

Multiple Node Flip Fast-SSC Decoding Algorithm for Polar Codes Based on Node Reliability

Guo Rui^{1*}, Yang Pei¹, Ying Na¹, and Wang Lixin¹

¹ School of Communication Engineering, Hangzhou Dianzi University, Hangzhou, China
[e-mail: guorui@hdu.edu.cn]

*Corresponding author: Guo Rui

*Received October 10, 2021; revised December 13, 2021; accepted January 15, 2022;
published February 28, 2022*

Abstract

This paper presents a fast-simplified successive cancellation (SC) flipping (Fast-SSC-Flip) decoding algorithm for polar code. Firstly, by researching the probability distribution of the number of error bits in a node caused by channel noise in simplified-SC (SSC) decoder, a measurement criterion of node reliability is proposed. Under the guidance of the criterion, the most unreliable nodes are firstly located, then the unreliable bits are selected for flipping, so as to realize Fast-SSC-Flip decoding algorithm based on node reliability (NR-Fast-SSC-Flip). Secondly, we extended the proposed NR-Fast-SSC-Flip to multiple node (NR-Fast-SSC-Flip- ω) by considering dynamic update to measure node reliability, where ω is the order of flip-nodes set. The extended algorithm can correct the error bits in multiple nodes, and get good performance at medium and high signal-to-noise (SNR) region. Simulation results show that the proposed NR-Fast-SSC-Flip decoder can obtain 0.27dB and 0.17dB gains, respectively, compared with the traditional Fast-SSC-Flip [14] and the newly proposed two-bit-flipping Fast-SSC (Fast-SSC-2Flip-E2) [18] under the same conditions. Compared with the newly proposed partitioned Fast-SSC-Flip (PA-Fast-SSC-Flip) ($s=4$) [18], the proposed NR-Fast-SSC-Flip- ω ($\omega=2$) decoder can obtain about 0.21dB gain, and the FER performance exceeds the cyclic-redundancy-check (CRC) aided SC-list (CRC-SCL) decoder ($L=4$).

Keywords: Polar codes, Dynamic update, Fast-ssc-flip, Node reliability, Multiple-bit-flipping.

This work was supported in part by the National Natural Science Foundation Joint Research Program of China under Project U1709220

1. Introduction

Polar code is linear error correcting code, and is strictly proven can achieve the capacity of binary discrete memoryless channel [1]. Due to its good performance, polar code has been used as the coding scheme of Enhanced Mobile Broadband (eMBB) control channel, and can be combined with channel estimation technology to further improve the system performance [2-3]. Although the complexity of SC decoding algorithm is low, the performance of the decoder is unsatisfactory when the code length is short or medium, so researchers proposed SCL [4] and CRC-SCL [5] algorithm. The performance of polar code using CRC-SCL decoder can completely compete with low density parity check (LDPC) and Turbo codes. For the sake of reducing the storage and computational complexity of CRC-SCL decoding, researchers proposed SC-Stack decoding (SCS) algorithm [6-7].

Except SCL decoder, researchers proposed SC-Flip decoder which can effectively find the first error bit in the SC decoding process to improve the performance of SC decoding at the expense of decoding delay [8]. The key point of the SC-Flip decoding algorithm is how to find the first error bit accurately. In response to this critical problem, the author in [9] exploited the structure of SC decoding tree to construct a critical set that can effectively include the first error bit. The author in [10] proposed a dynamic SC-Flip (D-SCF) algorithm by considering the nature of SC sequential decoding to select non-frozen bits into flip-bit-set. To reduce the complexity of D-SCF algorithm, the author in [11] chose the non-frozen bits as flipping bits when the bit error rate (BER) value calculated by Gaussian Approximation (GA) algorithm is higher than the estimated BER based on log-likelihood-ratio (LLR) after SC decoding. By introducing this criterion, the decoding complexity can effectively be reduced while the performance is close to D-SCF algorithm.

For the sake of reducing the latency of the SC decoding algorithm and improving the throughput, researchers proposed the SSC decoding algorithm by directly decoding Rate-0 and Rate-1 nodes [13]. The author in [12] further proposed single-parity-check (SPC) nodes and repetition (REP) nodes in the SC decoding process to further reduce decoding delay. The author in [15] proposed another four nodes that can be directly decoded. By directly decoding more special nodes, the latency of SC decoding can be sharply reduced, and the throughput of the polar decoders can be greatly improved, too.

A combination algorithm of Fast-SSC and SC-Flip is proposed in [14] to reduce the high latency of the SC-Flip decoding algorithm. Reference [16] further proposed a lossless SPC node flipping decoding algorithm and effectively improved the speed of Fast-SSC-Flip [14] decoding by extending the numbers of special node to five. The author in [17] improved Fast-SSC-Flip decoder performance by considering the probability of special nodes which is decoded correctly and the reliability of candidate bits (CBs) in each node. The author of [18] proposed two enhanced multi-bit flip Fast-SSC decoding algorithms, which further improved the performance of Fast-SSC decoding with the same complexity as traditional Fast-SSC-Flip algorithm. However, the current Fast-SSC-Flip decoding algorithm considers the reliability of CB to locate the first error bit. The complexity of this method will sharp increase with the increase of the number of CBs. On the other hand, in most of the current literature, Fast-SSC-Flip decoding algorithm can only correct one bit error, and the performance is limited.

In this article, we first propose a criterion to measure node reliability (NR), which can locate the unreliable nodes, and then find the error bits caused by channel noise in unreliable nodes to reduce complexity. In order to effectively flip multiple error bits introduced by

channel noise, we combine NR criterion with dynamic update process so that the new measurement method can measure the reliability of multiple nodes.

The innovation and contribution of this paper are summarized as follows.

- A simple method to measure single NR is proposed and present NR-Fast-SSC-Flip decoding algorithm based on NR. Compared with [17], the proposed NR-Fast-SSC-Flip algorithm can maintain similar performance and reduce complexity, and has better performance than the traditional Fast-SSC-Flip algorithm.
- The single node reliability measurement method is extended to measure multiple nodes and present NR-Fast-SSC-Flip- ω decoding algorithm based on new measurement method. Compared with the existing Fast-SSC-Flip algorithms, the proposed NR-Fast-SSC-Flip- ω algorithm has better performance.

