IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received June 10, 2019, accepted June 27, 2019, date of publication July 8, 2019, date of current version August 7, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2927391

Multi-CRC Polar Codes and M-SCFlip-Based Decoding

RUI GUO[1], KANGNI CHEN[1], AND HUAPING LIU[2], (Senior Member, IEEE)
[1]School of Communication Engineering, Hangzhou Dianzi University, Hangzhou 310018, China
[2]Electrical Engineering and Computer Science, Oregon State University, Corvallis, OR 97331, USA

Corresponding author: Rui Guo (guorui@hdu.edu.cn)

This work was supported in part by the Zhejiang Provincial Natural Science Foundation of China under Grant LY16F010013, and in part by the National Natural Science Foundation Joint Research Program of China under Project U1709220.

ABSTRACT A multi-cyclic redundancy check (Multi-CRC) polar code construction algorithm is proposed in this paper to solve the error propagation problem of successive cancellation decoding for polar codes. In this algorithm, the information sequence is optimized into several segments to allow decoding errors to be corrected in time, minimizing the impact of error propagation. An improved multi-successive cancellation bit flipping (M-SCFlip) decoding algorithm is proposed to execute the bit flipping operation after CRC check-in each segment. In the low-SNR region, the proposed new multi-CRC polar code with successive cancellation list (SCL) decoding has a slight frame-error rate (FER) degradation compared with the original CRC polar code. With the M-SCFlip decoding algorithm developed in this paper, it achieves a better FER performance compared with the CRC polar code with successive cancellation (SC) and SCL ($L=2$) decoding algorithms. In addition, it has a lower decoding delay and requires a lower memory space. For example, at a FER of 10^{-4} with the same code length and effective code rate, the proposed multi-CRC polar code with M-CFlip decoding achieves a 1.19 dB and 0.79 dB gains over existing CRC polar codes with the SC and SCL ($L=2$) decoding algorithms, respectively.

INDEX TERMS Polar codes, successive cancellation decoding, error propagation, multi-bit flipping.

I. INTRODUCTION

Polar code with fixed coding structure has been proved to approach the Shannon limit asymptotically. In the binary discrete memoryless channel (B-DMC), as the code length N tends to infinity, polar code with successive cancellation (SC) decoding could reach the Shannon limit [1]–[4]. However, for short-to-medium-length polar codes, SC decoding results in a poor error performance due to incomplete channel polarization [5], [6]. To solve this problem, successive cancellation list (SCL) and successive cancellation stack (SCS) decoding algorithms have been proposed [7]–[9]. Built upon the SC decoding algorithm, the SCL algorithm retains multiple decoding decision paths, each corresponding to a candidate decoding codeword. The path with the highest reliability would be the decoding result. By improving the local path search mode of the SC algorithm, the performance of SCL decoding improves significantly. The computational complexities of SCL and SCS are $O(LN\log N)$ and $O(DN\log N)$, respectively, where L is the list width and D is the stack depth. A cyclic redundancy check (CRC)-assisted SCL algorithm was proposed in [10], [11]. In this scheme, CRC is added to the decoding path decision based on the SCL algorithm. The CRC-SCL algorithm could obtain even better performance with short or medium code lengths than LDPC and Turbo codes. The above improved decoding algorithm solves the problem of local optimization of path search in SC decoding by selecting a decoding path from candidate paths by using a CRC criterion. However, as L increases, the decoding complexity of SCL increases dramatically. Afisadis *et al.* proposed a successive cancellation flip (SCFlip) decoding algorithm [12], in which CRC is used to determine whether to perform a bit flip operation or not at the end of the SC decoding process; if yes, it flips only one bit. Zhang *et al.* [13] proposed another SCflip decoding algorithm.

Due to the serialization in SC decoding, the current decoded bits will affect the decoding process of the following bits; that is, there exists error propagation, one of the main factors that degrades the decoding performance [14]. This paper aims to address the above problem by developing a

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

new codeword structure and proposing an improved decoding algorithm. Specifically, we propose a polar code structure with multiple CRCs, aiming to mitigate the error propagation effects. Additionally, since this multi-CRC structure results in more CRC opportunities, a multi-bit SCFlip algorithm, named M-SCFlip, fits particularly well this structure to execute multi-bit flipping for a significantly improved decoding performance.

II. PRILIMINARIES: POLAR ENCODING AND DECODING

A. POLAR CODE ENCODE

Polar code differs from other linear block codes mainly in its channel polarization process [15]. In polar codes, $N(N = 2^n)$ mutually independent B-DMC sub-channels are converted into mutually dependent sub-channels $W_N^{(i)} : X \to Y \times X^{i-1}$, $i = 1, 2, \cdots, N$, by channel combining and channel splitting, where X represents the channel input set, Y represents the channel output set, taking on the values $\{0, 1\}$, and $W_N^{(i)}$ is the mutually dependent sub-channels after channel splitting. In the encoding process, the reliability of each polarized sub-channel is calculated by using the Bhattacharyya parameter, density evolution or Gaussian approximation method. Based on the reliability of each sub-channel, an information bit sequence is transmitted on the K sub-channels with high reliability, and the remaining $(N-K)$ channels with the lower reliability is used to transmit the frozen bits (generally 0's), thereby constructing a coded polar codeword with rate $R = K/N$. The set A represents the information bit index set, and the set A^c, complement to A, represents the frozen bit index set. The encoding process of polar codes is described as:

