IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received May 21, 2019, accepted June 9, 2019, date of publication June 13, 2019, date of current version July 11, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2922664

Generalized Segmented Bit-Flipping Scheme for Successive Cancellation Decoding of Polar Codes With Cyclic Redundancy Check

SHIBAO LI, YUNQIANG DENG, XUN GAO, HE LI, LIN GUO, AND ZHENWEI DONG
College of Computer and Communication Engineering, China University of Petroleum, Qingdao 266555, China

Corresponding author: Shibao Li (lishibao@upc.edu.cn)

This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant 61872385, and in part by the Fundamental Research Funds for the Central University under Grant 18CX02134A, Grant 18CX02137A, and Grant 19CX05003A-4.

ABSTRACT As a new flavor for the successive cancellation decoding of polar codes, bit-flipping technology can be used to improve the frame error rate performance of polar codes at moderate code lengths. In this paper, a generalized segmented scheme of bit-flipping technology is taken into full consideration, in which several segmented decoding patterns are described. First, the basic ones considering single-error-correcting and multiple-error-correcting are described, where a fully protected CRC is used. Second, with a constructed critical set for the segmented scheme, the improved version of the above basic patterns is fully discussed. Unfrozen bits corresponding to the critical set called critical information bits are divided into several parts and protected by cyclic redundancy check codes. Third, for a better performance of the segmented scheme, a segmented strategy indicating the positions of segments is effectively designed by analyzing the effect of positions of cyclic redundancy check and the error probability of each bit. At last, the frame error rate performance and the average computational complexity of the segmented decoding patterns are analyzed at matching average computational complexity or under equivalent frame error rate for different rates of polar codes. By proper design, the simulation results in additive white Gaussian noise channel demonstrated that the segmented decoding patterns can efficiently improve the error-correction performance of the flip-based scheme at different rates of polar codes while keeping a lower complexity.

INDEX TERMS Polar codes, segmented scheme, successive cancellation flip.

I. INTRODUCTION

Polar codes, discovered by Arikan in [1], are a class of error-correcting codes proven to achieve channel capacity for various classes of channels [2], [3]. Construction of polar codes is very explicit and decoding of them can be efficiently performed with a low-complexity successive cancellation (SC) decoding [4], [5]. Thus, polar codes are attractive from a theoretical and practical point. Recently, they have been selected as the control channel coding candidate in the fifth generation (5G) mobile communications [6]–[8].

In order to promote the application of polar codes in practice, researchers concentrate on the design of various sophisticated decoders. Belief propagation (BP) decoder and SC decoder [5] are two concerned basic decoders. Although BP decoder has better BER performance and is inherently parallel, it requires a lot of memory space and the improvement is not satisfying [9]. Under SC decoding, polar codes can approach channel capacity at infinite code lengths [1]. However, they suffer from a degraded error-correcting performance at short to moderate code-lengths as the polarization effect degrades in this regime [10]. To tackle this issue, various sophisticated decoding algorithms are proposed in the literature [10]–[18]. SC-List decoding algorithm [10]–[14] is introduced to significantly improve the error-correction performance of the SC decoder, in which multiple decoding paths are considered concurrently at each decoding and the most likely among the paths is selected as the output. With the help of the cyclic redundancy check (CRC), the CRC-aided SC-List decoder [15], [16] selects the final codeword passing the CRC check among the candidate paths, further improving the performance of polar codes. However, SC-List based

The associate editor coordinating the review of this manuscript and approving it for publication was Muhammad Ali Imran.

decoders lead to high computational and storage complexity. The SC-Stack (SCS) [17] decoding algorithm aims to reduce the computational complexity of the SC-List decoding at the cost of the increasing storage complexity. Besides, adaptive SC-List (AD-SC-List) decoder [18] iteratively increases the list size until at least one survival path can pass CRC and Reused-Public-Path SC-List (RPP-SC-List) [19] can re-do the decoding from E-th level when the current decoding process fails. Both of them effectively reduce the complexity of the SC-List based decoders. In addition, fast series of SC and SCL decoding are introduced to increase the throughput of polar codes [20]–[22].

In order to improve the frame error rate (FER) performance of polar codes with low complexity, a new flavor of the SC decoding algorithm called the successive cancellation flip (SC-Flip/SCF) decoder is described in [23], which can identify and correct the wrong decisions made by SC. By correcting the first channel-induced erroneous decision of SC, the SC-Flip decoding can compete with the SCL decoder for $L = 2$ in terms of error performance while keeping the computational complexity equivalent to that of SC at medium to high signal-to-noise ratio (SNR) values. Modifications of the SC-Flip decoder are introduced in [24]–[31]. A set of indices is created from Rate-1 nodes in the decoding tree, used progressively to correct multiple errors with a high number of iterations [24], [25]. A generalization of the SC-Flip decoder [26], [27] is proposed, where the authors discuss the higher order bit-flipping and design a new scaling metric to improve the ability of SC-Flip to find the first error that occurred during the initial SC decoding. Through analyzing the values of logarithmic likelihood (LLR), an improved bit-flipping algorithm [28] uses a simulation-based threshold to select the flipping bits and improves the performance of the SC-Flip decoder. In addition, the state-of-the-art high-speed SC decoding algorithm is incorporated to fit the bit-flipping algorithm [29] and it is shown that the Fast-SSC-Flip decoder has a decoding speed close to an order of magnitude better than the traditional ones while retaining a comparable error-correction performance. Besides, BP bit-flip decoder for polar codes is designed to perform bit-flip in a polar BP decoder [30]. In [31], partitioned version of the SC-Flip (PSCF) algorithm is provided to correct multiple errors, bringing improvements of error-correction performance and computational complexity when they match each other. However, the CRC bits used in PSCF only check the information bits. In fact, the most critical part of non-frozen bits should be specially protected by CRC. Accordingly, it is necessary to sort and select the critical ones from the segments in the decoding process. In addition, the degradation of performance caused by distributed CRC can be fully considered in designing the segmented scheme. Thus, based on the above problems, a generalized segmented bit-flipping scheme should be carefully considered.

As a generalized version of the segmented bit-flipping scheme, segmented decoding patterns aiming at single-error-correcting (SEC) and multiple-error-correcting (MEC) are first described (named SDP-S and SDP-M respectively). Based on the above basic patterns, the two enhanced version of them (named ESDP-S and ESDP-M) that are focused on in our work is proposed to further improve the performance of segmented decoding patterns. For ESDP series, distributed CRCs are designed to protect the critical ones of non-frozen bits. Accordingly, one just needs to identify and sort the potential erroneous bits in the protected parts of segments. On the other hand, the locations of CRC bits and error probability of each non-frozen bit are analyzed together for a better segmented scheme. Indeed, CRC bits can degrade the error-correction performance of polar codes since more non-frozen bits can be introduced. Hence, it is necessary to use more reliable subchannels to transmit CRC bits in segments. Besides, generally speaking, the ability for CRC to detect errors in the codeword improves with the length of CRC bits. Thus, the size of distributed CRC bits should depend on the error probability of each non-frozen bit in a segment.

The main work and contributions of this paper can be summarized as follows:

A generalized segmentation framework for bit-flipping algorithm is described, in which the performance of four decoding patterns is analyzed and discussed.

To improve the FER performance of basic segmented decoding algorithms, improved version aiming to protect the critical parts of segments is proposed. It is shown that the proposed scheme can efficiently improve the FER performance of the segmented algorithms.

A new strategy indicating the positions of segments is provided, in which the locations of CRC bits and the error probability of each bit are considered together. It is shown that segmented decoding patterns using the segmented scheme can perform well at different code rates.

To show the use of different segmented decoding patterns, the FER performance of them is analyzed and discussed at matching average computational complexity for different rates of polar codes. As well as, the average computational complexity is calculated under equivalent FER for different rates of polar codes.

The remainder of this paper is organized as follows. In section II, the basic knowledge of polar codes is described. The basic and improved segmented decoding patterns for polar codes as well as a new strategy indicating the positions of segments are described in section III. In section IV, the simulation results are given, and are followed by the conclusions in section V.

II. PRELIMINARIES

A. POLAR CODES

Polar codes, derived from the phenomenon of channel polarization, can achieve the capacity of binary-input discrete memoryless channel (B-DMC). Channel polarization is an operation by which one manufactures N synthetic channels from N independent copies of a given B-DMC with the help of channel combining and splitting. It is assumed that $W : X \to Y$ denotes an arbitrary B-DMC, where X and Y denote the input and output alphabet of the channel, respectively. The transition probability of B-DMC is defined as $W(y|x)$, where $x \in X = \{0, 1\}$ and $y \in Y$. In this way, the capacity $I(W)$ of symmetric B-DMC can be expressed as

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

By using channel combining and splitting operations on N independent uses of W, the N successive uses of synthesized binary channels $W_N^{(i)}$ can be obtained. Accordingly, the channel transition probability is given as

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

where $W_N(y_1^N|u_1^N) = W(y_1|x_1)W(y_2|x_2)\cdots W(y_N|x_N)$, and $u_1^N = (u_1, u_2, \cdots, u_N)$ denotes the input vector of the synthesized channels. As a result, the synthesized channels are polarized in the sense that, as N goes to infinity through powers of two, some of synthesized channels become noiseless, while others become noisy. In practice, good subchannels can be used to transmit information.