The reminder of the paper is organized as follows: the second part describes the decoding algorithm of polar codes. The third part introduces the proposed node reliability measurement criteria and the proposed NR-Fast-SSC-Flip decoder. The proposed NR-Fast-SSC-Flip- ω decoder is introduced in the fourth part. The fifth part gives the simulation results and the effect comparison between the proposed decoder and the existing Fast-SSC-Flip decoders. Finally, conclusions are drawn in Section 6.

2. PRELIMINARIES

2.1 Polar Codes And SC Decoding Algorithm

Considering a code length $N = 2^n$ and rate $R = \frac{K}{N}$ polar code $P(N, K)$, where K represents the numbers of information bits, the information bits are transmitted at K reliable polarized subchannels and remain frozen bits are transmitted in unreliable subchannels I^c . Let u_1^N be the information sequence, the encoded sequence x_1^N can be obtained by

$$x_1^N = u_1^N G_N \quad (1)$$

Where $G_N = F^{\otimes n}$, $F = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$, and $\otimes n$ denotes n -th Kronecker power.

The process of SC algorithm can be described by a tree as **Fig. 1.1**, where white and blank leaf nodes correspond to the frozen bits and information bits, respectively. Each node receives a vector of LLR $\alpha = \{\alpha_1, \alpha_2, \dots, \alpha_{2^S}\}$ from its parent, and computes the left LLR vector $\alpha^l = \{\alpha_1^l, \alpha_2^l, \dots, \alpha_{2^{S-1}}^l\}$ and right LLR vector $\alpha^r = \{\alpha_1^r, \alpha_2^r, \dots, \alpha_{2^{S-1}}^r\}$ to its child nodes

$$\alpha^l = \text{sign}(\alpha[i]\alpha[i + N_v / 2]) \min(|\alpha[i]|, |\alpha[i + N_v / 2]|) \quad (2)$$

$$\alpha^r[i] = \begin{cases} \alpha[i] + \alpha[i + N_v / 2], & \text{when } \beta^l[i] = 0 \\ \alpha[i + N_v / 2] - \alpha[i], & \text{otherwise} \end{cases} \quad (3)$$

where $N_v = 2^S$ is the length of constituent code at stage S , β represents the partial sum from their left and right child nodes and is calculated as follows

$$\beta[i] = \begin{cases} \beta^l[i] \oplus \beta^r[i], & \text{when } i < N_v / 2 \\ \beta^r[i - N_v / 2], & \text{otherwise} \end{cases} \quad (4)$$

The \oplus is the bitwise XOR operation. β can be calculated as

$$\beta[i] = \begin{cases} 0, & \text{when } \alpha[i] \geq 0 \text{ or } i \in I^c \\ 1, & \text{otherwise} \end{cases} \quad (5)$$

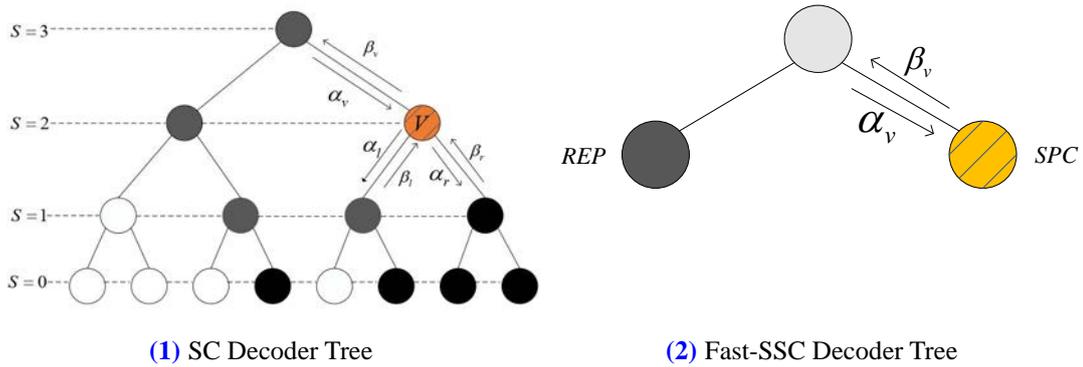


Fig. 1. Decoder tree representations of an (8,4) polar code

2.2 Fast-SSC And Fast-SSC-Flip Decoder

Fast-SSC decoder can further simplify SC decoding procedure by directly decoding various special node and improve the decoding latency performance of polar codes [12–13]. The corresponding pruned tree for Fast-SSC decoder is shown in Fig. 1.2.

Combining Fast-SSC decoding with SC-Flip decoding, Fast-SSC-Flip algorithm is given, which sharply reduces the delay of SC-Flip decoding. The more specific modifications are described as [16].

2.2.1 Rate-1 Node

The Rate-1 node v contains N_v information bits. The SSC decoding algorithm is a hard decision (HD) on the soft-input LLRs α_v . This node contains N_v CBs and the decision LLR value of the CB is

$$\lambda_i = |\alpha_v[i]| \quad (6)$$

2.2.2 REP Node

The REP node v has one single information bit in the last location. Thus, Fast-SSC decoding algorithm takes a sign detection on the sum of all input LLRs. This node only has one CB, and its decision LLR is given by

$$\lambda_i = \left| \sum_{j=0}^{N_v-1} \alpha_v[j] \right|, \text{ for } i = N_v - 1 \quad (7)$$

2.2.3 SPC Node

The SPC node v has $N_v - 1$ information bits and only one frozen bit on the most left index. These bits in SPC node must satisfy parity constraint, the HD estimated value of the soft-input is calculated as

$$\text{HD}(\alpha_v[i]) = \begin{cases} 0, & \text{when } \alpha_v[i] \geq 0 \\ 1, & \text{otherwise} \end{cases} \quad (8)$$

The parity of input LLR vector is p , the minimum decision LLR value of CBs is ϵ , and its index i_ϵ can be calculated as

$$\begin{cases} p = \bigoplus_{i=0}^{N_v-1} \text{HD}[i] \\ \epsilon = \min(|\alpha_0^{N_v-1}|) \\ i_\epsilon = \arg \min_i |\alpha_v[i]| \end{cases} \quad (9)$$