$$x_1^N = u_1^N G_N \tag{1}$$

where G_N is the generator matrix, $G_N = F^{\otimes n}$, $(n = \log N)$, $F = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$, $F^{\otimes n}$ represents the n-th Kronecher product of the basis matrix F, u_1^N is the input symbol sequence with K information bits and $(N - K)$ frozen bits, taking on the values $\{0, 1\}$, and x_1^N represents coded sequence, with the same definition field as the input symbol sequence. After x_1^N is fed into a modulator and sent over an AWGN channel, the sequence y_1^N is obtained. Taking $N = 8$ as an example, for which an encoding structure [16], [17] is shown in Fig. 1. When the variance of the AWGN channel is $\sigma^2 = 0.987$, set $A = \{4, 6, 7, 8\}$ and $A^c = \{1, 2, 3, 5\}$. Suppose $u_A = (1, 0, 1, 1)$, $u_{A^c} = (0, 0, 0, 0)$, and the input sequence is $u_1^N = (0, 0, 0, 1, 0, 0, 1, 1)$. Then, $x_1^N = u_1^N G_8 = u_1^N F^{\otimes 3} = (1, 0, 1, 0, 0, 1, 0, 1)$.

B. SC DECODING ALGORITHM

Here is the polar code decoding process: based on the received sequence, an estimated value of the information sequence is obtained through a series of iterative operations. $u_1^N = (u_A, u_{A^c})$ is the estimated value of u_1^N by the known sequence (y_1^N, u_{A^c}). The SC decoder includes N decoding units. The frozen bits are known to the decoder. For the information bits, the decoder performs the likelihood ratio

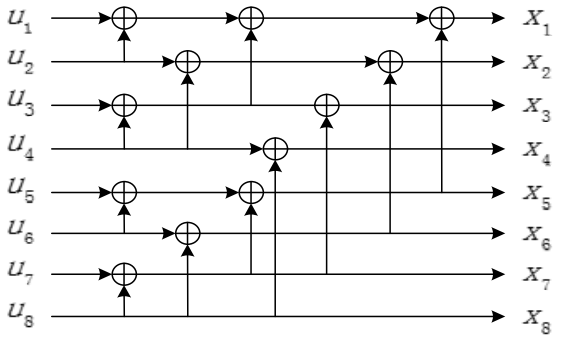


FIGURE 1. Encoding structure of $N = 8$ polar code.

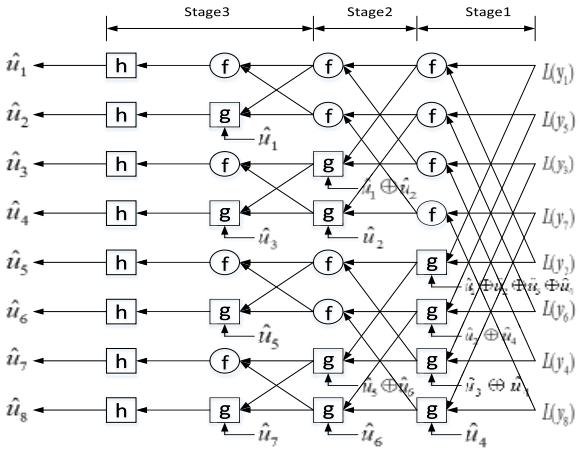


FIGURE 2. SC decoding structure of $N = 8$ polar code.

iterative operation and uses the previously estimated value $\hat{u}_1^{i-1}$ to decode the i-th bit, i.e., the current bit. The likelihood ratio is calculated as:

$$\hat{u}_i = h\left(L_N^{(i)}\left(y_1^N, \hat{u}_1^{i-1}\right)\right) \tag{2}$$

where $L_N^{(i)}\left(y_1^N, \hat{u}_1^{i-1}\right) \triangleq \log \frac{W_N^{(i)}\left(y_1^N, u_1^{i-1} | u_i = 0\right)}{W_N^{(i)}\left(y_1^N, u_1^{i-1} | u_i = 1\right)}$. The decision function h makes a decoding decision based on the calculated log-likelihood ratio expressed as:

$$h\left(L_N^{(i)}\left(y_1^N, \hat{u}_1^{i-1}\right)\right) = \begin{cases} 1, & L_N^{(i)}(y_1^N, \hat{u}_1^{i-1}) < 0 \text{ for bit } i \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

Fig. 2 shows the SC decoding structure for the case of $N = 8$. The calculation of the log-likelihood ratio (LLR) $L_N^{(i)}(\cdot)$ has two nodes, called g and f. The expressions to calculate the LLRs for g and f nodes are [18]

$$f(L_1, L_2) = \text{sign}(L_1) \cdot \text{sign}(L_2) \cdot \min(|L_1|, |L_2|), \tag{4}$$

$$g(L_1, L_2, u) = (-1)^u L_1 + L_2. \tag{5}$$

The iterative process of $L_N^{(i)}(\cdot)$ is expressed as:

$$\begin{aligned} &L_N^{(2i-1)}(y_1^N, \hat{u}_1^{2i-2}) \\ &= f\left[L_{N/2}^{(i)}(y_1^{N/2}, \hat{u}_{1,e}^{2i-2} \oplus \hat{u}_{1,o}^{2i-2}), L_{N/2}^{(i)}(y_{N/2+1}^N, \hat{u}_{1,e}^{2i-2})\right] \end{aligned} \tag{6}$$

TABLE 1. Error digits of polar codes with different code parameters.