Taking advantage of the polarization effect, polar codes can be constructed. A polar code is a linear block code characterized by a three-tuple $(N, K, \mathcal{A})$, where $N = 2^n$ denotes the code length, K denotes the information bits length, and $\mathcal{A}$ is the information set indicating the positions of the K information bits, corresponding to good subchannels. Bits whose positions belong to the set $\mathcal{A}^c$ are called frozen bits, usually set to zero, where $\mathcal{A}^c$ is the complementary set of $\mathcal{A}$. A polar code construction aims to choose the set $\mathcal{A}$ or $\mathcal{A}^c$ through channel parameters such as the Bhattacharyya parameter for binary erasure channel (BEC). When the encoding bit sequence u_1^N is generated, the encoded bit sequence under polar coding can be calculated through the following matrix multiplication:

$$x_1^N = u_1^N B_n F^{\otimes n} \tag{3}$$

where $n = \log_2 N$, B_n is a permutation matrix, $F^{\otimes n} = F \otimes F^{\otimes(n-1)}$ with $F = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$, and $\otimes n$ denotes n order Kronecker power. The encoding process for polar code $(8, 4)$ is depicted in FIGURE 1, where black indices represent information bits and gray indices represent frozen bits. It can be seen that an N-length polar code can be encoded from two polar codes of length $N/2$.

B. SUCCESSIVE-CANCELLATION DECODING

The SC decoding algorithm for polar codes can be seen as a depth-first binary tree search. As shown in FIGURE 2, black leaf nodes represent information bits and white leaf nodes represent frozen bits. In the decoding tree, message passing operations are executed, in which a local decoder at node v receives a soft information vector α_v and is responsible

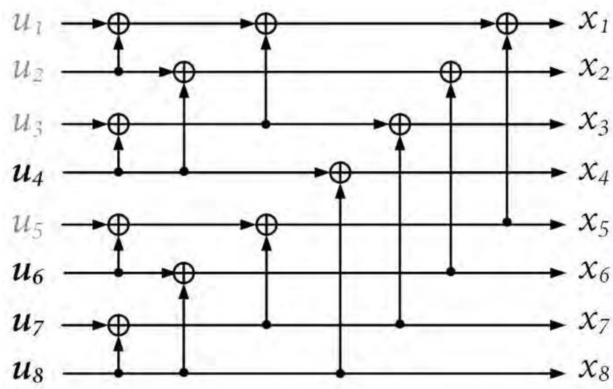


FIGURE 1. The encoding process for polar code (8, 4).

for producing a codeword β_v. At the receiver, the received vector y_1^N from the channel is the input of the decoder and the initial LLR can be defined by $\lambda_i = \log(\frac{Pr(y_i|x_i=0)}{Pr(y_i|x_i=1)})$. The root node initializes its soft information vector by $\alpha_{v_{root}} = (\lambda_1, \lambda_2, \cdots, \lambda_N)$. For an activated internal node v, its left node is activated and the soft information α_{v_l} is calculated through

$$\alpha_{v_l}[i] = \alpha_v[2i-1] \boxplus \alpha_v[2i] \text{ for } i = 1, \cdots, 2^{n-d_v-1} \tag{4}$$

where d_v is the depth of the node v. For real numbers x and y, the binary operator $\boxplus$ is defined as $x \boxplus y = 2\tanh^{-1}(\tanh(\frac{x}{2})\tanh(\frac{y}{2}))$. After receiving β_{v_l} from v_l, node v activates its right node v_r and calculates α_{v_r} through

$$\alpha_{v_r}[i] = \alpha_v[2i-1](1 - 2\beta_{v_l}[i]) + \alpha_v[2i] \quad \text{for } i = 1, \cdots, 2^{n-d_v-1}. \tag{5}$$

The node v then receives β_{v_r} from v_r and calculates β_v via

$$\beta_v[2i] = \beta_{v_r}[i] \text{ and } \beta_v[2i-1] = \beta_{v_l}[i] \oplus \beta_{v_r}[i] \quad \text{for } i = 1, \cdots, 2^{n-d_v-1}. \tag{6}$$

The message passing process of current node finishes once node v gets β_v. For the following nodes, the same operations are performed until the leaf node of the decoding tree. Last, leaf node decoders perform the following decision:

$$\hat{u}_i = \begin{cases} 0, & \text{if } \alpha_i \geq 0 \text{ and } i \in \mathcal{A}; \\ 1, & \text{if } \alpha_i < 0 \text{ and } i \in \mathcal{A}; \\ u_i, & \text{if } i \in \mathcal{A}^c; \end{cases} \tag{7}$$

C. SUCCESSIVE-CANCELLATION FLIP DECODING

It was observed that erroneous bit decisions in the SC decoding can be caused by channel noise or by error propagation due to previous erroneous bit decisions. Besides, the first erroneous decision is always caused by the channel noise since there are no previous errors, and error propagation does not affect the FER of polar codes. Thus, it is possible to improve the FER of the SC decoding by correcting the first erroneous decision. Based on the above analysis,

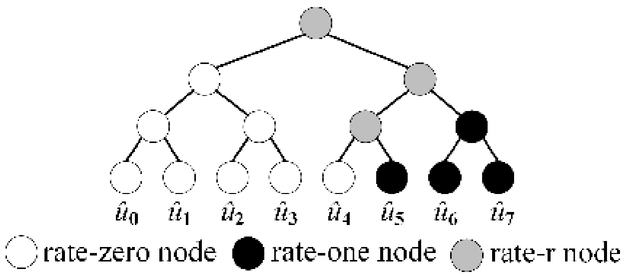


FIGURE 2. SC decoding for polar code (8, 3).

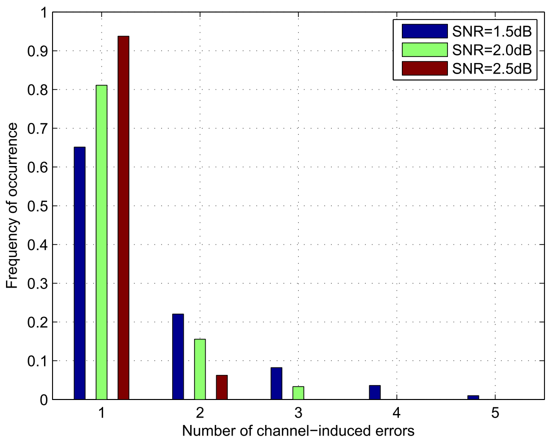


FIGURE 3. Frequency of occurrence of channel-induced errors at various SNR points for PC (1024, 512).

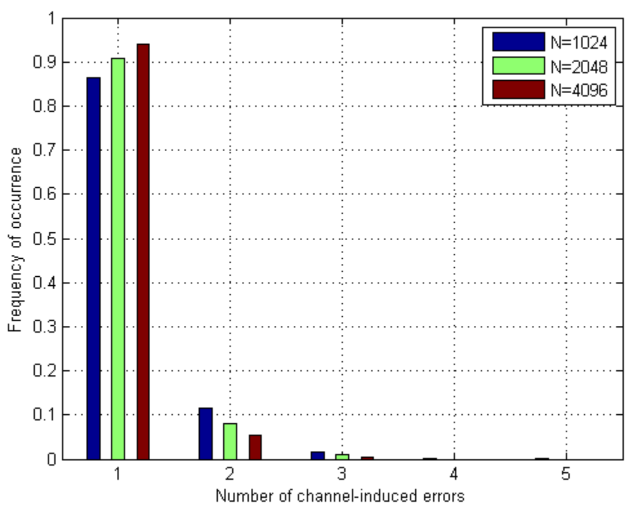


FIGURE 4. Frequency of occurrence of channel-induced errors at three different code lengths.

the frequency of occurrence for channel-induced errors is depicted in FIGURE 3 and FIGURE 4. It is shown that most of the decoding failures are due to a single channel-induced error, whose frequency increases with SNR and code lengths. In our work, we denote by Ω1 the occurrence of a single channel-induced error. To evaluate the impact of Ω1 event, an Oracle-SC decoder is designed, in which the first channel-induced error can be identified and it is ensured that SC can estimate the bit correctly. Consequently, the FER under the Oracle-SC decoder can be regarded as a lower bound of the SC-Flip decoder. The aim of SC-Flip decoding is to identify and correct the first erroneous decision that occurs during SC decoding without the aid of an oracle. To this end, an C-bit CRC is incorporated to indicate whether the estimated vector $\hat{u}_1^N$ is valid or not. In the SC-Flip decoding algorithm, a standard SC decoding is first performed to produce an estimated vector $\hat{u}_1^N$. If $\hat{u}_1^N$ passes the CRC, the decoding is completed. If the CRC fails, T_{max} SC decoding attempts are initialized in order to identify the first channel-induced error in the SC decoding. At each attempt, one of the selected bits is flipped. The decoding process terminates when the estimated $\hat{u}_1^N$ passes the CRC or when all T_{max} attempts have failed.

