Perturbation-Based Decoding Schemes for Long Polar Codes

Zhongjun Yang[†], Li Chen[†‡], Kangjian Qin[§], Xianbin Wang[§], and Huazi Zhang[§]
[†] School of Electronics and Information Technology, Sun Yat-sen University, Guangzhou 510006, P. R. China
[‡] Guangdong Province Key Laboratory of Information Security Technology, Guangzhou 510006, P. R. China
[§] Hangzhou Research Center, Huawei Technologies Co. Ltd., Hangzhou 310000, P. R. China
Email: yangzhj59@mail2.sysu.edu.cn, chenli55@mail.sysu.edu.cn, {qinkangjian1, wangxianbin1, zhanghuazi}@huawei.com

2025 IEEE International Symposium on Information Theory (ISIT) | 979-8-3315-4399-0/25/$31.00 ©2025 IEEE | DOI: 10.1109/ISIT63088.2025.11195489

***Abstract*—For polar codes, the bit-flipping strategy can significantly improve performance of its successive cancellation (SC) decoding. However, the gain derived from SC-flip (SCF) decoding diminishes as the codeword length increases. Addressing this issue, this paper proposes a novel hybrid perturbation-based SC (HPSC) decoding. If the initial SC decoding fails, the algorithm will generate multiple SC decoding attempts, each of which introduces stochastic perturbations to the received symbols. By soft information perturbations, the SC decoding can divert from the initial erroneous estimation and converge to the intended one. Our simulation results show that the proposed HPSC decoding consistently yields stable coding gains over various codeword lengths and rates. With the same number of decoding attempts, the HPSC decoding outperforms the thresholded SCF (TSCF) decoding. Moreover, it can achieve a similar performance as the cyclic redundancy check (CRC) aided SC list (CA-SCL) decoding, without any path sorting and expansion requirements.**

***Index Terms*—Hybrid perturbation scheme, long polar codes, soft information perturbation, successive cancellation decoding.**

I. Introduction

POLAR codes, invented by Arkan [1], are the first channel codes that can be proven to achieve Shannon capacity through the successive cancellation (SC) decoding. However, in the short-to-moderate codeword length regime, SC decoding exhibits poor performance. Addressing this issue, the SC list (SCL) decoding [2], [3] and the cyclic redundancy check (CRC) aided SCL (CA-SCL) decoding [4] were proposed. The above mentioned SCL-based decoding [2]–[8] demonstrate excellent frame error rate (FER) performance. However, when the codeword length is large, e.g., beyond ten thousand bits, these algorithms exhibit huge decoding computation and memory costs. These drawbacks limit the application of SCL-based decoders in decoding the long polar codes.

Addressing the above mentioned challenges, the SC-flip (SCF) decoding was proposed in [9]. It freezes the possibly erroneous bit to the opposite of its SC decoding estimations. Afterward, an additional SC decoding attempt is performed to re-estimate the information bits. This method works well if the flipped bit is the true first erroneous bit (FEB). However, constructing a flipping set (FS) that includes the true FEB is challenging. In [9]–[14], numerous metrics were proposed for the formation of the FS, including the ones based on the magnitude of decision log-likelihood ratios (LLRs) [9], and the error probability of bit estimation [10]. Since the critical set [15]–[17] is highly likely to contain the position of FEB in polar decoding algorithms, it is widely used as an offline FS. In [17], the thresholded SCF (TSCF) decoding was proposed. By utilizing the LLR-threshold, it can further reduce the searching space for possible erroneous positions within the critical set. Based on [10], the dynamic SCF (DSCF) decoding was proposed [18]. It can flip multiple erroneous bits in one decoding attempt. While the above mentioned SCF decoding algorithms perform well, the efficiency of bit-flipping vanishes rapidly as the codeword length increases, especially when the codeword length exceeds 4096. For long polar codes, it also becomes challenging to include the true FEB within a small position of unreliable bits.

To further improve the decoding efficiency of polar codes, the perturbation-enhanced decoding algorithm was proposed in [19], [20]. By adding an artificial noise to the received symbol, and trying a maximum of ten attempts, it can achieve the decoding performance of CA-SCL decoding with a list size of 16, using only a list size of eight. However, this method is less effective, if the perturbation is conducted on SC decoding.

To address this issue, this paper proposes a novel hybrid perturbation-based SC (HPSC) decoding. If the initial SC decoding fails, the algorithm will generate extra decoding attempts. During each attempt, subtle artificial perturbations are doped into the received symbols. By soft information perturbations, the SC decoding can divert from the initial erroneous estimation and converge to the intended one. Our numerical results show that with the same number of decoding attempts, the HPSC decoding outperforms the state-of-the-art TSCF decoding. Moreover, for polar codes with various codeword lengths and rates, the HPSC decoding consistently delivers robust performance improvements, while exhibiting a comparable FER performance to the CA-SCL decoding.