The final output of node v is

$$\beta_v[i] = \begin{cases} \text{HD}[i] \oplus p, & \text{when } i = i_\epsilon \\ \text{HD}[i], & \text{otherwise} \end{cases} \quad (10)$$

An algorithm almost without performance loss about calculating decision LLR was proposed in [16]

$$\lambda_i = |\alpha_v[i]| + (1-p) \cdot \epsilon, \text{ for } \forall 0 \leq i < N_v - 1 \text{ and } i \neq i_\epsilon \quad (11)$$

3. NR-Fast-SSC-Flip Decoding

In the current literature, the Fast-SSC-Flip decoding algorithm considers the reliability of the CB within a node, and used the decision LLR of the CB to locate the most unreliable decoding bit directly, which increases redundant calculations. The improved NR-Fast-SSC-Flip decoding algorithm can first locate the decoding error bits within one special node, and then select the most unreliable bit in the special node to be flipped. With the increase of code length or rate, NR-Fast-SSC-Flip decoding algorithm can reduce decoding complexity. Since the node position where the first error bit is located can be determined very accurately, compared with the traditional Fast-SSC-Flip decoding algorithm, the error-correction ability of the proposed decoder can be effectively improved. Simulation results in the fifth part also show that the performance of proposed decoder is better than traditional Fast-SSC-Flip decoder existing in the current literature.

3.1 Statistics of the number of error bits in the node

In our work, we first use the oracle-assisted SSC (SSC-Oracle) decoder with known information bit at the receiver to count the probability distribution of the number of error bits in each special node caused by channel noise when the SSC decoding fails.

Table 1. Influence of SNR (dB) and code length on the number proportion of error bits in first decoding error node

numbers \ SNR	1	2	3
1	0.9988	0.9997	1.0000
2+	0.0012	0.0003	0

(1) Length=1024

numbers \ Length	512	1024	2048
1	0.9994	0.9997	1.0000
2+	0.0006	0.0003	0

(2) SNR=2dB

As shown in **Table 1**, there is extremely high probability that only one error bit is contained in the first special node where decoding error occurs, and only one error bit require flip. Thus we approximately consider that the node is decoded correctly after effectively flipping trial and propose a criterion to measure node reliability. This criterion mainly guides us to locate the error bit in the decoding process to a special node, and then try to flip one certain error bit in this node.

3.2 Calculation of Q for Special Nodes

Notations: Let $\hat{\mathbf{v}}_i$ and \mathbf{v}_i be the decoded bit vector and corrected bit vector of the i -th node, respectively. $P(\hat{\mathbf{v}}_i)$ is the probability that the i -th node is incorrectly decoded when the channel output is \mathbf{y} and the previous nodes are correctly estimated, i.e. $P(\hat{\mathbf{v}}_i) \approx \Pr(\hat{\mathbf{v}}_i \neq \mathbf{v}_i | \hat{\mathbf{v}}_0 = \mathbf{v}_0, \dots, \hat{\mathbf{v}}_{i-1} = \mathbf{v}_{i-1}, \mathbf{y})$. However, since we cannot guarantee previous nodes have been correctly decoded. Thus, we use $Q(\hat{\mathbf{v}}_i) \approx \Pr(\hat{\mathbf{v}}_i \neq \mathbf{v}_i | \hat{\mathbf{v}}_0, \hat{\mathbf{v}}_1, \dots, \hat{\mathbf{v}}_{i-1}, \mathbf{y})$ to approximate $P(\hat{\mathbf{v}}_i)$.

Similarly, let e and \hat{e} be a CB and its estimate value, $E(\hat{e}) \approx \Pr(\hat{e} \neq e | \hat{\mathbf{v}}_0 = \mathbf{v}_0, \dots, \hat{\mathbf{v}}_{i-1} = \mathbf{v}_{i-1}, \mathbf{y})$ indicate the probability of bit e in i -th node which is incorrectly estimated. In practice, we use $S(\hat{e}) \approx \Pr(\hat{e} \neq e | \hat{\mathbf{v}}_0, \hat{\mathbf{v}}_1, \dots, \hat{\mathbf{v}}_{i-1}, \mathbf{y})$ to approximate $E(\hat{e})$.

Let $R(O_k) = \Pr(O_k | \hat{\mathbf{v}}_0, \hat{\mathbf{v}}_1, \dots, \hat{\mathbf{v}}_{i-1}, \mathbf{y})$ represent the probability of only k -th bit decoded incorrectly in the i -th node, and can be calculate as follows

$$R(O_k) = S(\hat{v}_k) \prod_{\substack{j=0 \\ j \in \xi/k}}^{n-1} (1 - S(\hat{v}_j)) \quad (12)$$

Where ξ is the index set of CBs in the i -th node. According to the statistical results from [Table 1](#), it is clear that almost one bit of decoding error occurs in a special node, we can calculate $Q(\hat{v}_i)$ approximately as

$$Q(\hat{v}_i) \approx \max(R(O_k | \hat{v}_0, \hat{v}_1, \dots, \hat{v}_{i-1}, \mathbf{Y})) \quad (13)$$

The $Q(\hat{v}_i)$ of Rate-1, REP and SPC nodes are discussed below due to decoding methods of various nodes are different.

Let λ_{i_σ} be minimum decision LLR value of CB in i -th node.

3.2.1 Rate-1 Node

For this node, each bit is a CB and estimation value of the CBs are independent to each other. The decision LLR of each CB v_i is $\lambda_i = |\alpha_v[i]|$, thus $S(\hat{v}_i)$ and $R(O_k)$ can be calculated as

$$S(\hat{v}_i) = \frac{1}{1 + \exp(\lambda_i)} \quad (14)$$

$$R(O_k) \leq \frac{1}{1 + \exp(\lambda_k)} \leq \frac{1}{1 + \exp(\lambda_{i_\sigma})} \quad (15)$$

$Q(\hat{v}_i)$ can be represented as

$$Q(\hat{v}_i) \approx \frac{1}{1 + \exp(\lambda_{i_\sigma})} \quad (16)$$

The error probability of the rate-1 node is set to its upper bound.