III. DESCRIPTION OF SEGMENTED SCHEME

In this section, the basic segmented decoding patterns are described. Then, the enhanced segmented decoding patterns using critical information bits are proposed. Last, a new strategy of choosing positions of segments is given.

A. THE DESCRIPTION OF BASIC SEGMENTED DECODING PATTERNS

The CRC encoding and checking for single and segmented schemes are depicted in FIGURE 5. It is assumed that a polar code with K information bits (info bits) and C CRC bits is used, in which S segments are considered. For the traditional SC-Flip decoding, it is necessary that the CRC bits are calculated from the whole information bits in the process of CRC encoding and checked through the whole decoding bits in the process of CRC checking. However, for the segmented idea, distributed CRC bits are used to independently protect its

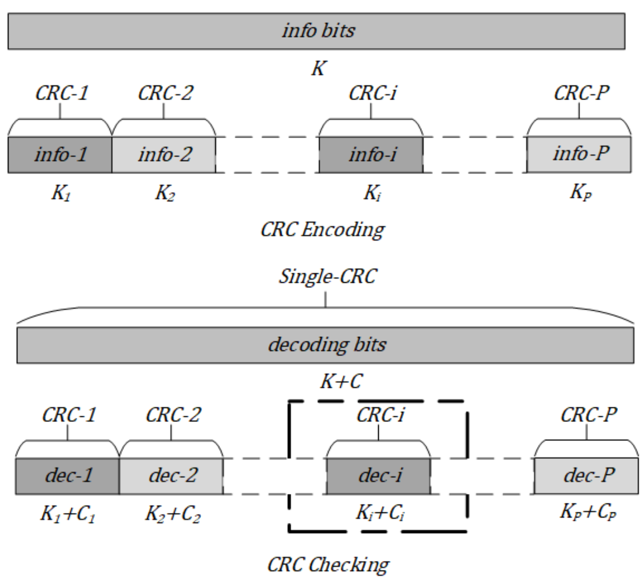


FIGURE 5. CRC Encoding and Checking for single-CRC scheme and basic segmented scheme.

own partial information bits. Accordingly, distributed CRC should be calculated via its protected decoding bits. Aiming at the first channel-induced error, standard SC-Flip decoding has up to T_{max} chances to flip a candidate bit at a time by observing the $K + C$ unfrozen bits. For the segmented scheme, if the first channel-induced error is in the i_{th} segment, CRC-i check will fail with great probability. In this way, one has up to T_{max} chances to flip a candidate bit at a time by observing the $K_i + C_i$ unfrozen bits. Thus, it is more possible to flip the first channel-induced erroneous bit since it is equivalent to an increase of T_{max}. Although a good aspect can be available, it should be noticed that it is possible that the length and positions of distributed CRC bits can degrade the performance of polar codes, which may offset the benefits. Hence, conditions in which segmented scheme is effective should be carefully considered.

Based on the above segmented scheme, a basic segmented decoding pattern aiming at SEC (SDP-S) can be described. If CRC-1 to CRC-$i-1$ passes while CRC-i fails, T_{max} decoding attempts is initialized by observing the LLR of the $K_i + C_i$ unfrozen bits in segment i. At each attempt, one of the unfrozen bits flips and SC decoding continues from the current index with CRC-i checking at the end of the segment. If CRC-i check passes by bit-flipping, the first channel-induced error is corrected and in the subsequent segments, only SC decoding is executed without CRC checking. If CRC-i check still fails under T_{max} decoding attempts, the decoding will terminate. Thus, the calculations of the subsequent segments can be avoided. To better understand the SDP-S decoding pattern, the algorithm is described in Algorithm 1. The input $\gamma = \{\gamma_i | 0 \le i \le S, i \in Z\}$ is a segment vector, where γ_i denotes the last index of each segment except for $\gamma_0 = -1$. The unfrozen-bit indices set $\tilde{\mathcal{A}} = \mathcal{A} \cup \mathcal{A}_{crc}$ indicates the positions of information bits and CRC bits, where the CRC bit indices set $\mathcal{A}_{crc}$ should choose the most C_i reliable positions after K_i information bit positions in each segment. Accordingly, $\gamma^{\tilde{\mathcal{A}}}$ denotes the mapping set of γ under unfrozen-bit indices set $\tilde{\mathcal{A}}$. For example, for a polar code (32, 16) with $S = 2$ and CRC-4, it is possible for $\gamma = \{-1, 15, 31\}$ and $\gamma^{\tilde{\mathcal{A}}} = \{-1, 9, 19\}$. The construction of γ will be discussed later. Besides, the function SORT sorts the absolute values of LLR in ascending order and the function INDEX selects the top T_{max} indices. It should be noted that the SDP-S can only correct the first channel-induced error and it will be enhanced later.

It is also obvious that multiple errors can be detected in the segmented scheme since each segment has an independent CRC check. Thus, one can try to correct one erroneous bit by executing T_{max} decoding attempts for every CRC-failed segment. In this way, the corrected errors can reach up to the number of segments. It should be noted that multiple errors in one segment cannot be corrected anyway. Based on the above analysis, a basic segmented decoding pattern that can correct multiple errors (SDP-M) is described in algorithm 2. The information set $\tilde{\mathcal{A}}$ and the segment vector γ are known by the decoder. For each segment, SC decoding is executed first and then CRC check is used. If CRC fails, the bit-flipping operations will be implemented.

Algorithm 1 SDP-S Algorithm

Require: y_1^N, T_{max}, $\tilde{\mathcal{A}}$, S, γ
Ensure: $\hat{u}_1^N$

$j = 1$, *flip*=TRUE
while $j \le S$ **do**
  $(\hat{u}_{\gamma_{j-1}^{\tilde{\mathcal{A}}}+1}^{\gamma_j^{\tilde{\mathcal{A}}}}, \alpha_{\gamma_{j-1}^{\tilde{\mathcal{A}}}+1}^{\gamma_j^{\tilde{\mathcal{A}}}}) \leftarrow \mathrm{SC}(y_1^N, \tilde{\mathcal{A}})$
  if $T_{max} > 1$ and *flip* =TRUE **then**
    $flag_{crc}$=CRC$(\hat{u}_{\gamma_{j-1}^{\tilde{\mathcal{A}}}+1}^{\gamma_j^{\tilde{\mathcal{A}}}})$
    if $flag_{crc}$ ==FALSE **then**
      $\alpha^s \leftarrow \mathrm{SORT}(|\alpha_{\gamma_{j-1}^{\tilde{\mathcal{A}}}+1}^{\gamma_j^{\tilde{\mathcal{A}}}}|)$
      $U^s \leftarrow \mathrm{INDEX}(\alpha^s, T_{max})$
      $t = 0$
      while $t < T_{max}$ **do**
        $\hat{u}_{\gamma_{j-1}^{\tilde{\mathcal{A}}}+1}^{\gamma_j^{\tilde{\mathcal{A}}}} \leftarrow \mathrm{SC}(y_1^N, \tilde{\mathcal{A}}, U^s[t])$
        if CRC$(\hat{u}_{\gamma_{j-1}^{\tilde{\mathcal{A}}}+1}^{\gamma_j^{\tilde{\mathcal{A}}}})$==TRUE **then**
          flip =FALSE
          break
        end if
        $t = t + 1$
      end while
      if $t == T_{max}$ **then**
        exit decoding
      end if
    end if
  end if
end while

B. THE DESIGN OF ENHANCED SEGMENTED DECODING PATTERNS

In the standard SC-Flip decoding, a single CRC is applied to protect the information bits, indicating whether the estimated bits are correct or not. If the CRC fails for SC decoding, the search space of bit-flipping is the information bits and CRC bits. In the basic segmented scheme, multiple CRCs are applied to protect the segmented information bits, in which the segments with the failed CRC check can implement the bit-flipping decoding. Thus, the search space of bit-flipping is limited to one segment during an initialized decoding attempt. Indeed, it is not necessary to protect the whole information bits in some cases since some of them are very reliable. One just needs to protect the critical ones that are very error-prone using CRC. Based on this idea, as shown in FIGURE 6, an enhanced segmented scheme is proposed, in which a critical set, which is a subset of the information