Codelength	$N=256$				$N=512$				$N=1024$			
	E_b/N_0 (dB)				E_b/N_0 (dB)				E_b/N_0 (dB)			
	0	1	2	3	0	1	2	3	0	1	2	3
1 bit error	16.3%	49.9%	63.3%	90.3%	6.5%	44.4%	74.5%	95.2%	0.6%	39.0%	86.72%	98.4%
2 bit errors	18.5%	25.6%	23.4%	5.5%	9.8%	25.4%	19.9%	4.0%	3.2%	25.8%	11.0%	1.14%
3 bit errors	16.3%	13.6%	7.2%	2.7%	8.7%	14.6%	4.4%	0.4%	4.5%	14.0%	2.0%	0.3%
4 bit errors	13.7%	5.6%	3.8%	1.3%	11.6%	6.3%	1.3%	0.3%	3.4%	11.4%	0.2%	0.1%
>4 bit errors	35.2%	5.3%	2.3%	0.2%	63.4%	9.3%	0.2%	0.1%	88.3%	9.8%	0.08%	0.06%

$$L_N^{(2i)}(y_1^N, \hat{u}_1^{2i-1}) = g\left[L_{N/2}^{(i)}(y_1^{N/2}, \hat{u}_{1,e}^{2i-2} \oplus \hat{u}_{1,o}^{2i-2}), L_{N/2}^{(i)}(y_{N/2+1}^N, \hat{u}_{1,e}^{2i-2}), \hat{u}_{2i-1}\right] \tag{7}$$

where $\hat{u}_e$ and $\hat{u}_o$ represent, respectively, the sequence of elements with even subscripts and the sequence of elements with odd subscripts. The initial value of the LLR is:

$$L(y_i) = \log \frac{W(y_i|0)}{W(y_i|1)}, \quad i = 1, 2, \cdots, N. \tag{8}$$

For an AWGN Channel, it can also be calculated as:

$$L(y_i) = \log \frac{2y_i}{\sigma^2}, \quad i = 1, 2, \cdots, N \tag{9}$$

III. CONSTRUCTION OF SEGMENTED MULTI-CRC POLAR CODES

A. EFFECT OF ERROR PROPAGATION UNDER SC DECODING ALGORITHM

As shown in Fig. 2, the likelihood ratio calculation for all g nodes requires some decoded bits. This example assumes information bit index set $A = \{3, 4, 7, 8\}$ and freezing bit index set $A^c = \{1, 2, 5, 6\}$. Suppose all the information bits are 0's for simplicity. The information bit $\hat{u}_3$ is the first erroneous bit due to channel noise and is decoded to be 1. In the decision process of the next information bit $\hat{u}_4$, the information $\hat{u}_3 = 1$ is used to calculate the LLR. The erroneous decision to $\hat{u}_3$ will propagate its effects. This reveals: a) The SC decoding process of polar codes is serial decoding performed in order; b) A current erroneous decoding decision will affect the decoding performance of subsequent bits; that is, SC decoding exhibits error propagation.

Here we simulate the impact of error propagation on SC decoding performance. The simulations assume that the information bits sent at the transmitter are all 0's. If the receiver receives 1's, error occurred. Obviously, the first error bit is caused by channel noise. In order to emulate the case without error propagation, the erroneous bits are corrected automatically during the decoding process; that is, all the received 1's bits are set to 0's automatically in the following decoding process, and this is named SC Emulate. The results in Fig. 3 are obtained from 1500,000 Monte Carlo experiments, and they show that the performance of polar codes

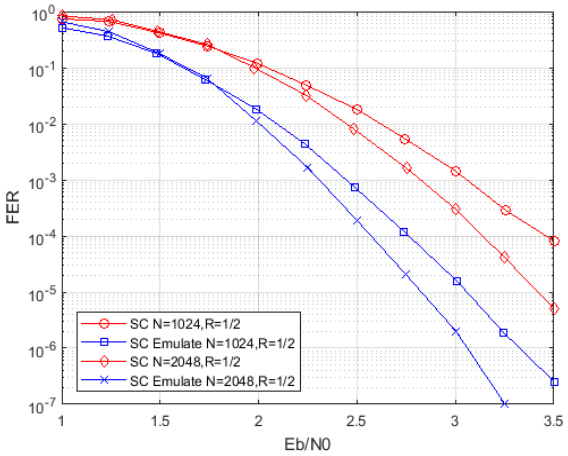


FIGURE 3. Performance comparison of SC and SC emulate without error propagation.

is significantly improved if error propagation effects can be eliminated.

In order to further assess the characteristics of error propagation effects on polar codes, the performance of codes of different lengths are evaluated, as shown in Table 1. For example, for code with length 512 at SNR of 1 dB, 44.4% of the frames are single-bit-error frames and 55.6% of the frames have multiple bit errors. As the code length and SNR increase, the number of erroneous bits due to error propagation in each frame gradually decreases.

