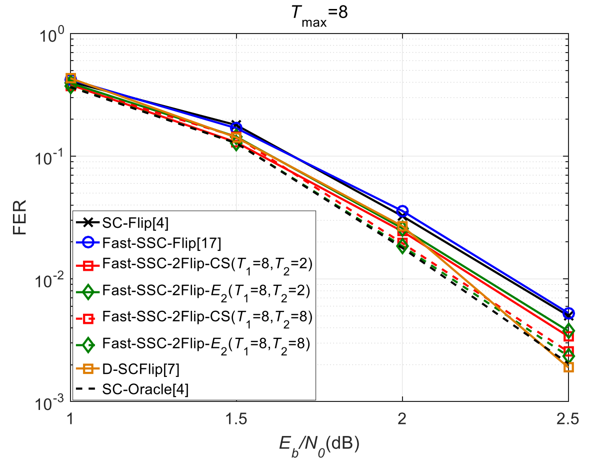


FIGURE 6. Error-correction performance comparison for Fast-SSC-2Flip-E$_2$ and contrastive decoding algorithms for a $P(512, 256)$ polar code with $T_{max} = 8$.

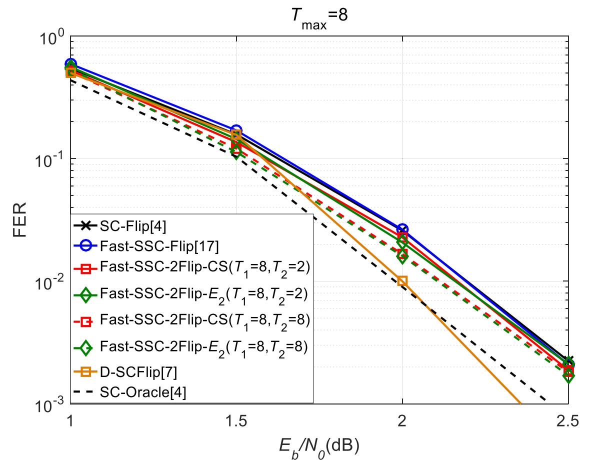


FIGURE 7. Error-correction performance comparison for Fast-SSC-2Flip-E$_2$ and contrastive decoding algorithms for a $P(1024, 512)$ polar code with $T_{max} = 8$.

In Fig. 6 we consider the error-correction performance for $P(512, 256)$, where $T_1 = 8$ and $T_2 = 2, 8$. It can be seen that the proposed Fast-SSC-2Flip-E$_2$ with $T_1 = 8$ and $T_2 = 2$ performs a better error-correction performance than Fast-SSC-Flip [17] with $T_{max} = 8$. When $T_1 = T_2 = 8$, the error-correction performance of Fast-SSC-2Flip-E$_2$ is close to that of SC-Oracle and has around 0.2 dB performance gain compared with the Fast-SSC-Flip [17]. When compared with the Fast-SSC-2Flip-CS for different T_2, the Fast-SSC-2Flip-E$_2$ always has no performance loss, which proves the validity of E_2. Compared with the D-SCFlip [7] which can correct multiple errors, the Fast-SSC-2Flip-E$_2$ outperforms the D-SCFlip [7] at moderate to low E_b/N_0 and is inferior to the D-SCFlip [7] at higher E_b/N_0. But the average computational complexity of the Fast-SSC-2Flip-E$_2$ is lower than that of D-SCFlip [7]. In Fig. 7, the code length is changed into 1024. Except for higher E_b/N_0, the error-correction performance of Fast-SSC-2Flip-E$_2$ with $T_1 = T_2 = 8$ outperforms that of D-SCFlip [7] and that of Fast-SSC-Flip [17]. For the

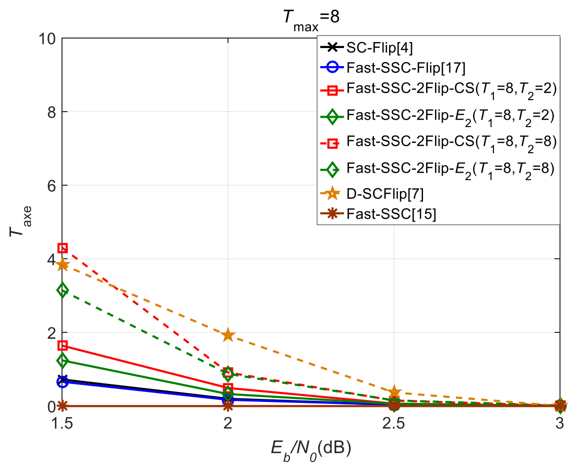


FIGURE 8. T_{axe1} **for Fast-SSC-2Flip-E$_2$ and contrastive decoding algorithms for a** $P(512, 256)$ **polar code with** $T_{max} = 8$.