II. Preliminaries

This section provides the fundamental concepts of polar code and its SC decoding process. Given an indexed set $\mathcal{S}$ with cardinality $|\mathcal{S}|$, let $\mathcal{S}[i]$ denote the i-th element of $\mathcal{S}$. Let a_i^j denote the vector $(a_i, a_{i+1}, \ldots, a_j)$, $\hat{a}_i$ denote the estimation of a_i, and $\oplus$ denote the bitwise XOR operation.

A. Polar Codes

Polar codes originate from the channel combining and splitting, i.e., the channel polarization phenomenon [1], [21].

Let $\mathcal{P}(N,K)$ denote the polar code with length N and dimension K. After channel polarization, the K most reliable subchannels are chosen to transmit information bits. The remaining $N-K$ least reliable subchannels are utilized to transmit the redundancy, which is known as frozen bits. Let $\mathcal{A}$ and $\mathcal{A}^c$ denote the index sets of information bits and frozen bits, respectively, where $\mathcal{A}\cap\mathcal{A}^c=\emptyset$.

Given a $\mathcal{P}(N,K)$ polar code, the N-bit message vector u_0^{N-1} consists K information bits and $N-K$ frozen bits. Its codeword x_0^{N-1} is generated by

$$x_0^{N-1}=u_0^{N-1}\mathbf{G}_N, \tag{1}$$

where $\mathbf{G}_N=\left[\begin{smallmatrix}1&0\\1&1\end{smallmatrix}\right]^{\otimes n}$, $\otimes n$ denotes the n-th Kronecker power, and $N=2^n$. In this work, it is assumed that the codeword is transmitted through the additive white Gaussian noise (AWGN) channel, using binary phase shift keying (BPSK) modulation. Let $c_i=1-2x_i$ be the modulation. Let y_0^{N-1} denote the received symbol vector. That says

$$y_i=c_i+n_i, \tag{2}$$

where $n_i\sim\mathcal{N}(0,\sigma^2)$ and σ^2 is the noise power. In this work, the channel observations are represented and utilized as the received symbol LLRs. Let $\mathcal{L}(y_i)$ denote the converted LLR of received symbol y_i. Note that in the AWGN channel with BPSK modulation, it follows that $\mathcal{L}(y_i)=2y_i/\sigma^2$ [6].

B. SC Decoding

The SC decoding performs a greedy search and estimation of information bits over the binary tree with $n+1$ layers and N leaves. For information bit u_i, $i\in\mathcal{A}$, its estimation is determined by the decision LLR $\mathcal{L}(u_i)$. For frozen bits, they are fixed as zero, i.e., $\hat{u}_i=0$, $\forall i\in\mathcal{A}^c$ [1]. That says

$$\hat{u}_i=\begin{cases}1, & \text{if } \mathcal{L}(u_i)\le 0 \text{ and } i\in\mathcal{A};\\ 0, & \text{otherwise.}\end{cases} \tag{3}$$

Let us consider the SC decoder in decoding the polar code with $N=2$, which has the simplest tree structure. Initially, the algorithm performs the f-function to obtain the decision LLR of bit u_0. This process is based on the channel observations of the received symbols in y_0^1, i.e., $\mathcal{L}(y_0)$ and $\mathcal{L}(y_1)$. That says

$$\mathcal{L}(u_0)=\operatorname{sign}\left(\mathcal{L}(y_0)\cdot\mathcal{L}(y_1)\right)\cdot\min\left(|\mathcal{L}(y_0)|,|\mathcal{L}(y_1)|\right). \tag{4}$$

Once $\hat{u}_0$ is obtained, the SC decoder invokes the g-function to compute the decision LLR of u_1 as

$$\mathcal{L}(u_1)=(1-2\hat{u}_0)\cdot\mathcal{L}(y_0)+\mathcal{L}(y_1). \tag{5}$$

In particular, when obtaining $\hat{u}_0^1$, the estimations of codeword, i.e., $\hat{x}_0^1$, can be further computed by

$$\begin{cases}\hat{x}_0=\hat{u}_0\oplus\hat{u}_1;\\ \hat{x}_1=\hat{u}_1.\end{cases} \tag{6}$$

For polar codes with $N\ge 2$, SC decoding is performed by recursively invoking the f and the g functions [6], [19]. This process continues until the decision LLR of all information bits are obtained, i.e., all leaves have been traversed.

Algorithm 1: The RPSC Decoding

Input: $\mathcal{A}$, $T_{\max}$, σ_p^2, y_0^{N-1};
Output: $\hat{u}_0^{N-1}$;
$\hat{u}_0^{N-1}\leftarrow SC(\mathcal{A}, y_0^{N-1})$; // Original SC decoding
If $CRC(\hat{u}_0^{N-1})=true$ **then**
  Return $\hat{u}_0^{N-1}$; // Decoding successful
For $t\leftarrow 0$ **to** $T_{\max}-1$ **do**
  Compute ${y'}_0^{N-1}$ as in (7);
  $\hat{u}_0^{N-1}\leftarrow SC(\mathcal{A}, {y'}_0^{N-1})$; // Re-decoding
  If $CRC(\hat{u}_0^{N-1})=true$ **then**
    Return $\hat{u}_0^{N-1}$; // Decoding successful
Return $\hat{u}_0^{N-1}$; // Decoding failed

III. The Perturbation-Based Decoding Schemes

This section introduces the perturbation-based SC decoding schemes, which can be considered particularly useful for long polar codes. They include the random perturbation-based SC (RPSC) decoding and the biased perturbation-based SC (BPSC) decoding. Among them, the BPSC decoding can efficiently correct the failed RPSC decoding estimations.