3.2.2 REP Node

The REP node has one single CB and its decision LLR is calculated by taking HD on the sum of all input LLRs. So, the $S(\hat{v}_i)$ and $Q(\hat{v}_i)$ can be calculated as

$$S(\hat{v}_i) = \frac{1}{1 + \exp(\lambda_i)}, \text{ for } i = N_v - 1 \quad (17)$$

$$Q(\hat{v}_i) = S(\hat{v}_{i_\sigma}) \quad (18)$$

Where $\sigma = N_v - 1$.

3.2.3 SPC Node

For the sake of simplification, we ignore the correlation between each estimated bit. The upper

bound of $S(\hat{v}_i)$ can be obtained by

$$S(\hat{v}_i) \leq \frac{1}{1 + \exp(\lambda_{i_\sigma})} \quad (19)$$

$R(O_k)$, $Q(\hat{v}_i)$ can be calculated as follows

$$R(O_k) \leq S(\hat{v}_k) \leq \frac{1}{1 + \exp(\lambda_{i_\sigma})}, k \neq i_\sigma \quad (20)$$

$$Q(\hat{v}_i) \approx \frac{1}{1 + \exp(\lambda_{i_\sigma})}$$

Similar to rate-1 node, we still take the upper bound of probability as the error probability of SPC node.

3.3 Calculation of Metric for Special Nodes

If an error occurs in a node, it may cause error propagation. Therefore, we consider the nature of Fast-SSC sequential decoding, and obtain the probability that the first $i - 1$ nodes decodes correctly and the i -th node decodes incorrectly

$$M_\alpha(i) = Q(\hat{v}_i) \prod_{\substack{k < i \\ k \in A}} (1 - Q(\hat{v}_k))$$

$$= \frac{1}{1 + \exp(\alpha \cdot \lambda_{i_\sigma})} \prod_{\substack{k < i \\ k \in A}} \frac{1}{1 + \exp(-\alpha \cdot \lambda_{k_\sigma})} \quad (21)$$

$$= \exp(-\alpha \cdot \lambda_{i_\sigma}) \cdot \prod_{\substack{k \leq i \\ k \in A}} \frac{1}{1 + \exp(-\alpha \cdot \lambda_{k_\sigma})}$$

By taking the logarithm of (21), and denoting $M'_\alpha(i) = -\frac{1}{\alpha} \log(M_\alpha(i))$, we can get the metric in the logarithmic domain

$$M'_\alpha(i) = \lambda_{i_\sigma} + \frac{1}{\alpha} \sum_{\substack{k \leq i \\ k \in A}} \log(1 + \exp(-\alpha \cdot \lambda_{k_\sigma})) \quad (22)$$

Where λ_{i_σ} is the decision LLR of i -th node, A is the set of index of non-Rate-0 nodes, parameter α is the weight, which is used to adjust the decoding order and decision LLR of node. When $\alpha \rightarrow 0$, the selection of the flip-nodes set mainly depends on the position of the node in the decoding order. When $\alpha \rightarrow 1$, the selection of the flip-nodes set mainly depends on the decision LLR of node.

3.4 Proposed NR-Fast-SSC-Flip Decoder

The NR-Fast-SSC-Flip decoder first executes initial SSC decoding algorithm. If the decoded codeword passes the CRC check, the decoded result is output and decoding process continuous. Otherwise, NR-Fast-SSC-Flip decoder selects and sorts the flip-nodes according to the metric in logarithmic domain. Continue to make new decoding attempts and perform node flip until the CRC check is passed or the maximum number of decoding attempts is reached. The detailed process of this algorithm is shown in **Table 2**. The functions in the algorithm are described as follows

Initial ($\lambda_1^{|A|}, T_1$): The metric of nodes in A is calculated by using (22), and sorted by ascend. Then, metric of first T_1 nodes are appended to set \mathbf{M}_Φ , and corresponding position indexes are appended to set Φ . Where \mathbf{M}_Φ is the node measurement set and Φ is the index position set of these nodes.

Table 2. NR-Fast-SSC-Flip Algorithm

Algorithm 1: NR-Fast-SSC-Flip Algorithm

```

 $[u_0^{N-1}, A, \lambda_1^{|A|}] \leftarrow \text{Fast-SSC}(y, I);$ 
if CRC Check( $\overline{u_0^{N-1}}$ ) fails then
     $[\Phi, \mathbf{M}_\Phi] \leftarrow \text{Initial}(\lambda_1^{|A|}, T_1);$ 
    for  $i = 1: T_1$  do
         $[u_0^{N-1}] \leftarrow \text{node flip}(y, I, T_2, \Phi(i));$ 
        if CRC Check( $\overline{u_0^{N-1}}$ ) success then
             $\overline{u_0^{N-1}} = u_0^{N-1};$ 
            break;
        end if
    end for
end if

```

The main difference between the proposed NR-Fast-SSC-Flip decoder and the traditional Fast-SSC-Flip decoder is that instead of directly choosing CB, we put candidate node (CN) into the set of flip-nodes. When a node needs to be flipped, proposed decoder selects CB from the node for attempting decoding. Since the number of non-Rate-0 nodes is much smaller than the number of CBs, the decoding complexity can be effectively reduced.