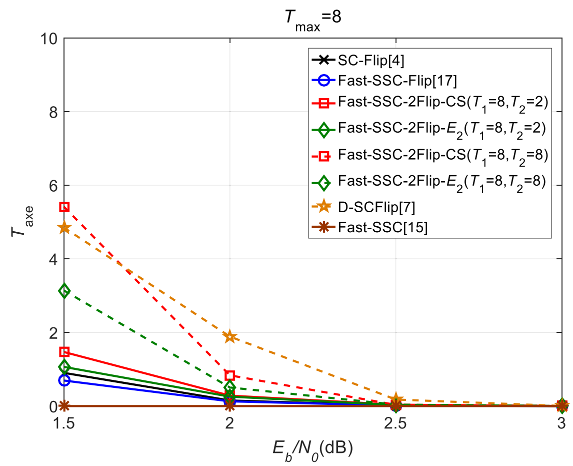


FIGURE 9. T_{axe1} **for Fast-SSC-2Flip-E$_2$ and contrastive decoding algorithms for a** $P(1024, 512)$ **polar code with** $T_{max} = 8$.

Fast-SSC-2Flip-E$_2$ with $T_1 = T_2 = 8$, the performance gain is about 0.1 dB at FER $= 10^{-1}$ compared with the Fast-SSC-Flip [17]. The performance gain of the Fast-SSC-2Flip-E$_2$ for $N = 1024$ is not so much impressive compared with that under $N = 512$. This is because that the degree of channel polarization for $N = 1024$ is better than that for $N = 512$. Therefore, for $N = 1024$, in most cases, the first flipping is enough to correct the error bit, and there is no need to flip the second one. For $N = 512$, it is necessary to flip the second error.

In this work, the average number of extra iterations T_{axe1} is used to represent the average computational complexity of the proposed Fast-SSC-2Flip-E$_2$. If the first Fast-SSC decoding cannot pass the CRC, more iterations are required. Except for the first iteration, the other iterations are called extra iterations. In each extra iteration, the decoding is performed from the index of the current bit to be flipped. Thus, the average number of extra iterations is denoted as

$$T_{axe1} = \sum\nolimits_{i=1}^{I} \frac{N - Ind_i}{N} / efr \tag{5}$$

where I is the number of extra iterations,Ind_i is the index of the current bit to flip, N is the code length, and efr is the number of erroneous frames. T_{axe1} is a criteria to indicate intuitively how many times of complete iterative decoding need to be executed after the first decoding trial fails. Fig. 8 and Fig. 9 show T_{axe1} for $P(512, 256)$ and $P(1024, 512)$ respectively. In Fig. 8, it can be seen that the proposed Fast-SSC-2Flip-E$_2$ with $T_1 = T_2 = 8$ suffers from higher T_{axe1} at low E_b/N_0. But as E_b/N_0 increases, T_{axe1} of Fast-SSC-2Flip-E$_2$ converges to that of Fast-SSC as 0. With $T_1 = 8$ and $T_2 = 2$, T_{axe1} of the proposed Fast-SSC-2Flip-E$_2$ is close to that of Fast-SSC-Flip [17] at $E_b/N_0 = 2.0$dB, but the Fast-SSC-2Flip-E$_2$ has better error-correction performance than Fast-SSC-Flip [17]. In addition, the Fast-SSC-2Flip-E$_2$ has a lower T_{axe1} than both Fast-SSC-2Flip-CS and D-SCFlip [7]. In Fig. 9, we can observe that the comparison of the T_{axe1} of the proposed Fast-SSC-2Flip-E$_2$ over contrastive algorithms for $P(512, 256)$ can be extended to the situation for $P(1024, 512)$.