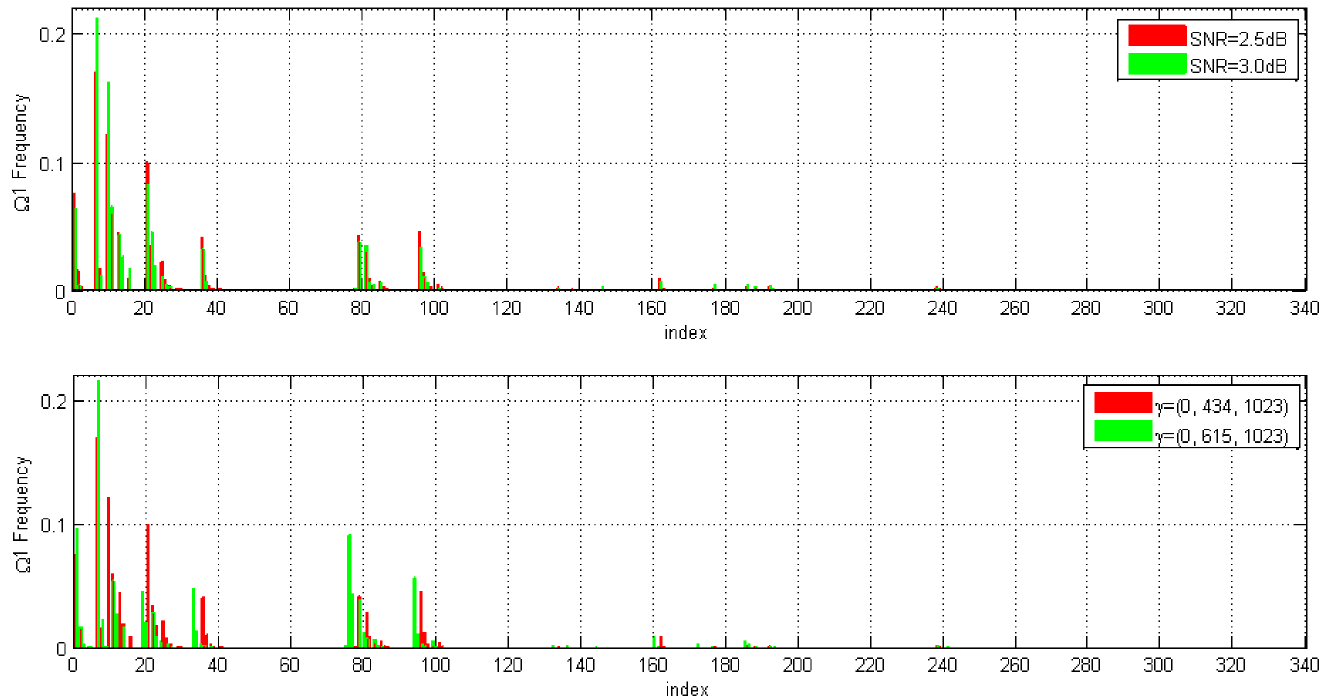


FIGURE 7. Frequency of Ω1 event occurrence for Polar Code (1024, 340) at different SNR points and different segment vector γ.

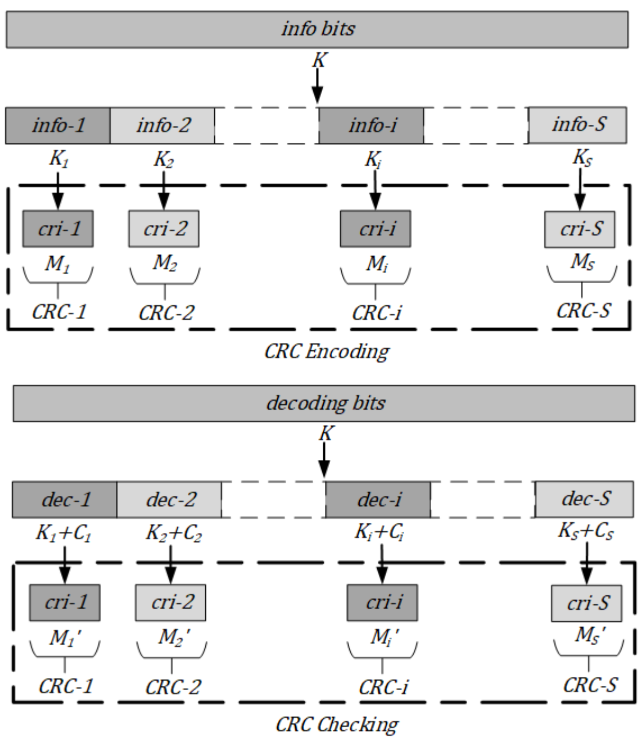


FIGURE 6. CRC encoding and checking for enhanced segmented decoding patterns.

bit set, is considered. For CRC encoding, information bits are divided into S parts, from which M_i information bits belonging to the critical set (called critical information bits) in every part are selected. Then, independent CRC is used to encode the critical information bits at each segment. For CRC checking, M_i' critical decoding bits from original decoding bits at each segment are checked by CRC to indicate the correctness of the corresponding segment. Accordingly, one just needs to select the flipping bits in the critical information bits in each segment. As a result, the protected bits and the corresponding search space of bit flipping is limited to the critical information bits for each segment. It is obviously equivalent to an increase of T_{max}. Compared to the basic segmented scheme, the CRC check can protect fewer bits, improving the effectiveness of the check. Besides, the unnecessary candidate flipping bits in the inner segment can be excluded efficiently. In this way, the erroneous bits can efficiently flip for a limited number of decoding attempts.

To apply the enhanced segmented scheme, an appropriate critical set is of significance. One of the ways to construct the critical set is to use monte carlo method with the help of Oracle-SC decoder, in which the error probability of each unfrozen bit is estimated by the frequency of errors. More specifically, the probability of Ω1 event for each unfrozen bit is calculated for approximation. FIGURE 7 depicts the frequency of this event occurrence over the unfrozen bit indices at different SNR points and different segment vector γ, obtained by simulating 500000 frames for polar code (1024, 340) with CRC-16 and $S = 2$. It can be seen that only a few of them have a high frequency of the first channel-induced error event occurrence and the majority of them have very low frequency or do not incur error.

Algorithm 2 SDP-M Algorithm

Require: y_1^N, T_{max}, $\tilde{\mathcal{A}}$, S, γ
Ensure: $\hat{u}_1^N$

$j = 1$
while $j \leq S$ **do**
  $(\hat{u}_{\gamma_{j-1}^{\tilde{\mathcal{A}}}+1}^{\gamma_j^{\tilde{\mathcal{A}}}}, \alpha_{\gamma_{j-1}^{\tilde{\mathcal{A}}}+1}^{\gamma_j^{\tilde{\mathcal{A}}}}) \leftarrow \mathrm{SC}(y_1^N, \tilde{\mathcal{A}})$
  if $T_{max} > 1$ **then**
    $flag_{crc}=\mathrm{CRC}(\hat{u}_{\gamma_{j-1}^{\tilde{\mathcal{A}}}+1}^{\gamma_j^{\tilde{\mathcal{A}}}})$
    if $flag_{crc}$ ==FALSE **then**
      $\alpha^s \leftarrow \mathrm{SORT}(|\alpha_{\gamma_{j-1}^{\tilde{\mathcal{A}}}+1}^{\gamma_j^{\tilde{\mathcal{A}}}}|)$
      $U^s \leftarrow \mathrm{INDEX}(\alpha^s, T_{max})$
      $t = 0$
      while $t < T_{max}$ **do**
        $\hat{u}_{\gamma_{j-1}^{\tilde{\mathcal{A}}}+1}^{\gamma_j^{\tilde{\mathcal{A}}}} \leftarrow \mathrm{SC}(y_1^N, \tilde{\mathcal{A}}, U^s[t])$
        if $\mathrm{CRC}(\hat{u}_{\gamma_{j-1}^{\tilde{\mathcal{A}}}+1}^{\gamma_j^{\tilde{\mathcal{A}}}})$==TRUE **then**
          break
        end if
        $t = t + 1$
      end while
      if $t == T_{max}$ **then**
        exit decoding
      end if
    end if
  end if
end while

Thus, those unfrozen bits having high probability of the first channel-induced error event occurrence can be seen as the elements of the critical set. For the selected critical indices, the sum of the frequency of the first channel-induced error event occurrence should large enough depending on the target FER. It is noted that the critical set also depends on the construction of polar codes, the required SNR and the segment vector. To better understand the construction of the critical set, the following steps are described.

Step 1 For a polar code $(N, K, \mathcal{A})$, the corresponding segment vector γ is initialized, followed by the position assignment of CRC bits at each segment.

Step 2 Under a target number of frames, random information bits are generated and distributed CRC bits are calculated.

Step 3 When receiving the codewords from the AWGN channel, the Oracle-SC algorithm is applied to decode them by identifying and recording the indices of the first channel-induced error.

Step 4 Based on the number of the first channel-induced error of each index, the error probability of each unfrozen bit is estimated by $e_i^{ufrz} = \frac{E_i}{\sum_i E_i}$, where E_i is the number of the first channel-induced error for index i and $1 \leq i \leq \sum_{i=1}^{S}(K_i + C_i)$.

