# Generalized Simplified Successive-Cancellation Decoding of Multi-kernel Polar Codes

Yanlong Zhao, Zhilu Wu, Zhendong Yin(✉), and Qingzhi Liu

School of Electronics and Information Engineering, Harbin Institute of Technology, Harbin, China
{16B905027,17B905015}@stu.hit.edu.cn, {wuzhilu,yinzhendong}@hit.edu.cn

**Abstract.** Multi-Kernel (MK) offers more flexibility code length selections for polar codes compared to size-2 kernel proposed by Arikan. In this paper, a generalized Simplified Successive-Cancellation List (SSCL) decoding algorithm is introduced. We first provide sufficient conditions and corresponding proofs to perform SSC and SSCL decoding on MK polar codes. These simplifications are proven valid for MK polar codes whose transform matrix are constructed by kernels who satisfy certain conditions. Time-complexity reduction of introducing generalized simplification decoding is discussed. Numerical results shows that our proposed methods can reduce time-step while preserving the error-correction performance.

**Keywords:** Polar codes · Successive-cancellation · List decoding · Multi-kernel

## 1 Introduction

Polar codes are the first error-correcting codes introduced by Arikan in [1] who are proven to be able to reach channel capacity for a large number of channels. In recent years, polar codes were adopted as channel coding for uplink and downlink control information for the enhanced mobile broadband communication service of 5G. Arikan proposes a $2 \times 2$ kernel $\boldsymbol{T}_2 = \left(\begin{smallmatrix} 1 & 0 \\ 1 & 1 \end{smallmatrix}\right)$ as transform matrix to construct polar codes, which only allows for code lengths power of 2. This constraint limits polar codes for typical 5G applications. Therefore, puncturing, shortening [2] and extending [3] are studied for polar codes 5G typical scenarios.

Authors in [4] exhibit that other kernels except $\boldsymbol{T}_2$ could also achieve polarization phenomenon, which means code length of polar codes is no longer limit to power of 2. This coding method allows to conjunct different size of polarizing matrices to build a polar code with more flexibility in code length. MK polar codes are introduced in [5] which provide a practical approach for constructing more flexible polar codes. To improve the error-correction performance,


Q. Guo et al. (Eds.): WiSATS 2021, LNICST 410, pp. 378–391, 2022.
https://doi.org/10.1007/978-3-030-93398-2_37

minimum-distance profile is proposed in [6]. Then, a hybrid design combining reliability and distance properties is further explored in [7]. [8] also designs a kernel substitution coding method which further improves the performance of MK polar codes.

Successive-Cancellation (SC) decoding is first used to decode polar codes when they were proposed. However, affecting by the serial nature, SC decoding suffers great latency and undesirable throughput. To overcome this drawback, Simplified Successive-Cancellation (SSC) has been proposed in [9] who introduced parallel decoder to SC decoding. And then, more special nodes have been raised in [10] and [11] etc. Recently, [12] provides a generalized approach to further reduce SC-based decoding latency. In order to fill the gap between the performance of SC and maximum-likelihood decoding, Successive-Cancellation List (SCL) proposed in [13] offers performance competitive with many other channel codes. The simplification algorithms of SCL are consecutively studied in [14] and [15]. Due to the coding structure, SC decoding is also performed for multi-kernel polar codes when they were proposed in [4]. Then, [16] explores general procedure to marginalize kernels of any size. Further, fast simplified SC decoding was also be presented in [17] to decode ternary kernel based polar codes.

This paper explores a generalized approach to SSC and SSCL decoding of MK polar codes to further reduce computation complexity. We provide sufficient conditions for kernels who are suitable for simplification calculations of SC and SCL decoding of multi-kernel polar codes. Then, the relative conditions are verified by deriving $\boldsymbol{T}_3$ kernel who was proposed in [16]. At next, we offer proofs for generalized simplified decoding methods of MK polar codes. We focus on Rate-1 and Rate-0 nodes which have much more simpler code structures. And the exact and approximation formulas of Path Metric (PM) are both considered and proven. The proofs and following numerical results indicate that these simplifications are valid for any kernels who are generalized for MK polar codes without incurring any performance degradation.