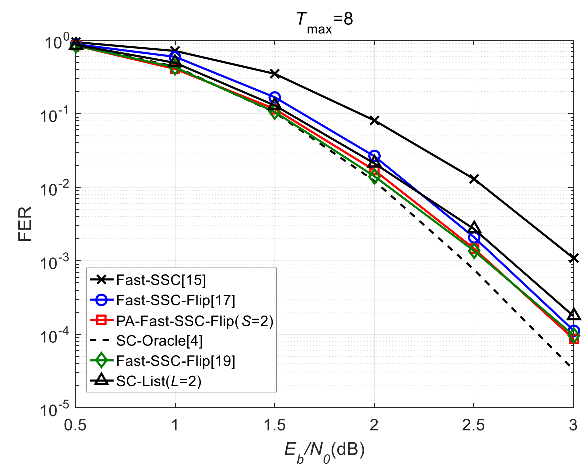


FIGURE 10. Error-correction performance comparison for PA-Fast-SSC-Flip and contrastive decoding algorithms for a $P(1024, 512)$ **polar code with** $T_{max} = 8$, $S = 2$.

B. THE PA-FAST-SSC-FLIP DECODING

The error-correction performance of PA-Fast-SSC-Flip is compared with Fast-SSC, Fast-SSC-Flip [17], Fast-SSC-Flip [19] and SC-Oracle. The length of CRC code in each partition is 8 and the CRC generator polynomial is $g(x) = x^8 + x^2 + x + 1$. For the contrastive decoding algorithms, the length of CRC is 16 and the CRC generator polynomial is $g(x) = x^{16} + x^{15} + x^2 + 1$.

The error-correction performance is shown in Fig. 10 for $P(1024, 512)$, $T_{max} = 8$, with the number of partitions $S = 2$. Compared with the Fast-SSC-Flip [17], the proposed PA-Fast-SSC-Flip has a better error-correction performance all the time, the performance gain is 0.16dB at FER $= 10^{-1}$. This is because the PA-Fast-SSC-Flip has the ability to flip up to $S \times T_{max}$ bits, enhancing the probability of identifying the error location. When compared with the

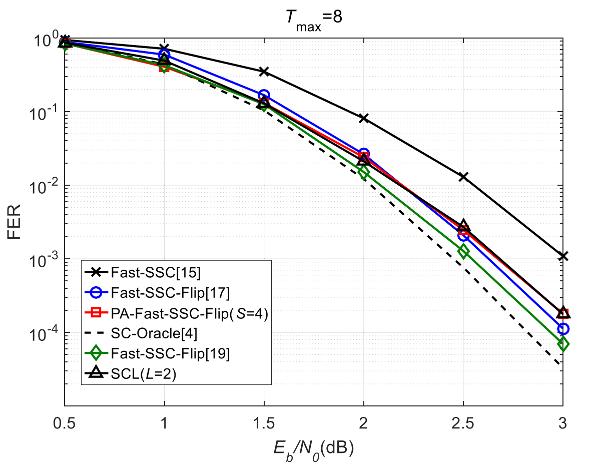


FIGURE 11. **Error-correction performance comparison for PA-Fast-SSC-Flip and contrastive decoding algorithms for a $P(1024, 512)$ polar code with $T_{\max} = 8$, $S = 4$.**

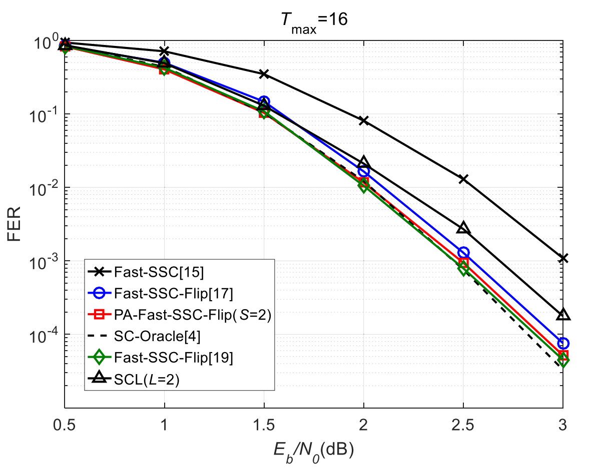


FIGURE 12. **Error-correction performance comparison for PA-Fast-SSC-Flip and contrastive decoding algorithms for a $P(1024, 512)$ polar code with $T_{\max} = 16$, $S = 2$.**