Algorithm 3 ESDP-M Algorithm

Require: y_1^N, T_{max}, $\mathcal{C}$, $\tilde{\mathcal{A}}$, γ
Ensure: $\hat{u}_1^N$

1: $j = 1$
2: **while** $j \leq S$ **do**
3:   $(\hat{u}_{\gamma_{j-1}^{\tilde{\mathcal{A}}}+1}^{\gamma_j^{\tilde{\mathcal{A}}}}, \alpha_{\gamma_{j-1}^{\tilde{\mathcal{A}}}+1}^{\gamma_j^{\tilde{\mathcal{A}}}}) \leftarrow \mathrm{SC}(y_1^N, \tilde{\mathcal{A}})$
4:   **if** $T_{max} > 1$ and $\mathrm{CRC}(\hat{u}_{\gamma_{j-1}^{\tilde{\mathcal{A}}}+1}^{\gamma_j^{\tilde{\mathcal{A}}}}, \mathcal{C}) ==$ FALSE **then**
5:     $\alpha^s \leftarrow \mathrm{SORT}(|\alpha_{\gamma_{j-1}^{\tilde{\mathcal{A}}}+1}^{\gamma_j^{\tilde{\mathcal{A}}}}|, \mathcal{C})$
6:     $U^s \leftarrow \mathrm{INDEX}(\alpha^s, T_{max})$
7:     $t = 0$
8:     **while** $t < T_{max}$ and $\mathrm{CRC}(\hat{u}_{\gamma_{j-1}^{\tilde{\mathcal{A}}}+1}^{\gamma_j^{\tilde{\mathcal{A}}}}, \mathcal{C}) ==$ FALSE **do**
9:       $\hat{u}_{\gamma_{j-1}^{\tilde{\mathcal{A}}}+1}^{\gamma_j^{\tilde{\mathcal{A}}}} \leftarrow \mathrm{SC}(y_1^N, \tilde{\mathcal{A}}, U^s[t])$
10:       $t = t + 1$
11:     **end while**
12:     **if** $\mathrm{CRC}(\hat{u}_{\gamma_{j-1}^{\tilde{\mathcal{A}}}+1}^{\gamma_j^{\tilde{\mathcal{A}}}}, \mathcal{C}) ==$ FALSE **then**
13:       exit decoding
14:     **end if**
15:   **end if**
16: **end while**

Step 5 After get the sorted version e_i^{sufrz} of the error probability e_i^{ufrz} in descending order, select the first M error probability that meets $\sum_{i=0}^{M-1} e_i^{sufrz} > \zeta$, where the value of ζ depends on the number of target frames and $\zeta \approx 1$.

Step 6 The indices included in the first M error probability of e_i^{sufrz} are the elements of the critical set $\mathcal{C}$.

Based on the above analysis, enhanced version of the basic segmented decoding patterns is proposed named ESDP-S and ESDP-M. For the basic version, distributed CRCs need to protect the information bits and execute the sorting and selection operations in them, which brings about the degraded error-correction performance of polar codes provided that a limited number of decoding attempts is used. Thus, the enhanced version aims to protect the critical information bits and flip them in the decoding process. For a better understanding of the decoding patterns, the decoding process of ESDP-M is described in Algorithm 3. The set $\mathcal{C}$ containing the critical information bit indices, the set $\tilde{\mathcal{A}}$ and the segment vector γ are determined in advance and acquired at the receiver. At each segment, the standard SC algorithm is implemented first (line 3). Then, the CRC reminder is calculated by the estimated critical information bits (line 4). Once the distributed CRC check fails for the standard SC decoding, the indices of the T_{max} critical information bits with the smallest absolute value of LLRs are identified (lines $4 - 6$). Then, for a maximum number of decoding attempts T_{max}, the SC algorithm is performed for each attempt, where one of the critical information bits corresponding to the identified

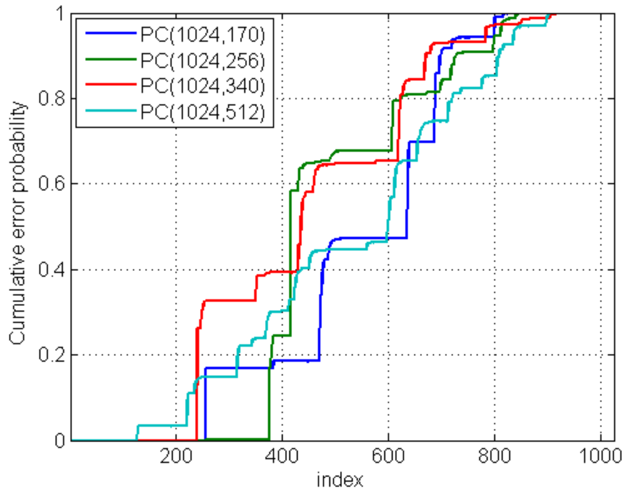


FIGURE 8. **Cumulative error probability of Ω1 event of information bits for different rates of polar codes.**

indices is flipped, until the CRC check passes or the maximum number of decoding attempts is reached (lines $7-11$). If the CRC check for the critical information bits still fails after T_{max} iterations, the decoding process is terminated(lines $12-14$). It should be noticed that the basic segmented decoding patterns is the special case of the enhanced segmented decoding patterns when critical set $\mathcal{C}$ is equal to the information set $\mathcal{A}$.

C. THE STRATEGY OF CHOOSING SEGMENT POSITION

One of the important topics of the segmented scheme is to decide the position of each segment, which can affect the performance of polar codes. In our work, the error probability of information bits and the effect of positions of distributed CRC bits are both considered. It is assumed that C_i CRC bits are assigned in i_{th} segment and the error probability of $\Omega 1$ event for information bits is denoted as $\{e_i^{\mathcal{A}}, 1 \leq i \leq K\}$, which can be calculated with the help of the Oracle-SC decoder. In general, the longer the CRC length, the more efficient the error detection ability. Thus, the i_{th} segment should meet

$$\frac{C_i}{C} = \sum_{j=\gamma_{i-1}^{\mathcal{A}}+1}^{\gamma_i^{\mathcal{A}}} e_j^{\mathcal{A}} \tag{8}$$

where $\gamma^{\mathcal{A}}$ is the segment vector in information bits set $\mathcal{A}$ corresponding to the segment vector γ. When the CRC length of each segment is equal, it should have an equal error probability over the CRC protected information bits. On the other hand, the most reliable C_i positions except for K_i information bit positions in each segment are used to put CRC bits. As a result, the corresponding $\sum_{i=1}^{S} C_i$ positions are very probably not the most reliable ones without considering information bit positions in N positions. That is, they will be degraded ones from the point of whole subchannels if not adjusted. Based on this, it is necessary to put the CRC bits in the more reliable positions while keeping each segment has an acceptable approximate error probability as soon as possible.

To implement the scheme, the cumulative error probability f_i of $\Omega 1$ event should be calculated, as shown in FIGURE 8, according to

$$f_i = \sum_{j=0}^{i} e_j \tag{9}$$

$$e_j = \begin{cases} e_{j'}^{\mathcal{A}}, & j \in \mathcal{A}; \\ 0, & j \in \mathcal{A}^c; \end{cases} \tag{10}$$

where j' is the index counting from 1 to K for information bits corresponding to j. A preliminary segment vector γ^{pre} is determined through the cumulative error probability, in which (8) should be satisfied. Then, γ^{pre} is adjusted by a factor ϵ, in which the more reliable positions of each segment are located by changing the last index of each segment while keeping that the changed range of the error probability is not greater than ϵ. To better understand the strategy of choosing the segment position, the following steps are described.

Step 1 For a polar code $(N, K, \mathcal{A})$, the error probability of $\Omega 1$ event for information bits is calculated as $\{e_i^{\mathcal{A}}, 1 \leq i \leq K\}$ with the help of the Oracle-SC decoder.

Step 2 The corresponding cumulative error probability $\{f_i, 1 \leq i \leq N\}$ is calculated based on $e_i^{\mathcal{A}}$ and $\mathcal{A}$.

Step 3 With a segment number S, the preliminary segment vector γ^{pre} is determined based on the cumulative error probability f_i and the number of CRC bits of each segment.

Step 4 Using a factor ϵ, γ^{pre} is adjusted by changing the last index of each segment while keeping $|\sum_{j=\gamma_{i-1}^{\mathcal{A}}+1}^{\gamma_i^{\mathcal{A}}} e_j^{\mathcal{A}} - \sum_{j=\gamma_{i-1}^{pre\mathcal{A}}+1}^{\gamma_i^{pre\mathcal{A}}} e_j^{\mathcal{A}}| < \epsilon$.

Step 5 For different values of ϵ, the adjusted γ that can achieve better FER can be seen as the segment vector.