Table 3. Node Flip Function**Algorithm2: Node Flip Function**

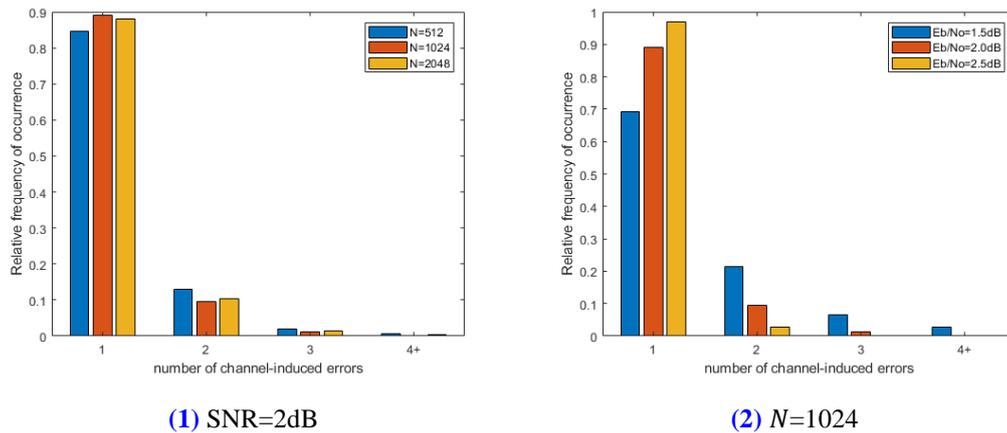
```

 $\varphi_{\Phi(i)} \leftarrow \text{sort}(LLR_{\Phi(i)});$ 
for  $ii = 1:T_2$  do
     $\widehat{u}_0^{N-1} \leftarrow \text{Fast-SSC-Flip}(y, I, \varphi_{\Phi(i)}(ii));$ 
    if  $\text{CRC Check}(\widehat{u}_0^{N-1})$  success then
         $\widehat{u}_0^{N-1} = \overline{\widehat{u}_0^{N-1}};$ 
        break;
    end if
end for

```

4. NR-Fast-SSC-Flip- ω DECODING

We use the SSC-Oracle decoder to obtain the normalized probability distribution diagram of the error nodes caused by channel noise in the SSC decoding process. Results are shown in Fig. 2. It can be seen that in the medium and low SNR regions, the decoding error of multiple nodes is still not negligible. Therefore, we further extend NR-Fast-SSC-Flip decoder to multiple nodes per extra decoding attempt so as to compensate the loss of performance.

**Fig. 2.** Influence of polar code length and SNR on the number of error bits caused by channel noise

4.1 PROPOSED NR-FAST-SSC-FLIP- ω DECODER

Table 4. Update of the Flip-nodes Set Φ algorithm

Algorithm3: Update of the Flip-nodes Set Φ algorithm

for $t = \text{last}(\Phi(i)) + 1, \dots, \text{last}(A)$ and $t \in A$ **do**

$\Phi = \Phi(i) \cup \{t\};$

$m = M_\alpha(\Phi);$

if $m < M(\Phi_t)$ success **then**

$[\Phi, M_\Phi] \leftarrow \text{Insert}(\Phi, M_\Phi, \Phi, m);$

end if

end for

return $[\Phi, M_\Phi]$

In NR-Fast-SSC-Flip- ω decoder, the flip-node set Φ includes indexes of the flip-nodes. Each element Φ_k in Φ will be flipped in one additional decoding attempts. The metric value of one node can be extended to the set of multiple nodes $M'_\alpha(\Phi_\omega)$ by using the thought in D-SCF algorithm [10]

$$M'_\alpha(\Phi_\omega) = \sum_{k \in \Phi_\omega} \lambda[\Phi_{\omega-1}]_{k_\sigma} + \frac{1}{\alpha} \sum_{\substack{k \leq i_\omega \\ k \in A}} \log(1 + \exp(-\alpha \cdot \lambda[\Phi_{\omega-1}]_{k_\sigma})) \quad (23)$$

Where Φ_ω represents the set of size ω obtained by adding element i_ω to $\Phi_{\omega-1}$.

NR-Fast-SSC-Flip- ω algorithm is performed by combining NR-Fast-SSC-Flip with the dynamic update process as shown in Table 4. When CRC check fails, try to expand the order of the candidate node set through the dynamic update function.

4.2 The Optimization of parameter α

We use Monte-Carlo simulations to obtain the impact of code rate, code length, and SNR on parameter α . As shown in Fig. 3.1, we can see that the effect of code length is negligible. We use an approximate expression for the optimized $\bar{\alpha}_{opt}$ similar to [11]

$$\bar{\alpha}_{opt} = \min(a \cdot e^{f(R,\gamma)} + b \cdot e^{g(\gamma)} + c, 1) \quad (24)$$

Where parameter R is code rate, γ represents SNR. The model function by minimizing the fit errors as follows

$$\bar{\alpha}_{opt} = \min(0.5917 \cdot e^{0.0075\gamma+1.024R} - 0.4194 \cdot e^{0.2372\gamma+0.1829} + 0.2961, 1) \quad (25)$$

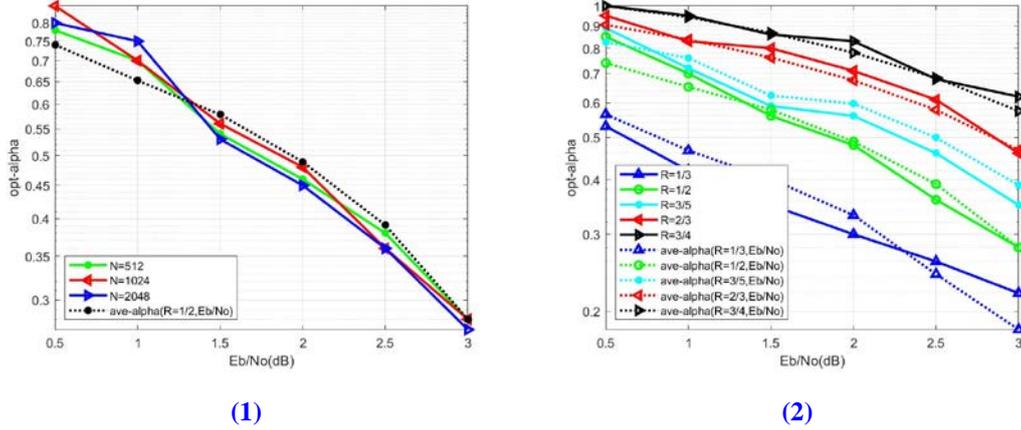


Fig. 3. The optimized parameter α_{opt} and $\bar{\alpha}_{opt}(R, \gamma)$ with different lengths N , rates R

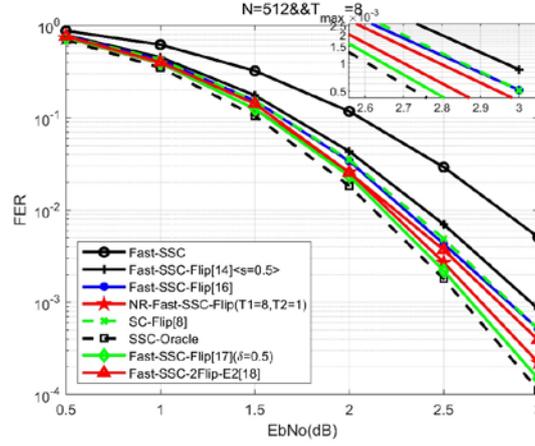
5. SIMULATION RESULTS AND ANALYSIS

In this part, we compare the FER performance between the proposed decoder and several existing Fast-SSC-Flip decoders. All simulation results are based on the BI-AWGN channel, the generator polynomial of CRC is $g(x) = x^{16} + x^{15} + x^2 + 1$. The information index set is obtained by Gaussian Approximation method proposed in [19]. The encoded method adopts the systematic polar code proposed in [20].