Fast-SSC-Flip [19], the error-correction performance of PA-Fast-SSC-Flip is close to that of Fast-SSC-Flip [19]. It also can be observed that at low E_b/N_0, the PA-Fast-SSC-Flip performance is close to the SC-Oracle, since it can correct more than one errors. The advantage of PA-Fast-SSC-Flip over SC-Oracle decreases as E_b/N_0 increases. This can be explained in Fig. 1 that the probability of a single channel-induced error increases as E_b/N_0 increases. In Fig. 11, the number of partitions is changed into $S = 4$, it can be noticed that the advantage of PA-Fast-SSC-Flip over the Fast-SSC-Flip [17] decreases at $E_b/N_0 = 2.25$dB, and advantage of PA-Fast-SSC-Flip over the Fast-SSC-Flip [19] decreases at $E_b/N_0 = 1.5$dB. This is because the CRC distribution becomes more sub-optimal with increasing partition S. In addition, the error-correction performance is given in Fig. 12 and Fig. 13 for $P(1024, 512)$, $S = 2, 4$ and $T_{\max} = 16$. We can observe that the error-correction performance of PA-Fast-SSC-Flip with $T_{\max} = 8$ over Fast-SSC-Flip can be extended to the situation of $T_{\max} = 16$.

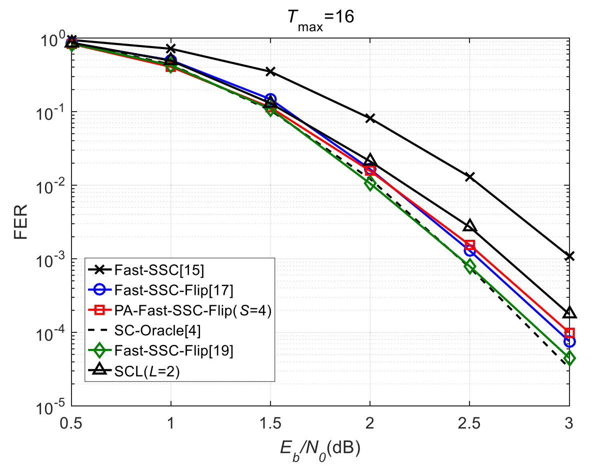


FIGURE 13. **Error-correction performance comparison for PA-Fast-SSC-Flip and contrastive decoding algorithms for a $P(1024, 512)$ polar code with $T_{\max} = 16$, $S = 4$.**

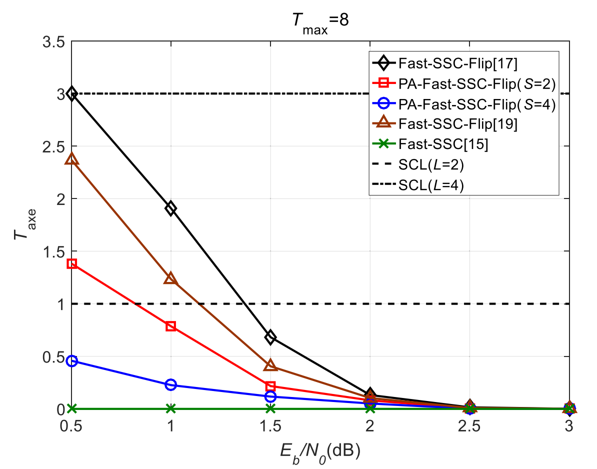


FIGURE 14. **T_{axe2} for PA-Fast-SSC-Flip and contrastive decoding algorithms for a $P(1024, 512)$ polar code with $T_{\max} = 8$.**

In this part, the average number of extra iterations T_{axe2} is also used to represent the average computational complexity of the proposed PA-Fast-SSC-Flip. However, there are some differences between PA-Fast-SSC-Flip and Fast-SSC-2Flip-E_2 due to the partitioned decoding. For the PA-Fast-SSC-Flip, T_{axe2} is denoted as

$$T_{axe2} = \sum\nolimits_{i=1}^{I} \frac{P_{ind} - Ind_i}{P_{ind}} / efr \tag{6}$$