B. CONSTRUCTION OF MULTI-CHECK POLAR CODE FOR SUPPRESSING ERROR PROPAGATION

The goal here is to construct a polar code that is more immune to error propagation than existing polar codes. The proposed structure consists of multiple segmented portions each with a CRC, which allows decoding error to be corrected in time to mitigate error propagation effects. For this particular structure, a new decoding algorithm is also developed that takes advantages of the proposed structure for very effective decoding. The structure is shown in Fig. 4. Two segmentation methods are proposed: a uniform segmentation structure and a heuristic segmentation structure.

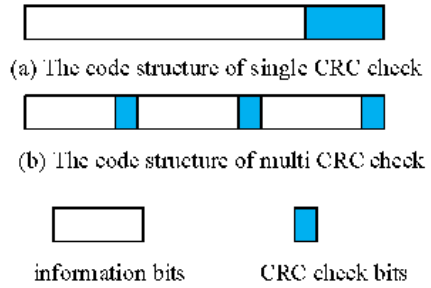


FIGURE 4. Segmented multi-CRC polarization code structure.

1) UNIFORM SEGMENTATION

In the uniform segmentation structure, the information bits are segmented uniformly, and CRC bits are added to each segment. Suppose the number of segments is P, and the highest power of the CRC generator polynomial is r. Then $M = K/P$ is the number of information bits per segment, and the last r bits are CRC bits, each arranged as

$$I_p = \left\{ i_p^{(1)}, i_p^{(2)}, \cdots, i_p^{(M-r)}, c_p^{(1)}, c_p^{(2)}, \cdots, c_p^{(r)} \right\}, \quad p \in \{1, 2, \cdots, P\} \quad (10)$$

where $i_p^{(s)}$, $1 \le s \le M - r$, represents the information bits and $c_p^{(t)}$, $1 \le t \le r$, represents the CRC bits.

2) HEURISTIC CONSTRUCTION

Uniform segmentation does not effectively utilize the statistical information of channel polarization, and it cannot guarantee an even distribution of the erroneous bits among the segments. In order to improve the effectiveness of the flipping and reduce the long decoding delay caused by unbalanced local segmentation, the number of error bits in each segment should be as balanced as possible. A self-heuristic segmentation structure that takes into consideration the channel polarization and error probability characteristics of the polar code in each sub-channel is proposed here.

The reliability distribution of each sub-channel can be calculated by using Bhattacharyya bounds method [19], [20]. Consider the example of code with length $N = 1024$, rate $R = 0.5$, and $Eb/N0 = 1dB$. Fig. 5 shows the capacity distribution of the sub-channels. It is observed that the errors occur in bursts with the channel polarization process. The bit errors are non-uniformly scattered in all information channels, and the error probability may increase dramatically at some locations.

The dotted line in Fig. 5 shows that the information sequence is divided into several segments according to the burst. The capacity gap between the two sub-channels adjacent to the burst error point is much larger than the capacity gap between other adjacent sub-channels. Taking the burst point as the dividing line, the information sequence is divided into P segments, and the last block of bits of each segment are the CRC bits. Assume that one segment contains L ($r \le L \le N$) information bits and r check bits.

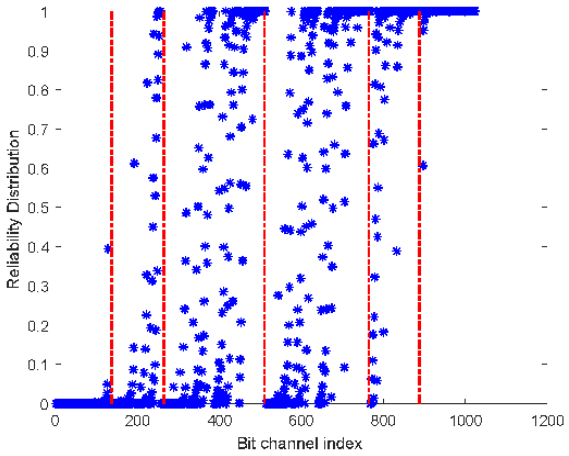


FIGURE 5. Polarized sub-channel capacity distribution ($N = 1024$).

The distribution can be expressed as:

$$I_p = \left\{ i_p^{(1)}, i_p^{(2)}, \cdots, i_p^{(L-r)}, c_p^{(1)}, c_p^{(2)}, \cdots, c_p^{(r)} \right\}, \quad p \in \{1, 2, \cdots, P\}. \quad (11)$$

As a heuristic segmentation method, the number of segments is not fixed. We can use the reliability changes of the adjacent sub-channels. If the reliability changes exceed a threshold T, a new segment is generated. In the examples simulated, a threshold $T = 0.5$ is chosen, resulting in 6 segments.

This heuristic segmentation utilizes the polarization characteristics of the polar code to ensure that the error bits are distributed as uniformly as possible in all segments to improve decoding performance.

IV. M-SCFLIP DECODING ALGORITHM

Based on the SC decoding algorithm, the successive cancellation flip decoding (SCFlip) algorithm flips the bit according to CRC results. The basic idea is as follows. In the encoding phase, first, r bits are placed at the end of the information sequence u_A. Then the entire sequence is coded by the polar code. In the decoding stage, after SC decoding CRC is performed on the decoded codeword. If the CRC check passes, then decoding is successful; otherwise, the T information bits at the position with smallest value of $|L_N^{(i)}(y_1^N, \hat{u}_1^{i-1})|$, $i \in A \cap \{1, 2, \cdots, \text{ind}_p\}$ will be flipped, but only one bit is flipped at a time. The previously flipped bit is restored as the second bit is flipped. After bit flipping, CRC is performed again. If after the T bits have been flipped one by one but CRC check still fails, then flipping process is declared failure, and the initial SC decoding result is considered as final.