Take polar code (1024, 256) with two segments as an example, the positions of CRC bits at different segment vector are shown in Fig.9, where the ''better positions'' mean that those positions that are more reliable than the most unreliable one of CRC bit positions. Considering the γ^{pre}, the segment line is 450, in which many better positions are not used. It is shown that the positions of CRC bits cannot be very reliable and may lead to a degraded performance. Considering the adjusted γ, the segment line is 600, where only one better position remains. Thus, the positions of CRC bits are very reliable, which can lead an improvement of performance for polar codes. Considering that the error probability of each information bit outside the critical set is very small, the strategy can be applied in the enhanced decoding patterns, in which the error probability of each critical information bit should be calculated. It should be considered that when $\epsilon = 0$, the strategy is just to divide codewords based on the error probability of each information bit, while the effect of positions of CRC bits is ignored. In addition, the choice of ϵ depends on the tradeoff between the error probability of each information bits and the effect of location of distributed CRC,

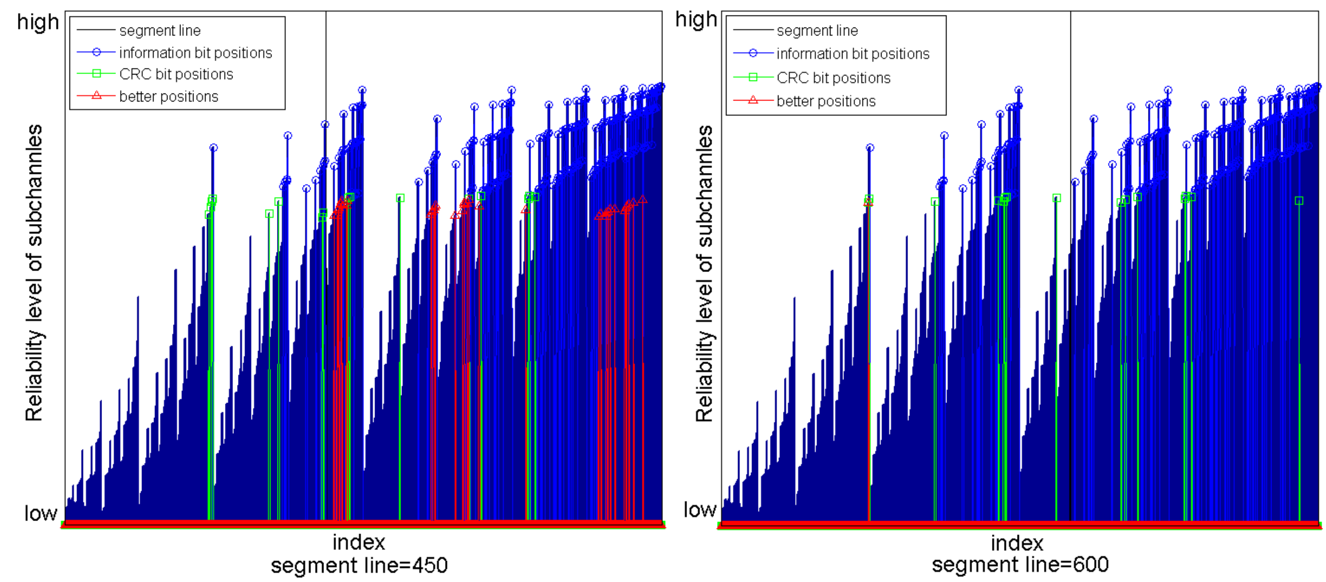


FIGURE 9. The positions of CRC bits at different segment vector.

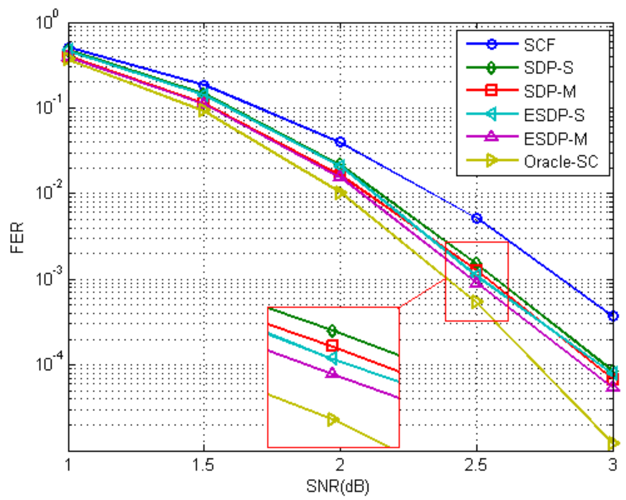


FIGURE 10. FER curves for SCF ($T_{max} = 9$), SDP-S, SDP-M, ESDP-S, ESDP-M and Oracle-SC decoders for polar code (1024, 340) by matching average computational complexity at SNR = 1.0dB.

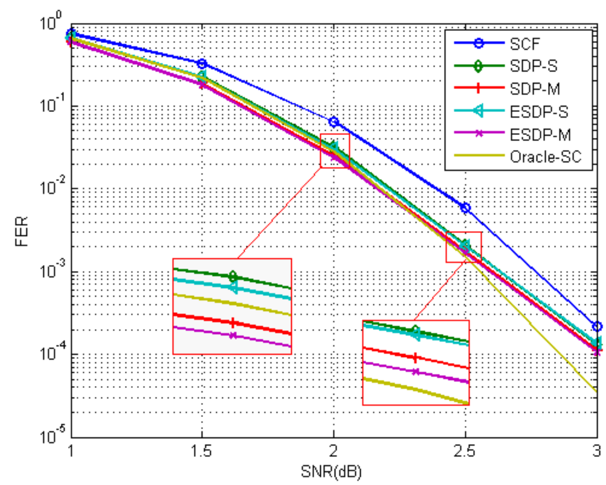


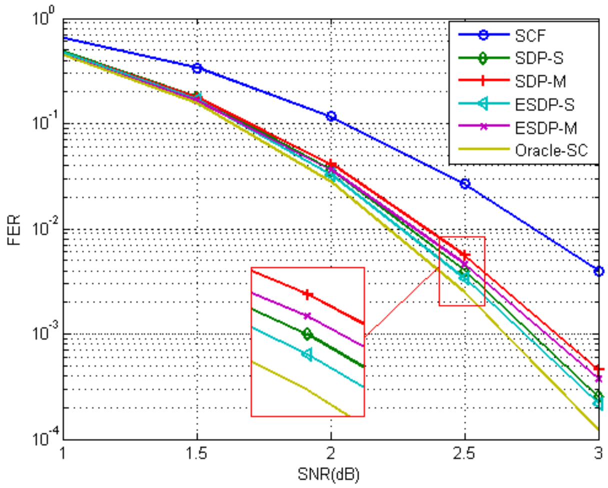
FIGURE 11. FER curves for SCF ($T_{max} = 8$), SDP-S, SDP-M, ESDP-S, ESDP-M and Oracle-SC decoders for polar code (1024, 512) by matching average computational complexity at SNR = 1.5dB.

which is difficult to decide in theory. Thus, it is recommended that the choice of ϵ can be determined according to the simulation results.

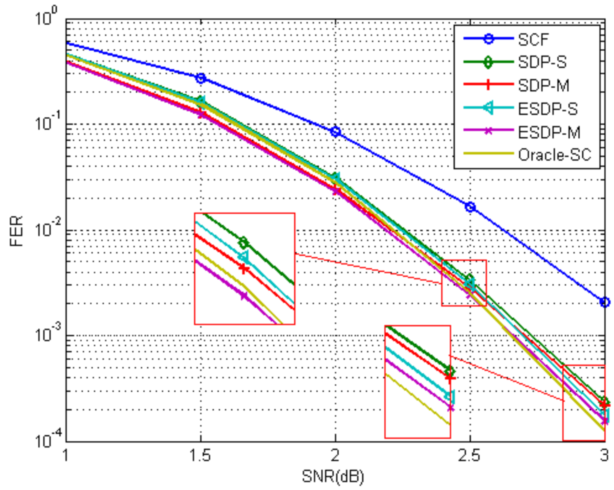
IV. SIMULATION RESULTS

In this section, the error-correction performance and the computational complexity of the proposed segmented decoding patterns are simulated and compared with that of SCF, Oracle-SC [23], and the PSCF decoders. Polar codes (1024, 170), (1024, 256), (1024, 340) and (1024, 512) are constructed targeting a SNR of 2.5dB under AWGN channel. Simulations have been run on the AWGN channel using binary phase shift keying (BPSK) modulation. The number of simulated frames is 5×10^5. For the SCF algorithm, a CRC check with $C = 16$ bits is used. For the segmented scheme, the information bits or the critical information bits are divided into two segments ($S = 2$), each protected by a CRC with 8 bits. It should be noted that while the bits allocated to the CRC are different, the positions of information bits are identical under the simulated algorithms. For different rates of polar codes, the critical set and the segment vector are constructed based on the proposed steps.