where I is the number of extra iterations,Ind_i is the index of the current bit to be flipped, P_{ind} is the last bit index of the current partition, and efr is the number of erroneous frames. T_{axe2} is shown in Fig. 14 and Fig. 15 for $P(1024, 512)$, $T_{\max} = 8, 16$ and $S = 2, 4$, respectively. It can be seen that at low E_b/N_0, T_{axe2} of PA-Fast-SSC-Flip is as high as that of SCL decoding with both list size $L = 2$ for $T_{\max} = 8$ in Fig. 14 and list size $L = 4$ for $T_{\max} = 16$ in Fig. 15. On the other hand, the worst case T_{axe2} of PA-Fast-SSC-Flip with $S = 2$ is only 50% about that of Fast-SSC-Flip [17] and 60% about that of Fast-SSC-Flip [19], for both $T_{\max} = 8$

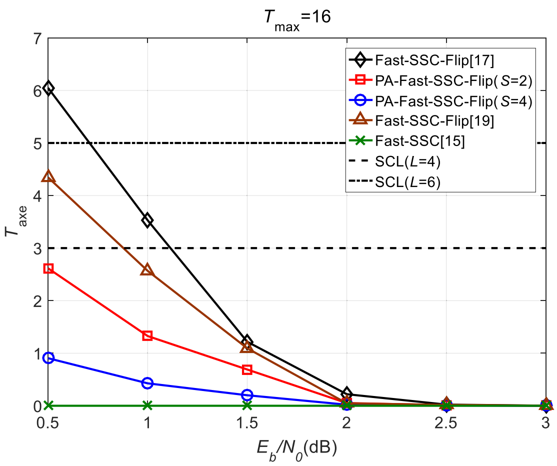


FIGURE 15. T_{axe2} **for PA-Fast-SSC-Flip and contrastive decoding algorithms for a** $P(1024, 512)$ **polar code with** $T_{max} = 16$.

and $T_{max} = 16$. At higher E_b/N_0, T_{axe2} of PA-Fast-SSC-Flip converges to that of Fast-SSC as 0. In Fig. 14, compared with the speed of the Fast-SSC-Flip [17], the speed of PA-Fast-SSC-Flip is up to 2.2 times faster with $S = 2$, and up to 6 times faster with $S = 4$. Similarly, in Fig. 15, it also can be obtained that the speed of PA-Fast-SSC-Flip is up to 2.4 times faster than that of Fast-SSC-Flip [17] with $S = 2$, and up to 6 times faster with $S = 4$.

V. CONCLUSION

In this work, two enhanced multiple bit-flipping Fast-SSC decoding algorithms are described and analyzed. For the Fast-SSC-2Flip-E_2, we first concentrate on the distribution of the second error (E_2) after flipping the first error bit in CS, and then propose the Fast-SSC-2Flip-E_2 decoding algorithm based on the E_2. In addition, the proposed Fast-SSC-2Flip-E_2 also can be expanded to flip multiple bits based on $E_1, E_2, \cdots, E_n$. For the PA-Fast-SSC-Flip, we first divide the decoder tree into S partitions, on which the Fast-SSC-Flip decoding is run. The simulation results show that the Fast-SSC-2Flip-E_2 has performance gain up to 0.2 dB while keeping the computational complexity close to the traditional Fast-SSC-Flip decoding. The decoding speed of the PA-Fast-SSC-Flip is up to 6 times faster than that of traditional Fast-SSC-Flip, and has an error-correction performance gain up to 0.16dB.

REFERENCES

[1] E. Arikan, "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] *Final Report of 3GPP TSG RAN1 #86bis V1.0.0*, document R1-1611081, Reno, NV, USA, Nov. 2016. [Online]. Available: https://www.3gpp.org/ftp/tsg_ran/WG1_RL1/TSGR1_87/Docs/R1-1611081.zip

[3] I. Tal and A. Vardy, "List decoding of polar codes," *IEEE Trans. Inf. Theory*, vol. 61, no. 5, pp. 2213–2226, May 2015.

[4] O. Afisiadis, A. Balatsoukas-Stimming, and A. Burg, "A low-complexity improved successive cancellation decoder for polar codes," in *Proc. 48th Asilomar Conf. Signals, Syst. Comput.*, Nov. 2014, pp. 2116–2120.

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

[6] L. Chandesris, V. Savin, and D. Declercq, "An improved SCFlip decoder for polar codes," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2016, pp. 1–6.