For the constructed multi-CRC polar code, the SCFlip operation can be executed in each segment, and is called M-SCFlip. The polar code with the segmented structure increases the number of CRC check times and the bit flipping opportunities, which will lead to better decoding performance.

TABLE 2. Parameter settings of SCFLIP and M-SCFLIp algorithm.

Decoding algorithm	Code A			Code B			Code C		
	P	T	R	P	T	R	P	T	R
M-SCFlip (1)	3	16	11/16	6	16	11/16	8	16	5/8
M-SCFlip (2)	4	16	3/4	4	16	5/8	4	16	9/16
SCFlip	-	16	9/16	-	16	17/32	-	16	33/64

The specific process of the M-SCFlip algorithm is as follows.

Step 1: Let P be the number of segments of the polar code. The sub-channel indices of the segment points are placed in the set $\text{Ind} = \{\text{ind}_1, \text{ind}_2, \cdots, \text{ind}_p\}$. The information bit sequence is encoded to generate codeword x_1^N, and y_1^N is the received sequence.

Step 2: Perform SC decoding. After completing SC decoding for each segment $p \in \{1, 2, \cdots, P\}$, perform local CRC check. If CRC check fails, then sort $L_N^{(i)}(y_1^N, \hat{u}_1^{i-1})$ in ascending order. The bit set $Flip = \{F_1, F_2, \cdots, F_T\}$ to be flipped is constructed with the corresponding channel index I of the T minimum values $L_N^{(i)}(y_1^N, \hat{u}_1^{i-1})$.

Step 3: Select the bit corresponding to F_1 from the set $Flip$ to flip, and then perform SC decoding on the bit between bit F_1+1 and bit ind_p. If the decoding result does not pass CRC check, then bit F_1 is restored. Then, flip the bit corresponding to bit F_2 in the set $Flip$. After flipping the whole set $Flip$, if still no valid codeword is generated, then the flipping fails, and the result from the initial SC decoding is used as the final decoding result of this local segment, and the process moves forward to Step 4 below. If a decoding result passes CRC check, then store the flipping results and continue to Step 4.

Step 4: Repeat the above steps for all segments and output the whole decoded codeword $\hat{u}_1^N$.

V. SIMULATION AND RESULT ANALYSIS

A. SIMULATION PARAMETER SETTING

The parameters used in the simulation are as follows. Coherent binary phase shift keying (BPSK) modulation over an AWGN channel is considered. Gaussian approximation is used for information bit selection. Multi-CRC polynomial is $g(D) = D^4 + D + 1$, and CRC polynomial for single-segment case is $g(D) = D^{16} + D^{15} + D^2 + 1$. Since the added CRC bits have different lengths, the code rate is defined as $R = (K + r)/N$, and the effective bit rate is $R' = K/N$, where r is the number of bits allocated for CRC. In the comparison, the effective code rate is the same for all schemes. Three codes are adopted in the simulation: Code A ($N = 256$), Code B ($N = 512$), and Code C ($N = 1024$).

B. PERFORMANCE WITH SCL DECODING

The results in Fig. 6 show that when the SCL decoding algorithm is executed, the performance of the segmented multi-CRC codeword is slightly worse than that of the single-segment codeword. This main reason is that the segmentation structure shortens the code length of each part. Also to maintain the same code rate as the conventional polar code, the number of CRC bits per segment will be lower, which reduces its error-detection ability. But the performance difference in the low-SNR region is negligible; in high-SNR region, their performances are the same because the choice of the survival path is related to the reliability of the decoder output and the CRC check. As SNR increases, the decoding error probability decreases, and the error detection capability of the CRC plays a more important role.

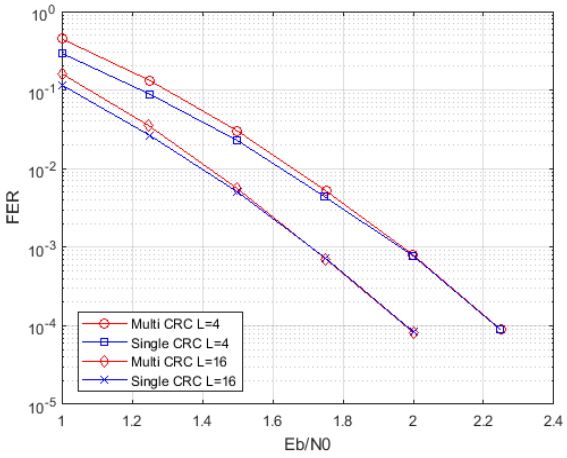
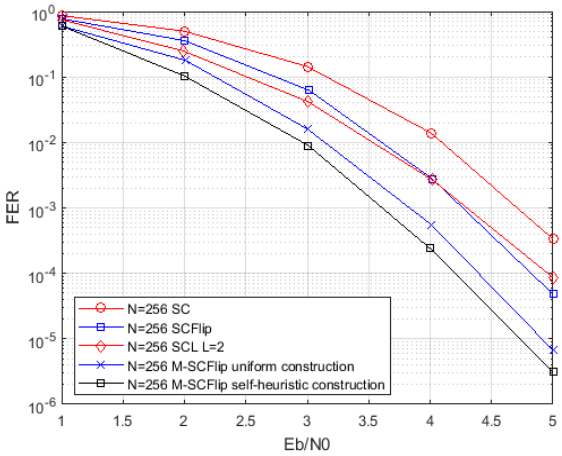


FIGURE 6. Performance of multi-CRC codewords with SCL decoding ($N = 1024$, $R = 0.5$).