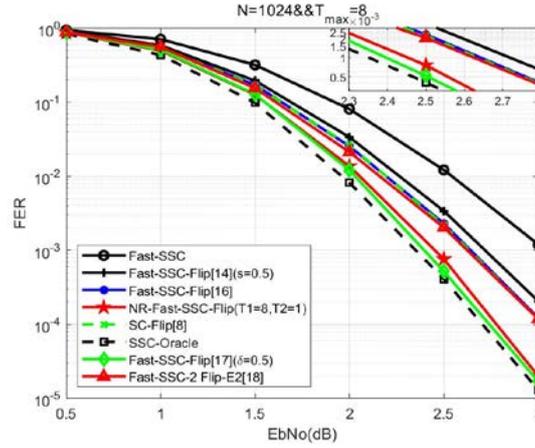
5.1 Performance Evaluation Of NR-Fast-SSC-Flip Decoder

5.1.1 FER Performance Analysis

We compared the FER of proposed NR-Fast-SSC-Flip decoder with existed Fast-SSC-Flip decoders, include Fast-SSC-Flip in [14], [16], Improved Fast-SSC-Flip in [17], Fast-SSC-2Flip-E2 in [18], PA-Fast-SSC-Flip($s=2$) [18]. For the fairness of comparisons, the parameter for all decoders is $R=\frac{1}{2}$, $P(1024, 496 + 16)$, and $P(512, 240 + 16)$. The maximum numbers of extra decoding attempts are $T_1 = 8$ and $T_2 = 1, 2$, $T_{max} = T_1 * T_2$.



(1)



(2)

Fig. 4. FER comparison for NR-Fast-SSC-Flip and contrastive decoding algorithms ($T_1=8, T_2=1$)

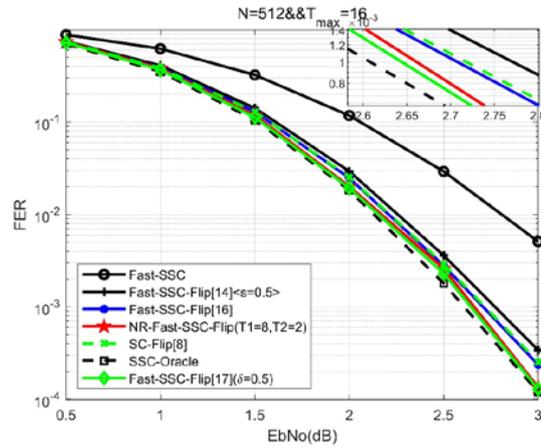
It can be seen from Fig. 4.1 that the performance of the proposed decoder is better than the existing Fast-SSC-Flip decoder exclude Improved Fast-SSC-Flip [17]. Compared with the Fast-SSC-Flip [14] with compensation coefficient $s = 0.5$, the proposed algorithm has 0.26dB performance gain when $FER=10^{-3}$. The proposed decoder can get 0.09 dB performance gain when $FER=10^{-3}$ compared with the Fast-SSC-2Flip-E2 decoder [18]. Compared with the Fast-SSC-Flip [16], the proposed algorithm achieves 0.14dB performance gain when $FER=10^{-3}$.

In Fig. 4.2, When we change the code length to $N=1024$, the proposed NR-Fast-SSC-Flip decoder can get 0.27dB performance gain compared with the Fast-SSC-Flip [14] and can obtain 0.17dB performance gain compared with the Fast-SSC-2Flip-E2 decoder [18] when $FER=10^{-3}$. Compared with the Fast-SSC-Flip [16], the proposed algorithm achieves 0.19dB performance gain when $FER=10^{-3}$.

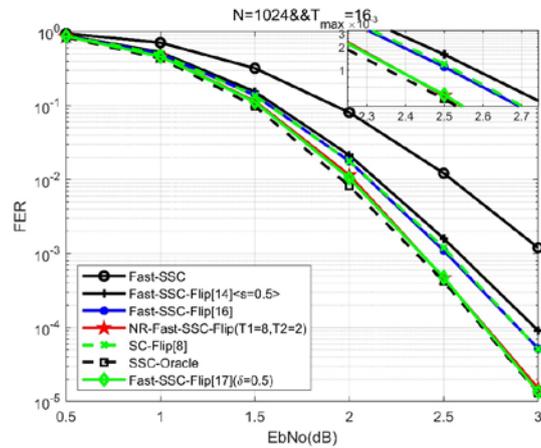
In Fig. 5, we set $T_2 = 2, T_{max} = 16$ and attempt to flip the second bit in Rate-1 and SPC nodes. Fig. 5 shows the proposed decoder has almost the same performance as Improved Fast-SSC-Flip [17]. This result shows that the main reason for the negligible performance loss in Fig. 4 is that the decision LLR values of two bits in the node are closer to each other in the

medium and high SNR region, and the bit with the smallest LLR is not the error bit introduced by channel noise, but we only select the bit with the smallest decision LLR in the node for flipping. It means the sub-optimal solution is selected.

In Fig. 5.2 we can see that the performance of the proposed decoder is close to the SSC-Oracle decoder, which is better than the traditional Fast-SSC-Flip decoder.



(1)



(2)

Fig. 5. FER comparison for NR-Fast-SSC-Flip and contrastive decoding algorithms ($T_1=8$, $T_2=2$)

5.1.2 Complexity Analysis

Compared with traditional Fast-SSC-Flip, the decoder we proposed selects CBs from unreliable nodes for flipping rather than from full information bits. So proposed decoder can effectively reduce complexity.