[7] L. Chandesris, V. Savin, and D. Declercq, "Dynamic-SCFlip decoding of polar codes," *IEEE Trans. Commun.*, vol. 66, no. 6, pp. 2333–2345, Jun. 2018.

[8] C. Condo, F. Ercan, and W. J. Gross, "Improved successive cancellation flip decoding of polar codes based on error distribution," in *Proc. IEEE Wireless Commun. Netw. Conf. Workshops (WCNCW)*, Apr. 2018, pp. 19–24.

[9] F. Ercan, C. Condo, and W. J. Gross, "Improved bit-flipping algorithm for successive cancellation decoding of polar codes," *IEEE Trans. Commun.*, vol. 67, no. 1, pp. 61–72, Jan. 2019.

[10] F. Ercan, C. Condo, S. A. Hashemi, and W. J. Gross, "Partitioned successive-cancellation flip decoding of polar codes," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2018, pp. 1–6.

[11] R. Guo, K. Chen, and H. Liu, "Multi-CRC polar codes and M-SCFlip-based decoding," *IEEE Access*, vol. 7, pp. 98366–98373, 2019.

[12] S. Li, Y. Deng, X. Gao, H. Li, L. Guo, and Z. Dong, "Generalized segmented bit-flipping scheme for successive cancellation decoding of polar codes with cyclic redundancy check," *IEEE Access*, vol. 7, pp. 83424–83436, 2019.

[13] 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. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2017, pp. 1–6.

[14] 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.

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

[16] P. Giard, A. Balatsoukas-Stimming, G. Sarkis, C. Thibeault, and W. J. Gross, "Fast low-complexity decoders for low-rate polar codes," *J. Signal Process. Syst.*, vol. 90, no. 5, pp. 675–685, May 2018.

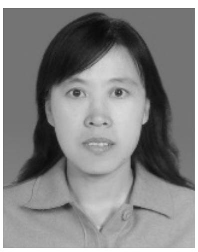
[17] P. Giard and A. Burg, "Fast-SSC-flip decoding of polar codes," in *Proc. IEEE Wireless Commun. Netw. Conf. Workshops (WCNCW)*, Apr. 2018, pp. 73–77.

[18] Y. Zhou, J. Lin, and Z. Wang, "A new fast-SSC-flip decoding of polar codes," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Shanghai, China, May 2019, pp. 1–6.

[19] Y. Zhou, J. Lin, and Z. Wang, "Improved fast-SSC-flip decoding of polar codes," in *IEEE Commun. Lett.*, vol. 23, no. 6, pp. 950–953, Jun. 2019.

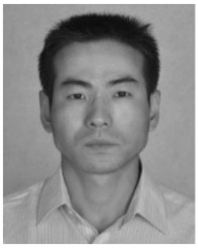
[20] M. Hanif and M. Ardakani, "Fast successive-cancellation decoding of polar codes: Identification and decoding of new nodes," *IEEE Commun. Lett.*, vol. 21, no. 11, pp. 2360–2363, Aug. 2017.

[21] S. A. Hashemi, A. Balatsoukas-Stimming, P. Giard, C. Thibeault, and W. J. Gross, "Partitioned successive-cancellation list decoding of polar codes," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Mar. 2016, pp. 957–960.



XIUMIN WANG was born in April 1963. She received the master's degree in communication and electronic system from the Dalian University of Technology. She is currently a Professor with the College of Information Engineering, China Jiliang University. Her research interests include channel coding, signal detection, and signal and information processing.

TING WANG received the B.E. degree in communication engineering from Hubei Polytechnic University, in 2017. She is currently pursuing the master's degree with the College of Information Engineering, China Jiliang University. Her research interest includes signal and information processing.



JUN LI was born in February 1977. He received the Ph.D. degree in communication and information system from the South China University of Technology, in 2005. He is currently an Associate Professor with the Binjiang College, Nanjing University of Information Science and Technology. His research interests include channel coding, LTE physical layer standard, and wireless resource optimization.

YUE ZHANG received the B.E. and M.E. degrees from the Beijing University of Post and Telecommunications, China, in 2001 and 2004 respectively, and the Ph.D. degree from Brunel University, U.K., in 2008. He is currently an Associate Professor with the Department of Engineering, University of Leicester. His research interests are signal processing, multimedia, and wireless networks.