A. Random Perturbation-Based SC Decoding

Introducing perturbation or noise to received symbols so as to enhance the belief propagation (BP) decoding performance has been proposed by [22]. It is shown in [23]–[25] that, through perturbing (or doping) the *a prior* LLRs of the least reliable bits, the BP decoding output can re-converge to the correct codeword (or the bit estimations). Inspired by this, this work first proposes the RPSC decoding for polar codes.

In this paper, the CRC code is utilized for identifying the erroneous SC decoding estimations. Let $CRC(\cdot)$ denote the CRC function. In the RPSC decoding, the original SC decoding is first performed. It then invokes the CRC to validate the estimation $\hat{u}_0^{N-1}$. If the validation fails, denoted as $CRC(\cdot)=false$, the algorithm performs a prespecified number of extra SC decoding attempts. During each attempt, the received symbol y_i is perturbed by doping a random perturbation $n_i'\sim\mathcal{N}(0,\sigma_p^2)$, where σ_p^2 denotes the perturbation power. Let y_i' denote the perturbed version of y_i. Based on (2), y_i' can be computed as

$$y_i'=y_i+n_i'. \tag{7}$$

Afterward, based on the perturbed ${y'}_0^{N-1}$, SC decoding is performed to obtain the new estimations of u_0^{N-1}. This decoding process terminates only when the correct estimations $\hat{u}_0^{N-1}$ is obtained, denoted as $CRC(\cdot)=true$, or all the perturbation decoding attempts have been exhausted. Let $T_{\max}$ denote the maximum number of predefined decoding attempts. Let $SC(\cdot)$ denote the underlying SC decoding with inputs $\mathcal{A}$ and y_0^{N-1}. The RPSC decoding is summarized as in Algorithm 1.

Fig. 1 shows performance of the RPSC decoding for $T_{\max}\in\{10,30,50,100\}$, where $N=4096$, $K=2048$ and $\sigma_p^2=0.053$. It can be observed that little performance gain can be achieved when $T_{\max}\ge 30$. This is because, as a depth-first (sub-optimal) decoding algorithm, SC decoding is prone to get

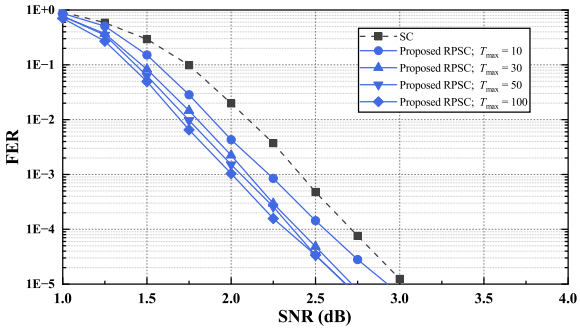


Fig. 1. RPSC decoding performance with different perturbation attempts $T_{\max} \in \{10, 30, 50, 100\}$, where $N = 4096$ and $K = 2048$.

trapped in the local optimal state with the same bit estimations [1], [6]. Note that the proposed RPSC decoding performs the underlying SC decoding recursively, and y'^{N-1}_0 is derived from y^{N-1}_0, as in (7). During different decoding attempts, the RPSC decoding might repeatedly output the same $\hat{u}^{N-1}_0$. This drawback limits the performance of RPSC algorithm.

B. Biased Perturbation-Based SC Decoding

In order to address the above mentioned RPSC decoding limit, this subsection proposes the BPSC decoding. It aims to align the estimation of information bits (or the codeword) with that of the maximum-likelihood (ML) decoding by introducing biased perturbations. The following Lemma 1 characterizes the ML decoding of polar codes.

Lemma 1. Consider a $\mathcal{P}(N, K)$ polar code with a codebook $\mathbb{C}$, the ML decoder yields the codeword that satisfies

$$\hat{x}^{N-1}_0 = \underset{x^{N-1}_0 \in \mathbb{C}}{\arg\max} \sum_{i=0}^{N-1} (1 - 2x_i)\, y_i. \tag{8}$$

Proof: The research work of [6], [26] shows that an ML decoder yields the estimation $\hat{x}^{N-1}_0$ satisfying

$$\hat{x}^{N-1}_0 = \underset{x^{N-1}_0 \in \mathbb{C}}{\arg\max} \sum_{i=0}^{N-1} \ln P\left(y_i \mid x_i\right), \tag{9}$$

where $P\left(y_i \mid x_i\right)$ denotes channel transition probability.