In order to effectively compare the error-correction performance of different decoders, FER curves at matching average computational complexity are illustrated. For polar code (1024, 340), T_{max} for SCF decoder is set to 9 and the



(a) $T_{max} = 4$



(b) $T_{max} = 10$

FIGURE 12. FER curves for SCF ($T_{max} = 4$ and $T_{max} = 10$), SDP-S, SDP-M, ESDP-S, ESDP-M and Oracle-SC decoders for polar code (1024, 256) by matching average computational complexity at SNR = 1.0dB. (a) $T_{max} = 4$. (b) $T_{max} = 10$

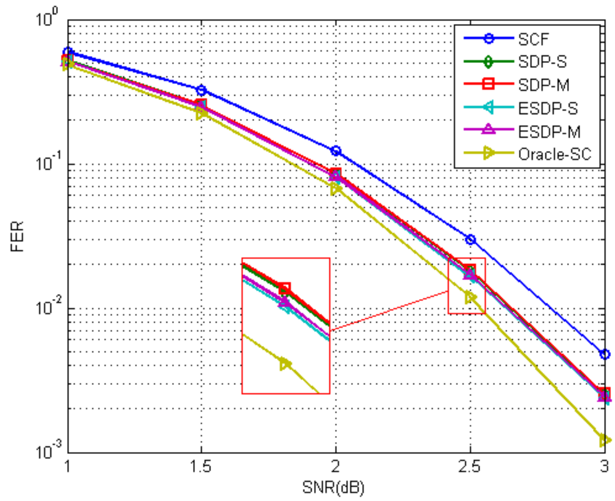


FIGURE 13. FER curves for SCF ($T_{max} = 12$), SDP-S, SDP-M, ESDP-S, ESDP-M and Oracle-SC decoders for polar code (1024, 170) by matching average computational complexity at SNR = 1.0dB.

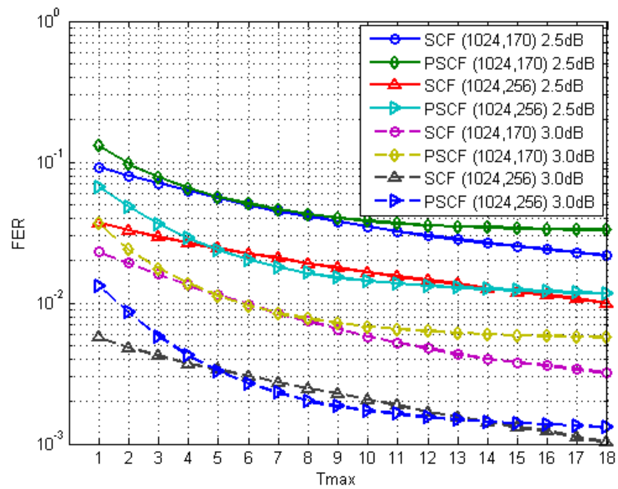
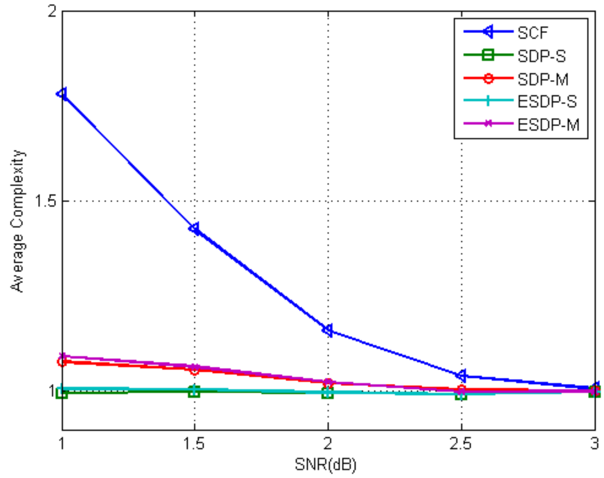
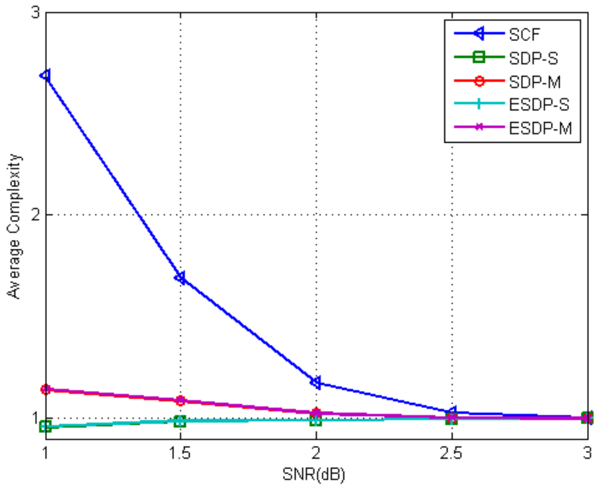


FIGURE 14. FER curves of SCF and PSCF with two segments for polar codes (1024, 170) and (1024, 256) at SNR = 2.0 and 2.5dB for different values of T_{max}.

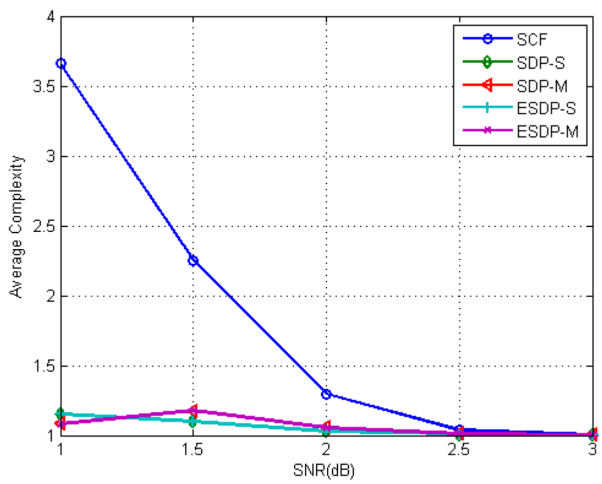
segment vector is $\gamma = \{-1, 434, 1023\}$ with $S = 2$. Besides, a total of 86 critical information bits are used for ESDP-S and ESDP-M. In addition, the CRC bits are considered in Oracle-SC decoder. The FER curves for SCF, SDP-S, SDP-M, ESDP-S, ESDP-M and Oracle-SC decoders for polar code (1024, 340) by matching average computational complexity at SNR = 1.0dB are depicted in FIGURE 10. It can be observed that ESDP-M approaches the Oracle-SC decoders at low SNR points and has better FER performance than other four decoders in all SNR points, followed by ESDP-S. This is because only the critical information bits are protected and flip in the decoding attempts in ESDP-S and ESDP-M and the ESDP-M can correct more high-order errors at low SNR points. As SNR grows, the FER curves of segmented decoding patterns deviate from the FER curve of Oracle-SC since the probability of multiple errors decrease in this region. In addition, the segmented decoding patterns can efficiently outperform the SCF decoders since it is equivalent to an increase of T_{max} for segmented ones while the ability of CRC check is enough. FIGURE 11 shows the FER curves of the above decoders for polar code (1024, 512), where the T_{max} of SCF is 8 and the complexity is matched at SNR = 1.5dB. It can be seen that the SDP-M and ESDP-M can outperform the Oracle-SC decoders at low SNR points such as SNR = 2.0dB. But with the increase of SNR, Oracle-SC outperforms them. The FER curves of SDP-M and ESDP-M are almost same at this code rate. It should be noted that the performance of PSCF is same as the SDP-M for polar codes (1024, 340) and (1024, 512) because the same segment vector is adopted. Thus, ESDP-S and ESDP-M outperform the traditional PSCF decoder for polar code (1024, 340) and ESDP-M has almost same FER performance as the PSCF for polar code (1024, 512).



(a) Polar codes $(1024, 256)$ with $T_{max} = 4$ for SCF



(b) Polar codes $(1024, 340)$ with $T_{max} = 9$ for SCF



(c) Polar codes $(1024, 512)$ with $T_{max} = 8$ for SCF

FIGURE 15. Average computational complexity of SCF, SDP-S, SDP-M, ESDP-S, ESDP-M for different rates of polar codes at matched FER. (a) Polar codes (1024; 256) with $T_{max} = 4$ for SCF. (b) Polar codes (1024; 340) with $T_{max} = 9$ for SCF. (c) Polar codes (1024; 512) with $T_{max} = 8$ for SCF.

The FER performance of low code rate of polar codes is considered. FIGURE 12 shows the FER curves for SCF, SDP-S, SDP-M, ESDP-S, ESDP-M and Oracle-SC decoders for polar code (1024, 256) by matching average computational complexity at SNR $=$ 1.0dB for different T_{max} of SCF. It is observed in FIGURE 12(a) that ESDP-S has better FER performance than SCF and other decoding patterns. This can be explained by the increase of single errors at low code rate of polar codes. In this way, the use of ESDP-M can lead to the increased complexity. By the way, decoding patterns using critical information bits contribute to the improvement of FER performance. Hence, the ESDP-S is recommended in this code rate. It is depicted in FIGURE 12(b) that when $T_{max} = 10$ for SCF, the FER curves of the segmented decoding patterns can approach or even outperform (for ESDP-M) the curve of the Oracle-SC. For polar code (1024, 170), as shown in FIGURE 13, the FER curves of ESDP-M cannot compete with Oracle-SC since the segment vector is adjusted a lot, which make ESDP-M almost lost the ability to correct multiple errors. However, the FER of the segmented decoding patterns are almost same and outperform the SCF decoder. Considering the calculation of CRC and critical set, the SDP-S is recommended at this code rate. In addition, the FER performance of PSCF decoders cannot compete with that of the basic SCF decoder at low code rates because the positions of CRC bits are very unreliable, which makes PSCF more vulnerable to errors than SC-Flip. This can be observed in FIGURE 14, where the FER performance of SCF and PSCF are compared under different T_{max}. Hence, all the segmented decoding patterns can outperform SCF and PSCF decoders at low code rate of polar codes.