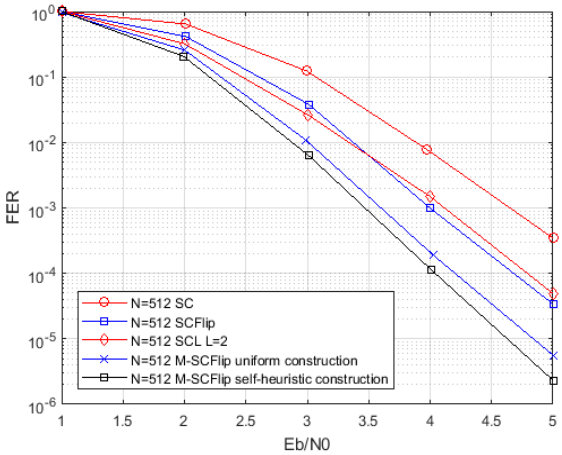
C. PERFORMANCE WITH M-SCFLIp DECODING

In this section, the performance of the proposed code employing the M-SCFlip algorithm, constructed with both uniform and self-heuristic methods is simulated and compared the SC, SCL, SCFlip, and M-SCFlip algorithms for three different codes, Code A, Code B, and Code C. Assuming the same effective code rate, the parameters of the three code words in the case of the SCFlip and the M-SCFlip algorithms are as shown in Table 2. M-SCFlip (1) represents the code constructed from a heuristic method, and M-SCFlip (2) represents the code constructed with the uniform method. Parameter P is the number of segments and T is the length of the set of bits to be flipped.

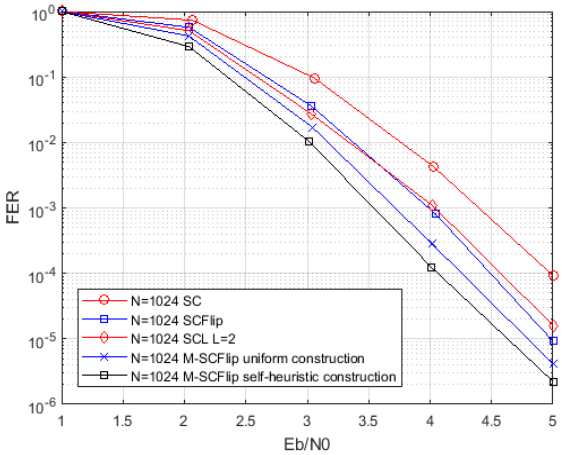
It is observed from Fig. 7 that the M-SCFlip algorithm—the improved SC bit-flip decoding algorithm proposed in this



(a) Code A, $N = 256$



(b) Code B, $N = 512$



(c) Code C, $N = 1024$

FIGURE 7. Frame error rate of different decoding algorithms.

paper−drastically outperforms SC decoding. It is also much more effective than the SCL ($L = 2$) and the SCFlip algorithms. Because the M-SCFlip algorithm implements

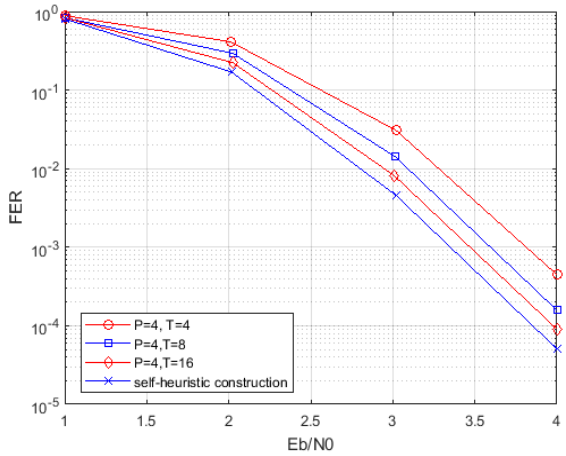


FIGURE 8. Performance as a function of parameter T (Code B, $N = 512$).

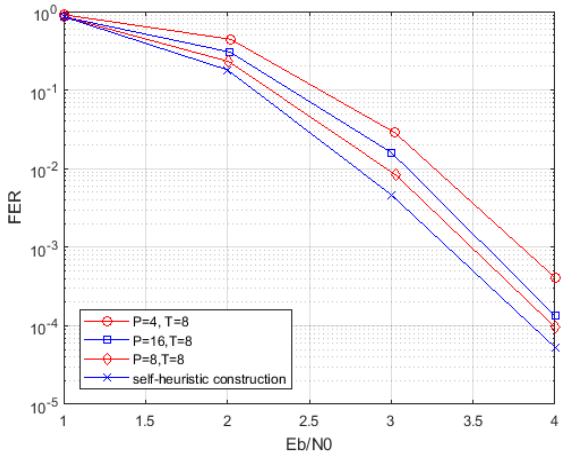


FIGURE 9. Performance as a function of parameter P (Code B, $N = 512$).

multiple flips in time, the error propagation issue with SC decoding is significantly mitigated. The M-SCFlip algorithm constructed with a heuristic method performs slightly better than random construction because the heuristic segmentation construction method takes the channel polarization of the polar code into consideration, so the error bits are distributed more evenly in each segment.