Given y^{N-1}_0, $\sum_{i=0}^{N-1} \ln P\left(y_i \mid 1\right)$ can be determined [26]. Thus, we have

$$\begin{aligned} \hat{x}^{N-1}_0 &= \underset{x^{N-1}_0 \in \mathbb{C}}{\arg\max} \left(\sum_{i=0}^{N-1} \ln P\left(y_i \mid x_i\right) - \sum_{i=0}^{N-1} \ln P\left(y_i \mid 1\right) \right) \\ &= \underset{x^{N-1}_0 \in \mathbb{C}}{\arg\max} \sum_{i=0}^{N-1} (1 - x_i) \mathcal{L}\left(y_i\right). \end{aligned} \tag{10}$$

Then, (10) can be further simplified as

$$\hat{x}^{N-1}_0 = \underset{x^{N-1}_0 \in \mathbb{C}}{\arg\max} \left(\frac{1}{2} \sum_{i=0}^{N-1} \mathcal{L}\left(y_i\right) + \frac{1}{2} \sum_{i=0}^{N-1} (1 - 2x_i) \mathcal{L}\left(y_i\right) \right). \tag{11}$$

Note that, given y^{N-1}_0, $\sum_{i=0}^{N-1} \mathcal{L}\left(y_i\right)/2$ can also be determined. Thus, (11) can be reformulated as

$$\hat{x}^{N-1}_0 = \underset{x^{N-1}_0 \in \mathbb{C}}{\arg\max} \sum_{i=0}^{N-1} (1 - 2x_i) \mathcal{L}\left(y_i\right). \tag{12}$$

Since $\mathcal{L}\left(y_i\right) = 2y_i/\sigma^2$ and $\sigma^2 \geq 0$, we have

$$\hat{x}^{N-1}_0 = \underset{x^{N-1}_0 \in \mathbb{C}}{\arg\max} \sum_{i=0}^{N-1} (1 - 2x_i) y_i. \tag{13}$$

□

Lemma 1 implies that for the ML decoding, the estimation $\hat{x}^{N-1}_0$ tends to align with the hard decisions of y^{N-1}_0. In this case, it follows that $1 - 2\hat{x}_i = \operatorname{sign}(y_i)$, where $\operatorname{sign}(\cdot)$ denotes the sign function. Let

$$\Lambda(\hat{x}^{N-1}_0) = \sum_{i=0}^{N-1} (1 - 2\hat{x}_i)\, y_i. \tag{14}$$

The likelihood of estimations between the ML and the SC-based decoding can be efficiently measured by $\Lambda(\hat{x}^{N-1}_0)$. Therefore, it can be conjectured that for the RPSC decoding failure, trapped in the local optimal state (as mentioned in Sec. III-A), the estimation $\hat{x}^{N-1}_0$ corresponds to a small $\Lambda(\hat{x}^{N-1}_0)$. Moreover, it is shown in [23], [24], [27] that correcting the erroneous symbols y_i or enhancing the reliable ones both can enhance the performance of polar decoders. Hence, this paper proposes the BPSC decoding to further facilitate the RPSC decoding.

Let us consider the estimation $\hat{x}_i$ disagrees with the hard decision of y_i, i.e., $1 - 2\hat{x}_i \neq \operatorname{sign}(y_i)$. They are also known as the disagree bit (DB). Let $\mathcal{D}$ denote the index set of all DBs between $\hat{x}^{N-1}_0$ and y^{N-1}_0, defined as

$$\mathcal{D} = \{i \in \{0, 1, \ldots, N-1\} \mid \operatorname{sign}(y_i) \neq (1 - 2\hat{x}_i)\}. \tag{15}$$

Note that in $\mathcal{D}$, for any $i < j$, there exists $|y_{\mathcal{D}[i]}| \geq |y_{\mathcal{D}[j]}|$.

Lemma 1 implies that the DBs result in a smaller $\Lambda(\hat{x}^{N-1}_0)$. This makes the estimations of RPSC decoding deviate from those of the ML decoding. To rectify this, during the BPSC decoding, only the unreliable received symbols identified by the set $\mathcal{D}$ are doped with biased perturbations. Similar to (7), the biasedly perturbed y'_i is defined as

$$y'_i = (1 - 2\hat{x}_i) \cdot \varepsilon, \;\; i \in \mathcal{D}, \tag{16}$$

where $\hat{x}_i$ is the estimation made upon the original SC decoding, and ε denotes the magnitude of the biased perturbation.