The complexity comparison between the proposed decoder and SC-Flip [8], Fast-SSC-Flip [14], Fast-SSC-Flip [17] and PA-Fast-SSC-Flip [18] is shown in the Table 5. Since all the flip algorithms are based on SC decoding, the complexity of basic SC decoding is $O(N \log N)$, which is same for all the flip algorithms. So, we only list the sorting and computational

complexity in **Table 5**.

The sorting complexity in [8] and [14] is $O(K \log K)$, where K is the number of information bits. The sorting complexity in [17] mainly includes two parts, one part within the SPC node is $O(|\tau_0| \log |\tau_0|)$, where $|\tau_0|$ represents the number of CBs in SPC nodes that contains the most CBs, and the other part is $O(K \log K)$. The sorting complexity in [18] is $O(S|\tau_1| \log |\tau_1|)$, where S is the number of partitions, $|\tau_1|$ is the number of CBs in the one partition that contains the most CBs.

The mainly sort complexity of the proposed NR-Fast-SSC-Flip decoder is $O(|A| \log |A|)$ to select flip-nodes and $O(|\tau| \log |\tau|)$ to select flipped bits within the node, where $|\tau|$ is the number of CBs in the special node that contains the most CBs.

The complexity of calculating CB reliability in [14] and [18] is $O(K\tau)$ and [17] is $O(K|A||\tau|)$, and the complexity of calculating node metric reliability of the proposed NR-Fast-SSC-Flip decoder is $O(|A|^2)$. Obviously, $|A| < K$ and $|\tau| < K$, so the overall complexity of the proposed decoder is lower than [17].

Table 5. The complexity of constructing flip-bits set comparison for various decoding algorithms

	Sorting	Metric Computation	Overall
SC-Flip [8]	$O(K \log K)$	-	$O(K \log K)$
Fast-SSC-Flip [14]	$O(K \log K)$	$O(K\tau)$	$O(K \log K + K\tau)$
Fast-SSC-Flip [17]	$O(K \log K) + O(\tau_0 \log \tau_0)$	$O(K A \tau)$	$O(K A \tau + K \log K)$
PA-Fast-SSC-Flip [18]	$O(S \tau_1 \log \tau_1)$	$O(K\tau)$	$O(S \tau_1 \log \tau_1 + K\tau)$
NR-Fast-SSC-Flip	$O(A \log A) + O(\tau \log \tau)$	$O(A ^2)$	$O(A ^2 + \tau \log \tau)$

5.2 Performance Evaluation Of NR-Fast-SSC-Flip- ω Decoder

NR-Fast-SSC-Flip- ω decoder allow multiple error-correction in one additional decoding process.

We can see from **Fig. 6** that the proposed NR-Fast-SSC-Flip- ω has 0.21dB performance gain compared to PA-SSC-Flip ($s=4$) [18] and 0.11dB gain compared to PA-Fast-SSC-Flip ($s=2$) [18] when $T_1 = 16$, $T_2 = 4$, $\omega = 2$ and $\text{FER}=10^{-4}$. It is clear that the proposed NR-Fast-SSC-Flip- ω has better error correction ability than Fast-SSC-Flip [17] in medium and high SNR area. This is because the proposed decoder can effectively correct multiple errors.

When we set $T_1 = 100$, $T_2 = 8$, $\omega = 3$, the proposed decoder can obtain 0.13dB performance gain when $\text{FER}=10^{-4}$ compared with the SSC-Oracle decoder from **Fig. 7**. It is observed from **Fig. 7** that the performance of NR-Fast-SSC-Flip- ω is better than CRC-SCL decoder with $L=4$.

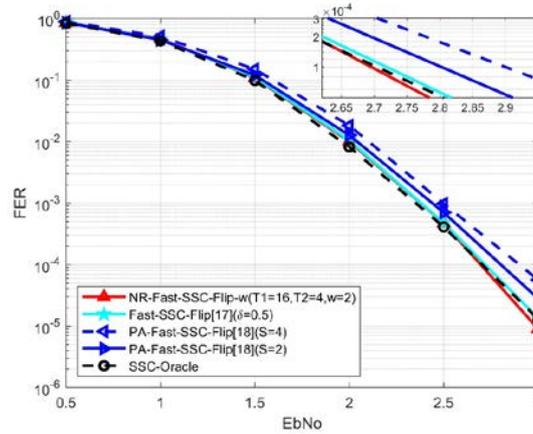


Fig. 6. FER comparison for NR-Fast-SSC-Flip- ω and contrastive decoding algorithms ($\omega = 2, N = 1024$)

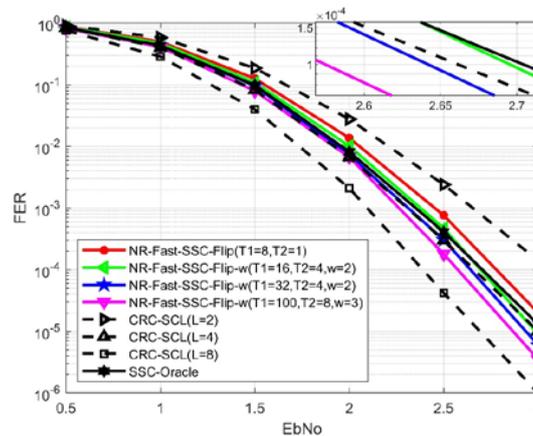


Fig. 7. FER performance of NR-Fast-SSC-Flip- ω decoder and CRC-SCL decoder ($N = 1024$)

6. Conclusion

In this work, we firstly put forward a criterion to measure the reliability of nodes. Under the guidance of this criterion, we proposed a NR-Fast-SSC-Flip decoder, which can locate the error bits more accurately than the traditional Fast-SSC-Flip decoder so as to improve the performance effectively. The dynamic update process was introduced to update the order of the flip-node set after each additional decoding failure. The NR-Fast-SSC-Flip- ω decoding algorithm was put forward to further improve the error-correction performance of the decoder. Simulation results show that the proposed decoder has better performance and lower complexity than Improved Fast-SSC-Flip decoder.