Fig. 8 shows the error performance of Code B, aiming to assess the influence of the number bits to be flipped T on the decoding performance of the M-SCFlip decoding algorithm. The number of segments of the heuristic structure P is set to 4. The results show that when the number of segments P is the same, the larger the number of bits to be flipped, the better the performance. Also, the heuristic M-SCFlip algorithm performs better than the uniform structure under the same conditions.

Fig. 9 evaluates the effect of the number of segments P on the performance of the M-SCFlip algorithm. With uniform segmentation, the performance with $P = 8$ is better than with $P = 4$. However, when P reaches 16, the performance of

TABLE 3. Simulation results of SCFLIP and M-SCFLIp algorithms.

Algorithm	Code A		Code B		Code C	
	E_b/N_0 (dB)	Gain (dB)	E_b/N_0 (dB)	Gain (dB)	E_b/N_0 (dB)	Gain (dB)
M-SCFlip(1)	3.20	-	2.96	-	2.77	-
M-SCFlip(2)	3.37	0.17	3.13	0.17	2.94	0.17
SCFlip	3.80	0.60	3.62	0.66	3.20	0.43
SCL ($L = 2$)	3.95	0.75	3.75	0.79	3.41	0.64
SC	4.25	1.05	4.15	1.19	3.66	0.89

the algorithm deteriorates because the number of segments is too large, and the actual rate of the polarization code is large, resulting in a performance degradation. The decoding performance of the M-SCFlip algorithm based on the heuristic construction method with $P = 6$ is better than the uniform M-SCFlip algorithm under the same conditions. It observed from Figs. 8 and 9 that the heuristic configuration is a better strategy for the M-SCFlip algorithm, because it segments the information bits according to the error burst, which results in error bits more or less evenly distributed in all segments, thereby improving the performance.

Table 3 shows the gains achieved by using the proposed new algorithm over existing algorithms for Code A, Code B, and Code C at a frame error rate (FER) of 10^{-4}. The proposed M-SCFlip algorithm with heuristic multi-CRC construction achieves gains of approximately 0.60 dB, 0.66 dB, and 0.43 dB, respectively, over the SCFlip algorithm. Compared to SCL ($L = 2$), M-SCFlip achieves approximately 0.75 dB, 0.79 dB, and 0.64 dB of gains for code A, code B, and code C; compared with SC, the gains are approximately 1.05 dB, 1.19 dB, and 0.89 dB.

D. DECODING DELAY AND COMPUTATIONAL COMPLEXITY

Here, we use the computational complexity to evaluate the decoding delay. The computational complexity of the M-SCFlip algorithm is related to the number of flipped bits. Since the number of flipped bits in the decoding process cannot be determined exactly, an average computational complexity is a more proper measure. In the M-SCFlip decoding process, bit flipping is performed only when a decision error occurs. Therefore, the average complexity of the algorithm should take account into the frame error rate of the SC decoding algorithm. Thus, the average complexity can be expressed as

$$O(N \log N) + PT \cdot P_e(R, SNR) \cdot O\left(\frac{N}{P} \log N\right) \quad (12)$$

where N is the code length, P is the number of segments, T is the size of the set of bits to be flipped, and $P_e(R, SNR)$ is the FER as a function of SNR and code rate R. As SNR increases, $P_e(R, SNR)$ decreases, the computational complexity of the M-SCFlip algorithm also decreases. The decoding delay of M-SCFlip is obviously lower than that of SCL, especially for large L.

For SCL decoding, all L surviving paths with the size K bits must be saved in memory until the K-th information bit is decoded. So, $K \times L$ memory units are required to store the L surviving paths. In the M-SCFlip decoder, CRC check is done on each sub-block and only the selected sequences are stored, which releases many memory units for use in the next decoding process. An approximate memory requirement is as follows. Let the information sequence be divided into P sub-blocks, meaning that the length of each sub-block is K/P. Then K/P memory units are required to execute the SC decoding process for each sub-block. The decoder needs an extra of T memory units to store the selected bits to be flipped and K bits to store the output bits. For the whole decoding process, the total memory space needed is $K/P + T + K$.

VI. CONCLUSION

This paper proposes a segmented multi-CRC polar code and develops an algorithm, called M-SCFlip decoding algorithm, for this particular code structure to significantly improve the error performance. Based on the channel polarization characteristics, the algorithm divides the information sequences and places the CRC bits at the end of each segment. The decoding errors are corrected in time, which effectively mitigates error propagation in the decoding process. Results obtained show that under the same code rate and code length, the performance of the constructed code with the proposed M-SCFlip algorithm results in much better error performance than existing codes and decoding algorithms. Additionally, the proposed code has a lower decoding delay and requires less memory space for decoding than existing codes and decoding algorithms.

REFERENCES

[1] E. Arıkan, "Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels," *IEEE Trans. Inf. Theory*, vol. 55, no. 7, pp. 3051–3073, Jul. 2009. doi: 10.1109/TIT.2009.2021379.