In the BPSC decoding, the set $\mathcal{D}$ is first constructed based on $\hat{x}^{N-1}_0$ and y^{N-1}_0. In particular, based on Lemma 1, the received symbol y_i, $i \in \mathcal{D}$, with a larger magnitude is given priority to be biasedly perturbed. During each decoding attempt, the algorithm is performed on one of the positions in $\mathcal{D}$. For instance, in the t-th attempt, the received symbol y_{i^*} is perturbed with the value of $(1 - 2\hat{x}_{i^*})\,\varepsilon$, where $i^* \leftarrow \mathcal{D}[t]$. For other symbols, they are randomly perturbed as in (7). After that, the perturbed y'^{N-1}_0 is re-decoded through the underlying SC decoding. Let $T^{\mathrm{B}}_{\max}$ denote the maximum number of decoding attempts for the BPSC decoding. Once all perturbation attempts have been

Algorithm 2: The BPSC Decoding

Subroutine *BiasedPSC*($\mathcal{A}$, $T_{\max}^{\mathrm{B}}$, σ_p^2, ε, y_0^{N-1}, $\hat{x}_0^{N-1}$)**:**
  Initialize: $\mathcal{D} \leftarrow \emptyset$;
  Construct set $\mathcal{D}$ as in (15);
  For $t \leftarrow 0$ **to** $T_{\max}^{\mathrm{B}} - 1$ **do**
    $i^* \leftarrow \mathcal{D}[t]$; // Obtaining position
    $y'_{i^*} \leftarrow (1 - 2\hat{x}_{i^*}) \cdot \varepsilon$; // Resetting value
    For $i \in \{0, 1, \ldots, N-1\} \setminus \{i^*\}$ **do**
      Compute y'_i as in (7);
    $\hat{u}_0^{N-1} \leftarrow SC(\mathcal{A}, {y'}_0^{N-1})$; // Re-decoding
    If $CRC(\hat{u}_0^{N-1}) = true$ **then**
      Return $\hat{u}_0^{N-1}$; // Decoding successful
  Return $\hat{u}_0^{N-1}$; // Decoding failed

exhausted, i.e., $t = T_{\max}^{\mathrm{B}}$, or the correct estimations is obtained, i.e., $CRC(\cdot) = true$, the decoding process is terminated. The BPSC decoding is summarized as in Algorithm 2.

C. Hybrid Perturbation-Based SC Decoding

In order to further improve the SC decoding performance, this paper proposes a novel hybrid perturbation-based SC (HPSC) decoding. It contains a two-level scheme that integrates both the RPSC and the BPSC decoding. At first, the first-level scheme, i.e., the RPSC decoding, is performed. When the RPSC decoding is trapped in the local optimal state, the second-level scheme, i.e., the BPSC decoding, will be invoked. By the use of the above mentioned schemes, the perturbation-based SC decoding can efficiently escape from the local optimal state that containing erroneous estimations.

For the HPSC decoding, challenges lie in accurately identifying the decoding process trapped in the state of local optimal. For this, the CRC codeword, generated by the CRC function, is used. The CRC codes serve for two purposes. The primary one is used as a detector for selecting the valid bit estimations from multiple decoding candidates. On the other hand, it can also help identify the local optima state during the RPSC decoding process. When the CRC validation fails, the HPSC decoding can assess whether the RPSC decoding estimations falls into a local optimal state based on the obtained CRC codeword. As mentioned in Sec. III-A, in this case, the RPSC decoding has been outputting the same $\hat{u}_0^{N-1}$. If the CRC codewords between two consecutive decoding attempts are identical, the state of local optima has been successfully identified. Thus, the BPSC decoding is invoked; otherwise, the algorithm performs the next decoding attempts.

Let m denote the CRC codeword length, and $s_0^{m-1}(t)$ denote the CRC codeword in the t-th decoding attempt. The proposed HPSC algorithm is summarized as in Algorithm 3. In particular, all the above mentioned decoding schemes, i.e., the RPSC, the BPSC, and the HPSC algorithms, can be further enhanced by the use of other underlying decoding algorithms, such as the SCL decoding [2]–[8] and the BP decoding [25].

IV. Simulation Results

This section presents the numerical results of the proposed perturbation-enhanced decoding algorithms. The length-16 CRC code of the 5G standard [28] is employed with $m = 16$. For the $\mathcal{P}(N, K)$ code, its rate is $R = (K - m)/N$. Polar codes with $N \in \{1024, 4096, 16384\}$ and $R \in \{1/3, 1/2, 2/3\}$ are considered in our simulation. The information set $\mathcal{A}$ is obtained through Gaussian approximation (GA) [29] at the SNR of 2.5 dB. Moreover, it is set that $\varepsilon = 1$ and $T_{\max} = 15$. Note that the optimized $(\sigma_p^2, \varepsilon)$ depend on the code and the channel signal-to-noise ratio (SNR) regimes, which can be determined through Monte-Carlo simulations [19].