References

- [1] E. Arıkan, "Channel Polarization: A Method for Constructing Capacity-Achieving Codes for Symmetric Binary-Input Memoryless Channels," *IEEE Transactions on Information Theory*, vol. 55, no. 7, pp. 3051-3073, July 2009. [Article \(CrossRef Link\)](#)
- [2] Kong, D., Xia, X.G., Liu, P. et al., "MMSE channel estimation for two-port demodulation reference signals in new radio," *Sci. China Inf. Sci.*, 64, 169303, 2021. [Article \(CrossRef Link\)](#)
- [3] D. Kong, J. Li, K. Luo and T. Jiang, "Reducing Pilot Overhead: Channel Estimation With Symbol Repetition in MIMO-FBMC Systems," *IEEE Transactions on Communications*, vol. 68, no. 12, pp. 7634-7646, Dec. 2020. [Article \(CrossRef Link\)](#)
- [4] I. Tal and A. Vardy, "List Decoding of Polar Codes," *IEEE Transactions on Information Theory*, vol. 61, no. 5, pp. 2213-2226, May 2015. [Article \(CrossRef Link\)](#)
- [5] K. Niu and K. Chen, "CRC-Aided Decoding of Polar Codes," *IEEE Communications Letters*, vol. 16, no. 10, pp. 1668-1671, October 2012. [Article \(CrossRef Link\)](#)
- [6] K. Niu and K. Chen, "Stack decoding of polar codes," *Electronics Letters*, vol. 48, no. 12, pp. 695-697, 7 June 2012. [Article \(CrossRef Link\)](#)
- [7] L. Xiang, Z. B. Kaykac Egilmez, R. G. Maunder and L. Hanzo, "CRC-Aided Logarithmic Stack Decoding of Polar Codes for Ultra Reliable Low Latency Communication in 3GPP New Radio," *IEEE Access*, vol. 7, pp. 28559-28573, 2019. [Article \(CrossRef Link\)](#)
- [8] O. Afisiadis, A. Balatsoukas-Stimming and A. Burg, "A low-complexity improved successive cancellation decoder for polar codes," in *Proc. of 2014 48th Asilomar Conference on Signals, Systems and Computers*, pp. 2116-2120, 2014. [Article \(CrossRef Link\)](#)
- [9] 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. [Article \(CrossRef Link\)](#)
- [10] L. Chandesris, V. Savin and D. Declercq, "Dynamic-SCFlip Decoding of Polar Codes," *IEEE Transactions on Communications*, vol. 66, no. 6, pp. 2333-2345, June 2018. [Article \(CrossRef Link\)](#)
- [11] X. Zhang, Y. Liu and S. Chen, "BER Evaluation Based SCFlip Algorithm for Polar Codes Decoding," *IEEE Access*, vol. 8, pp. 3042-3054, 2019. [Article \(CrossRef Link\)](#)
- [12] G. Sarkis, P. Giard, A. Vardy, C. Thibeault and W. J. Gross, "Fast Polar Decoders: Algorithm and Implementation," *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 5, pp. 946-957, May 2014. [Article \(CrossRef Link\)](#)
- [13] A. Alamdar-Yazdi and F. R. Kschischang, "A Simplified Successive-Cancellation Decoder for Polar Codes," *IEEE Communications Letters*, vol. 15, no. 12, pp. 1378-1380, December 2011. [Article \(CrossRef Link\)](#)
- [14] P. Giard and A. Burg, "Fast-SSC-flip decoding of polar codes," in *Proc. of 2018 IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*, pp. 73-77, 2018. [Article \(CrossRef Link\)](#)
- [15] M. Hanif and M. Ardakani, "Fast Successive-Cancellation Decoding of Polar Codes: Identification and Decoding of New Nodes," *IEEE Communications Letters*, vol. 21, no. 11, pp. 2360-2363, Nov. 2017. [Article \(CrossRef Link\)](#)
- [16] Y. Zhou, J. Lin and Z. Wang, "A New Fast-SSC-Flip Decoding of Polar Codes," in *Proc. of ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*, pp. 1-6, 2019. [Article \(CrossRef Link\)](#)
- [17] Y. Zhou, J. Lin and Z. Wang, "Improved Fast-SSC-Flip Decoding of Polar Codes," *IEEE Communications Letters*, vol. 23, no. 6, pp. 950-953, June 2019. [Article \(CrossRef Link\)](#)
- [18] X. Wang, T. Wang, J. Li and Y. Zhang, "Improved Multiple Bit-Flipping Fast-SSC Decoding of Polar Codes," *IEEE Access*, vol. 8, pp. 27851-27860, 2020. [Article \(CrossRef Link\)](#)
- [19] I. Tal and A. Vardy, "How to Construct Polar Codes," *IEEE Transactions on Information Theory*, vol. 59, no. 10, pp. 6562-6582, Oct. 2013. [Article \(CrossRef Link\)](#)
- [20] G. Sarkis, I. Tal, P. Giard, A. Vardy, C. Thibeault and W. J. Gross, "Flexible and Low-Complexity Encoding and Decoding of Systematic Polar Codes," *IEEE Transactions on Communications*, vol. 64, no. 7, pp. 2732-2745, July 2016. [Article \(CrossRef Link\)](#)



Guo Rui received the Ph.D. degree from the Zhejiang University, Hangzhou, China, in 2007. He is currently an associate professor with the School of Communication Engineering, Hangzhou Dianzi University, Hangzhou, China. He is a visiting scholar at Oregon State University from August 2018 to August 2019. His research interests include wireless communication and channel coding.



Yang Pei is a Master student in Hangzhou Dianzi University. He received the B.E. degree in Communication Engineering from Huaibei Normal University, in 2019. His research interests include channel coding and wireless communication.



Ying Na received the Ph.D. degree from the Jilin University, Changchun, China, in 2006. She is currently an associate professor with the School of Communication Engineering, Hangzhou Dianzi University, Hangzhou, China. Her research interests include signal processing and its application.



Wang Lixin is currently an professor with the School of Communication Engineering, Hangzhou Dianzi University, Hangzhou, China. His research interests include wireless communication and signal processing.