[2] I. Tal and A. Vardy, "How to construct polar codes," *IEEE Trans. Inf. Theory*, vol. 59, no. 10, pp. 6562–6582, Oct. 2013. doi: 10.1109/TIT.2013.2272694.

[3] P. Trifonov and V. Miloslavskaya, "Polar subcodes," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 2, pp. 254–266, Feb. 2016. doi: 10.1109/JSAC.2015.2504269.

[4] M. Mondelli, S. H. Hassani, I. Maric, D. Hui, and S.-N. Hong, "Capacity-achieving rate-compatible polar codes for general channels," in *Proc. Wireless Commun. Netw. Conf. Workshops (WCNCW)*, San Francisco, CA, USA, Mar. 2017, pp. 1–6.
[5] P. Trifonov, "Efficient design and decoding of polar codes," *IEEE Trans. Commun.*, vol. 60, no. 11, pp. 3221–3227, Nov. 2012. doi: 10.1109/TCOMM.2012.081512.110872.
[6] A. Hadi, E. Alsusa, and K. M. Rabie, "A method to enhance the performance of successive cancellation decoding in polar codes," in *Proc. 10th Int. Symp. Commun. Syst., Netw. Digit. Signal Process. (CSNDSP)*, Prague, Czech Republic, Jul. 2016, pp. 1–5.
[7] K. Chen, K. Niu, and J. R. Lin, "List successive cancellation decoding of polar codes," *Electron. Lett.*, vol. 48, no. 9, pp. 500–501, 2012. doi: 10.1049/el.2011.3334.
[8] I. Tal and A. Vardy, "List decoding of polar codes," *IEEE Trans. Inf. Theory*, vol. 61, no. 5, pp. 2213–2226, May 2015. doi: 10.1109/TIT.2015.2410251.
[9] K. Niu and K. Chen, "Stack decoding of polar codes," *Electron. Lett.*, vol. 48, no. 12, pp. 695–697, Jun. 2012. doi: 10.1049/el.2012.1459.
[10] K. Niu and K. Chen, "CRC-aided decoding of polar codes," *IEEE Commun. Lett.*, vol. 16, no. 10, pp. 1668–1671, Oct. 2012. doi: 10.1109/LCOMM.2012.090312.121501.
[11] W. Song, H. Zhou, Y. Zhao, S. Zhang, X. You, and C. Zhang, "Low-complexity segmented CRC-aided SC stack decoder for polar codes," in *Proc. 50th Asilomar Conf. Signals, Syst. Comput.*, Pacific Grove, CA, USA, Nov. 2016, pp. 1189–1193.
[12] 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.*, Pacific Grove, CA, USA, Nov. 2014, pp. 2116–2120.
[13] 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. doi: 10.1109/ACCESS.2018.2873821.
[14] Y. Wang, K. R. Narayanan, and Y.-C. Huang, "Interleaved concatenations of polar codes with BCH and convolutional codes," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 2, pp. 267–277, Feb. 2016. doi: 10.1109/JSAC.2015.2504320.
[15] W. Wang and L. Li, "Efficient construction of polar codes," in *Proc. Int. Wireless Commun. Mobile Comput. Conf. (IWCMC)*, Valencia, Spain, Jun. 2017, pp. 1594–1598.
[16] B. Yuan and K. K. Parhi, "Low-latency successive-cancellation polar decoder architectures using 2-bit decoding," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 61, no. 4, pp. 1241–1254, Apr. 2014. doi: 10.1109/TCSI.2013.2283779.
[17] B. Yuan and K. K. Parhi, "Successive cancellation list polar decoder using log-likelihood ratios," in *Proc. 48th Asilomar Conf. Signals, Syst. Comput.*, Pacific Grove, CA, USA, Nov. 2014, pp. 548–552.
[18] 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. doi: 10.1109/TSP.2015.2439211.
[19] H. Li and J. Yuan, "A practical construction method for polar codes in AWGN channels," in *Proc. IEEE Tencon-Spring* ydney, NSW, Australia, Apr. 2013, pp. 223–226.
[20] H. Vangala, E. Viterbo, and Y. Hong, "A comparative study of polar code constructions for the AWGN channel," Jan. 2015, *arXiv:1501.02473*. [Online]. Available: https://arxiv.org/abs/1501.02473

RUI GUO received the Ph.D. degree from Zhejiang University, Hangzhou, China, in 2007. He is currently an Associate Professor with the School of Communication Engineering, Hangzhou Dianzi University, Hangzhou. He is also a Visiting Scholar with Oregon State University, from 2018 to 2019. His research interests include wireless communication and channel coding.

KANGNI CHEN is currently pursuing the master's degree with the School of Communication Engineering, Hangzhou Dianzi University. Her research interests include wireless communication and channel encoding.

HUAPING LIU (S'95–M'97–SM'08) received the B.S. and M.S. degrees from the Nanjing University of Posts and Telecommunications, Nanjing, China, in 1987 and 1990, respectively, and the Ph.D. degree from the New Jersey Institute of Technology, Newark, NJ, USA, in 1997, all in electrical engineering.

From 1997 to 2001, he was with Lucent Technologies, Whippany, NJ, USA. In 2001, he joined the School of Electrical Engineering and Computer Science, Oregon State University, Corvallis, OR, USA, where he has been a Full Professor, since 2011. His research interests include modulation and detection, multiple-antenna techniques, and localization systems.