Algorithm 3: The HPSC Decoding

Input: $\mathcal{A}$, $T_{\max}$, σ_p^2, ε, y_0^{N-1};
Output: $\hat{u}_0^{N-1}$;
Initialize: $s_0^{m-1}(-1) \leftarrow \emptyset$;
$\{\hat{u}_0^{N-1}, \hat{x}_0^{N-1}\} \leftarrow SC(\mathcal{A}, y_0^{N-1})$; // Original SC
If $CRC(\hat{u}_0^{N-1}) = true$ **then**
  Return $\hat{u}_0^{N-1}$; // Decoding successful
For $t \leftarrow 0$ **to** $T_{\max} - 1$ **do**
  // First-level: RPSC decoding
  Compute ${y'}_0^{N-1}$ as in (7);
  $\hat{u}_0^{N-1} \leftarrow SC(\mathcal{A}, {y'}_0^{N-1})$; // Re-decoding
  If $CRC(\hat{u}_0^{N-1}) = true$ **then**
    Return $\hat{u}_0^{N-1}$; // Decoding successful
  $s_0^{m-1}(t) \leftarrow CRC(\hat{u}_0^{N-1})$; // Storing codeword
  // Second-level: BPSC decoding
  If $s_0^{m-1}(t-1) = s_0^{m-1}(t)$ **then**
    $\hat{u}_0^{N-1} \leftarrow BiasedPSC(\mathcal{A}, T_{\max} - t - 1, \sigma_p^2, \varepsilon, y_0^{N-1}, \hat{x}_0^{N-1})$;
    Return $\hat{u}_0^{N-1}$; // Decoding terminated
Return $\hat{u}_0^{N-1}$; // Decoding failed

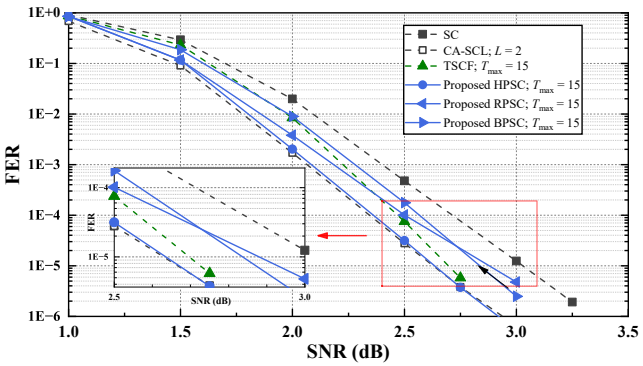
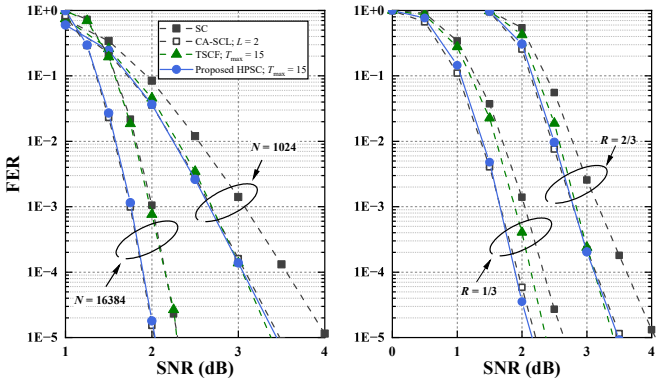


Fig. 2. Decoding FER performance for the $\mathcal{P}(4096, 2048 + 16)$ code.

A. Decoding Performance

Fig. 2 shows performance of the proposed perturbation-based decoding algorithms for the $\mathcal{P}(4096, 2048 + 16)$ code. It can be observed that compared with the conventional SC decoding, all three proposed algorithms can achieve significant performance gains. In particular, performance of the HPSC decoding is superior to that of the RPSC decoding and that of the BPSC decoding. Therefore, the following analyses will focus on the HPSC decoding. As illustrated in Fig. 2, the proposed HPSC decoding outperforms the state-of-the-art TSCF decoding in [17]. Moreover, it can achieve a similar



(a) $N \in \{1024, 16384\}$ and $R = 1/2$ (b) $N = 4096$ and $R \in \{1/3, 2/3\}$

Fig. 3. Decoding performance of the HPSC decoding for long polar codes. (a) various codeword lengths $N \in \{1024, 16384\}$ and $R = 1/2$; (b) various rates $R \in \{1/3, 2/3\}$ and $N = 4096$.

error-correction performance as the CA-SCL decoding with the list size of two, i.e., $L = 2$.

Fig. 3 shows performances of the HPSC decoding for various codeword lengths and rates. Fig. 3(a) shows decoding performance of the proposed HPSC algorithm for long polar codes, where $N \in \{1024, 16384\}$ and $R = 1/2$. It can be observed that, compared with the TSCF decoding, the HPSC decoding still yields better performances in the longer codeword length regime, e.g., $N \in \{4096, 16384\}$. This is because, for the SCF-based decoding, only when the FEB has been corrected, its FER performance will be improved. However, for long polar codes, it becomes challenging to accurately correct the true FEB with a small number of flipping attempts, e.g., $T_{\max} \leq 15$. In contrast, the HPSC decoding perturbs the received symbols y_0^{N-1} without the need for identifying the erroneous $\hat{u}_i$. Fig. 3(b) shows the FER performance of HPSC decoding for various rates $R \in \{1/3, 2/3\}$, where $N = 4096$. It can be seen that the HPSC decoding achieves performance comparable to the CA-SCL decoding ($L = 2$) over all rates. Unlike the CA-SCL decoding, it does not need to perform path sorting. Based on Figs. 2 and 3, it can be concluded that, compared with the bit-flipping strategy, the proposed perturbation-based decoding scheme, i.e., the HPSC decoding, can provide robust performance improvements under various lengths and rates when decoding the long polar codes.