It is assumed that an application of the SC algorithm over the N codeword bits is defined as the unit computational complexity and the average computational complexity is calculated at equivalent FER. Based on this, the average computational complexity of the SCF, SDP-S, SDP-M, ESDP-S and ESDP-M are depicted in FIGURE 15. It is shown that at low SNR points, the average computational complexity of the segmented decoding patterns is much lower than that of SCF decoder because they just need to continue decoding in one segment if CRC check fails. It is also demonstrated that the average computational complexity of the enhanced segmented decoding patterns is about a little higher than that of the basic ones since many critical information bits are found towards the beginning of the segments, resulting in almost a full iteration for each additional decoding attempt in that segment. At high SNR points, approximately above 2.5 dB, the average computational complexity of all algorithms converges to that of SC, where the complexity is one.

V. CONCLUSIONS

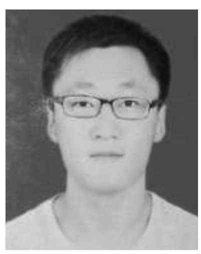
In this work, a generalized segmented bit-flipping scheme for SC decoding of polar codes is described and analyzed, in which the basic (SDP-S and SDP-M) and enhanced

(ESDP-S and ESDP-M) segmented decoding patterns are considered. At the same time, a new strategy of choosing segment positions is designed based on the error probability of information bits and the effect of location of distributed CRC bits. The FER performance at matching average computational complexity and the average computational complexity at equivalent FER are analyzed by simulations. It is demonstrated that the proposed segmented decoding patterns can almost outperform the SCF and PSCF decoder at all code rates and can approach or even outperform the Oracle-SC decoder in some code rates of polar codes while keeping a lower average computational complexity at equivalent FER in all SNR points.

REFERENCES

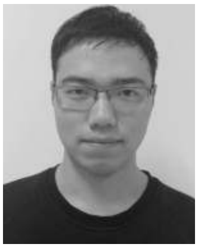
[1] E. Arikan, "Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels," *IEEE Trans. Inf. Theory*, vol. 55, no. 7, pp. 3051–3073, Jul. 2009.
[2] R. Mori and T. Tanaka, "Performance and construction of polar codes on symmetric binary-input memoryless channels," in *Proc. IEEE Int. Symp. Inf. Theory*, Jun./Jul. 2009, pp. 1496–1500.
[3] D. Zhou, K. Niu, and C. Dong, "Construction of polar codes in Rayleigh fading channel," *IEEE Commun. Lett.*, vol. 23, no. 3, pp. 402–405, Mar. 2019.
[4] K. Chen, K. Niu, and J. Lin, "Improved successive cancellation decoding of polar codes," *IEEE Trans. Commun.*, vol. 61, no. 8, pp. 3100–3107, Aug. 2013.
[5] K. Niu, K. Chen, J. Lin, and Q. T. Zhang, "Polar codes: Primary concepts and practical decoding algorithms," *IEEE Commun. Mag.*, vol. 52, no. 7, pp. 192–203, Jul. 2014.
[6] S. A. Hashemi, C. Condo, F. Ercan, and W. J. Gross, "On the performance of polar codes for 5G eMBB control channel," in *Proc. Asilomar Conf. Signals, Syst. Comput.*, Oct./Nov. 2017, pp. 1764–1768.
[7] F. Ercan, C. Condo, S. A. Hashemi, and W. J. Gross, "On error-correction performance and implementation of polar code list decoders for 5G," in *Proc. 55th Annu. Allerton Conf. Commun., Control, Comput. (Allerton)*, Oct. 2017, pp. 443–449.
[8] J. Dai, K. Niu, and J. Lin, "Polar-coded MIMO systems," *IEEE Trans. Veh. Technol.*, vol. 67, no. 7, pp. 6170–6184, Jul. 2018.
[9] A. Balatsoukas-Stimming, P. Giard, and A. Burg, "Comparison of polar decoders with existing low-density parity-check and turbo decoders," in *Proc. IEEE Wireless Commun. Netw. Conf. Workshops (WCNCW)*, Mar. 2017, pp. 1–6.
[10] I. Tal and A. Vardy, "List decoding of polar codes," *IEEE Trans. Inf. Theory*, vol. 61, no. 5, pp. 2213–2226, May 2015.
[11] A. Balatsoukas-Stimming, M. B. Parizi, and A. Burg, "LLR-based successive cancellation list decoding of polar codes," *IEEE Trans. Signal Process.*, vol. 63, no. 19, pp. 5165–5179, Oct. 2015.
[12] Z. Zhang, L. Zhang, X. Wang, C. Zhong, and H. V. Poor, "A split-reduced successive cancellation list decoder for polar codes," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 2, pp. 292–302, Feb. 2016.
[13] X. Liang, H. Zhou, Z. Wang, X. You, and C. Zhang, "Segmented successive cancellation list polar decoding with joint BCH-CRC codes," in *Proc. 51st Asilomar Conf. Signals, Syst., Comput.*, Oct./Nov. 2017, pp. 1509–1513.
[14] S. A. Hashemi, M. Mondelli, S. H. Hassani, R. Urbanke, and W. J. Gross, "Partitioned list decoding of polar codes: Analysis and improvement of finite length performance," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2017, pp. 1–7.
[15] K. Niu and K. Chen, "CRC-aided decoding of polar codes," *IEEE Commun. Lett.*, vol. 16, no. 10, pp. 1668–1671, Oct. 2012.
[16] 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.
[17] K. Niu and K. Chen, "Stack decoding of polar codes," *Electron. Lett.*, vol. 48, no. 12, pp. 695–697, Jun. 2012.
[18] 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, Dec. 2012.
[19] S. Li, L. Lu, Y. Deng, J. Liu, and T. Huang, "A reused-public-path successive cancellation list decoding for polar codes with CRC," *IEEE Commun. Lett.*, vol. 21, no. 12, pp. 2566–2569, Dec. 2017.
[20] 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.
[21] 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.
[22] 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, Nov. 2017.
[23] 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.
[24] Z. Zhang, K. Qin, L. Zhang, H. Zhang, and G. T. Cen, "Progressive bit-flipping decoding of polar codes over layered critical sets," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2017, pp. 1–6.
[25] Z. Zhang, K. Qin, L. Zhang, and G. T. Chen, "Progressive bit-flipping decoding of polar codes: A critical-set based tree search approach," *IEEE Access*, vol. 6, pp. 57738–57750, 2018.
[26] 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.
[27] 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.
[28] 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.
[29] P. Giard and A. Burg, "Fast-SSC-flip decoding of polar codes," in *Proc. Wireless Commun. Netw. Conf. (WCNC)*, Barcelona, Spain, Apr. 2018, pp. 73–77.
[30] Y. Yu, Z. Pan, N. Liu, and X. You, "Belief propagation bit-flip decoder for polar codes," *IEEE Access*, vol. 7, pp. 10937–10946, 2018.
[31] 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.
[32] D. Wu, Y. Li, and Y. Sun, "Construction and block error rate analysis of polar codes over AWGN channel based on Gaussian approximation," *IEEE Commun. Lett.*, vol. 18, no. 7, pp. 1099–1102, Jul. 2014.
[33] I. Tal and A. Vardy, "How to construct polar codes," *IEEE Trans. Inf. Theory*, vol. 59, no. 10, pp. 6562–6582, Oct. 2013.

SHIBAO LI was born in 1978. He received the B.S. and M.S. degrees in computer science from the University of Petroleum, China, in 2002 and 2009, respectively. He is currently an Associate Professor with the China University of Petroleum, Qingdao, China. His research interests include indoor localization technology, the Internet of Things, wireless networks, and mobile computing.



YUNQIANG DENG was born in 1993. He is currently pursuing the master's degree with the China University of Petroleum, Qingdao, China. His research interest includes channel coding.

XUN GAO was born in 1996. He is currently pursuing the master's degree with the China University of Petroleum, Qingdao, China. His research interest includes channel coding.



HE LI was born in 1994. He is currently pursuing the master's degree with the China University of Petroleum, Qingdao, China. His research interests include natural language processing and mobile computing.

LIN GUO received the B.Eng. degree in communication engineering from the China University of Petroleum, in 2018, where she is currently pursuing the M.E. degree with the Department of Electronics and Communication Engineering. Her current research interest includes interference alignment.

ZHENWEI DONG was born in 1997. He is currently pursuing the master's degree with the China University of Petroleum, Qingdao, China. His research interest includes channel coding.