B. Decoding Complexity and Latency

For the proposed HPSC decoding, computational complexity is composed by the complexity of the underlying SC decoding, and that of constructing the DB set $\mathcal{D}$. According to [13], the complexity of selecting $T_{\max}$ elements from an ordered set $\mathcal{D}$ is $T_{\max}(|\mathcal{D}| - 1)$. Let T_{avg} denote the average number of perturbation attempts in the HPSC decoding. The average complexity of HPSC decoding is

$$\Phi_{\text{HPSC}} = T_{\text{avg}} \cdot N \log_2(N) + T_{\max} \cdot (|\mathcal{D}| - 1), \tag{17}$$

where $|\mathcal{D}| \in [T_{\max}, N]$, and $\mathcal{O}(N \log_2 N)$ denotes the complexity of SC decoding [1], [30]. Moreover, in the worst-case, complexity of the HPSC decoding is $\mathcal{O}(T_{\max} N \log_2 N)$, i.e., $T_{\max}$ times that of the SC decoding.

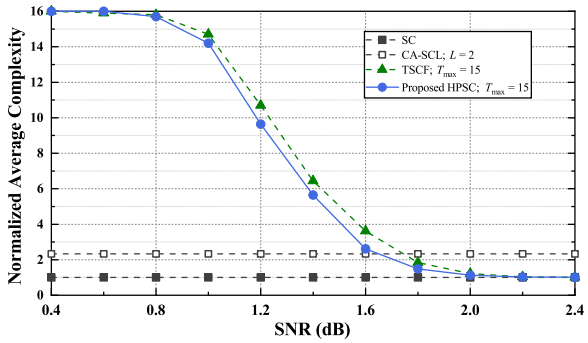


Fig. 4. Normalized complexity in decoding the $\mathcal{P}(4096, 2048 + 16)$ code.

TABLE I
AVERAGE LATENCY IN DECODING THE $\mathcal{P}(4096, 2048 + 16)$ CODE

Polar Decoder	$T_{\max}$	SNR (dB)			
		1	1.5	2	2.5
SC [30]	–	8,190	8,190	8,190	8,190
CA-SCL [4]	$L = 2$	10,238	10,238	10,238	10,238
TSCF [17]	15	117,138	40,359	9,828	8,222
Proposed HPSC	15	114,609	30,187	9,235	8,222

Fig. 4 shows the normalized average complexity in decoding the $\mathcal{P}(4096, 2048 + 16)$ code, where the normalization factor is $N \log_2 N$. For the CA-SCL decoding, the computational complexity of path sorting is $2KL\log_2(2L)$ [6]. As illustrated in Fig. 4, the complexity of HPSC decoding is still lower than that of the TSCF decoding. In the high SNR regime (e.g., SNR > 2 dB), the complexity of HPSC decoding reduces to that of the SC decoding. Compared with the CA-SCL decoding ($L = 2$), the HPSC decoding can reduce complexity by more than half, but without degrading the decoding performance.

In this paper, decoding latency is measured by the number of required clock cycles (CCs) [8], [30]. The SC decoding requires $2N - 2$ CCs [30]. For the proposed HPSC decoding, in each perturbation decoding attempt, it requires one CC to perform the CRC validation (lines 8 to 10 in Algorithm 3). Moreover, for the process of generating set $\mathcal{D}$, it requires an additional CC. Thus, the average latency of HPSC decoding is defined as

$$\bar{\tau}_{\text{HPSC}} = T_{\text{avg}} \cdot (2N - 1) + 1. \tag{18}$$

Note that when $T_{\text{avg}} = T_{\max} + 1$, $\bar{\tau}_{\text{HPSC}}$ denotes the decoding latency in the worst-case.

Table I shows the average latency in decoding the $\mathcal{P}(4096, 2048+16)$ code. It can be noticed that compared with the TSCF decoding, the HPSC decoding requires fewer CCs, while maintaining a better FER performance. While compared with the CA-SCL decoding [4], the HPSC decoding yields a significant decoding latency advantage. E.g., when SNR = 2.5 dB, the CA-SCL decoding ($L = 2$) requires 10,238 CCs. In comparison, the HPSC decoding ($T_{\max} = 15$) requires only 8,222 CCs, reaching a CC reduction of 20%.

ACKNOWLEDGEMENT

This work is supported in part by the National Natural Science Foundation of China (NSFC) with project ID 62471503; and in part by the Natural Science Foundation of Guangdong Province (NSFGP) with project ID 2024A1515010213.

REFERENCES

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

[3] K. Chen, K. Niu, and J. Lin, "List successive cancellation decoding of polar codes," *Electron. Lett.*, vol. 48, no. 9, pp. 500–501, Apr. 2012.

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

[5] 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.

[6] A. Balatsoukas-Stimming, M. Bastani 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.

[7] M. C. Coskun and H. D. Pfister, "An information-theoretic perspective on successive cancellation list decoding and polar code design," *IEEE Trans. Inf. Theory*, vol. 68, no. 9, pp. 5779–5791, Sep. 2022.

[8] S. A.Hashemi, C. Condo, and W. J. Gross, "Fast and flexible successive-cancellation list decoders for polar codes," *IEEE Trans. Signal Process.*, vol. 65, no. 21, pp. 5756–5769, Nov. 2017.

[9] O. Afisiadis, A. Balatsoukas-Stimming, and A. Burg, "A low-complexity improved successive cancellation decoder for polar codes," in *Proc. 2014 48th Asilomar Conf. Signals, Syst. Comput. (ACSSC)*, Pacific Grove, USA, Nov. 2014.

[10] L. Chandesris, V. Savin, and D. Declercq, "An improved SCFlip decoder for polar codes," in *Proc. 2016 IEEE Glob. Commun. Conf. (GLOBECOM)*, Washington, USA, Dec. 2016.

[11] Y. Shen, W. Song, Y. Ren, H. Ji, X. You, and C. Zhang, "Enhanced belief propagation decoder for 5G polar codes with bit-flipping," *IEEE Trans. Circuits Syst. II*, vol. 67, no. 5, pp. 901–905, May 2020.

[12] Y. Shen, W. Song, H. Ji, Y. Ren, C. Ji, X. You, and C. Zhang, "Improved belief propagation polar decoders with bit-flipping algorithms," *IEEE Trans. Commun.*, vol. 68, no. 11, pp. 6699–6713, Nov. 2020.

[13] Z. Yang and L. Chen, "An enhanced belief propagation decoding algorithm with bit-flipping for polar codes," *IEEE Commun. Lett.*, vol. 29, no. 2, pp. 348–352, Feb. 2025.

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

[15] Z. Zhang, K. Qin, L. Zhang, H. Zhang, and G. T. Chen, "Progressive bit-flipping decoding of polar codes over layered critical sets," in *Proc. 2017 IEEE Glob. Commun. Conf. (GLOBECOM)*, Singapore, Dec. 2017.

[16] M. Rowshan and E. Viterbo, "Improved list decoding of polar codes by shifted-pruning," in *Proc. 2019 IEEE Inf. Theory Workshop (ITW)*, Visby, Sweden, Aug. 2019.

[17] 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.

[18] 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.

[19] X. Wang, H. Zhang, J. Tong, J. Wang, J. Ma, and W. Tong, "Perturbation-enhanced SCL decoder for polar codes," in *Proc. 2023 IEEE Globecom Workshops (GC Wkshps)*, Kuala Lumpur, Malaysia, Dec. 2023.

[20] X. Wang, H. Zhang, J. Tong, J. Wang, and W. Tong, "Adaptive perturbation enhanced SCL decoder for polar codes," *arXiv:2407.03555*, Jul. 2024.

[21] E. Arıkan, "Channel combining and splitting for cutoff rate improvement," *IEEE Trans. Inf. Theory*, vol. 52, no. 2, pp. 628–639, Feb. 2006.

[22] A. C. Arli and O. Gazi, "Noise-aided belief propagation list decoding of polar codes," *IEEE Commun. Lett.*, vol. 23, no. 8, pp. 1285–1288, Aug. 2019.

[23] B. Feng, R. Liu, and K. Tian, "A novel post-processing method for belief propagation list decoding of polar codes," *IEEE Commun. Lett.*, vol. 25, no. 8, pp. 2468–2471, Aug. 2021.

[24] S. Sun, S.-G. Cho, and Z. Zhang, "Post-processing methods for improving coding gain in belief propagation decoding of polar codes," in *Proc. 2017 IEEE Glob. Commun. Conf. (GLOBECOM)*, Singapore, Dec. 2017.

[25] Z. Yang, Z. Cai, L. Chen, and H. Zhang, "An efficient adaptive belief propagation decoder for polar codes," in *Proc. 2024 IEEE Inf. Theory Workshop (ITW)*, Shenzhen, China, Nov. 2024.

[26] 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.

[27] M. Zhang, Z. Li, and L. Xing, "An enhanced belief propagation decoder for polar codes," *IEEE Commun. Lett.*, vol. 25, no. 10, pp. 3161–3165, Oct. 2021.

[28] "Technical specification group radio access network," 3GPP, TS 38.212 version 15.2.0, Jul. 2018. [Online]. Available: https://www.3gpp.org/ftp/Specs/archive/38_series/38.212/

[29] P. Trifonov, "Efficient design and decoding of polar codes," *IEEE Trans. Commun.*, vol. 60, no. 11, pp. 3221–3227, Nov. 2012.

[30] C. Leroux, A. J. Raymond, G. Sarkis, and W. J. Gross, "A semi-parallel successive-cancellation decoder for polar codes," *IEEE Trans. Signal Process.*, vol. 61, no. 2, pp. 289–299, Jan. 2013.