The rest of this letter is organized as follows. Section 2 reviews MK polar codes and the conventional decoding methods. Section 3 describes the generalize simplified decoding of MK polar codes. The sufficient conditions and corresponding proofs are provided, and time-step requirements are discussed separately. Section 4 presents numerical results which indicate our proposed simplification do not damage error-correction performance. Section 5 draws the conclusions.

## 2 Preliminaries

In this section, we briefly introduce the basic knowledge of multi-kernel polar codes and their corresponding decoding method. A $N$ length polar code is presented by $\mathcal{P}(N, K)$, which carries $K$ bits information. The process of polar coding can be denoted as a matrix multiplication as $\boldsymbol{x} = \boldsymbol{u}\boldsymbol{G}$. $\boldsymbol{u} = \{u_0, u_1, ..., u_{N-1}\}$ is the input sequence. $\boldsymbol{x} = \{x_0, x_1, ..., x_{N-1}\}$ is the polar coded sequence. As Arikan first introducing polar codes in [1], $\boldsymbol{G}$ is the $n$-th Kronecker product of the polarizing matrix $\boldsymbol{G} = \boldsymbol{T}_2^{\otimes n}$.

### 2.1 Multi-kernel Polar Codes

Multi-kernel polar codes are a generalization of Arikan's polar codes. $\boldsymbol{T}_3 = \left(\begin{smallmatrix}1&0&0\\1&1&0\\1&0&1\end{smallmatrix}\right)$ was introduced in [16] which is invertible and can be used for systematic encoding and decoding. MK polar codes provide more flexible length in conventional polar coding whose transform matrix defines $\boldsymbol{G} = \boldsymbol{T}_{k_0} \otimes ...\boldsymbol{T}_{k_i} ... \otimes \boldsymbol{T}_{k_{S-1}}$, where $\boldsymbol{T}_{k_i}$ is a $N_{k_i} \times N_{k_i}$ kernel matrix. The size of decoder $v$'s output is $N_v = \prod_{i=0}^{S-1} N_{k_i}$, where $S$ is the number of kernels. The design and arrangement of $\boldsymbol{T}_{k_i}$ influence MK polar codes performance. We collect set $\mathcal{T}_v = \{\boldsymbol{T}_{k_i} | 0 \leq i < S\}$ to represent the corresponding kernel matrices of decoder $v$.

### 2.2 SC and SSC Decoding


(a) Size-2 kernel.
(b) Size-3 kernel.

**Fig. 1.** Block in Tanner graph, corresponding to $N_k \times N_k$ kernel $\boldsymbol{T}_k$.

SC decoding could also be used in MK polar codes on the Tanner graph of the code. Figure 1 describes the Tanner graph of corresponding MK polar codes. Denoting $l_i$ the input Log-Likelihood Ratios (LLRs) of decoder and $\lambda_i$ the output LLRs.

$$\begin{aligned} x_0 &= u_0 \oplus u_1, \\ x_1 &= u_1, \end{aligned} \tag{1}$$

describes coding relation defined in $\boldsymbol{T}_2$ corresponding Tanner graph. Based on the inverse of the update rule, the estimation of received code $\hat{u}_i$ is calculated by

$$\begin{aligned} \hat{u}_0 &= \mathrm{h}(\lambda_0) = \mathrm{h}(l_0 \boxplus l_1), \\ \hat{u}_1 &= \mathrm{h}(\lambda_1) = \mathrm{h}((-1)^{\hat{u}_0} l_0 + l_1). \end{aligned} \tag{2}$$

In Eq. (2), $\mathrm{h}(x)$ is hard decision function, which equals 0 when $x > 0$, and otherwise, 1. $\boxplus$ defines as

$$a \boxplus b = \ln\left(\frac{1 + e^{a+b}}{e^a + e^b}\right) \approx \min(a, b)(\mathrm{sgn}(a)\mathrm{sgn}(b)). \tag{3}$$

$\mathrm{sgn}(x)$ is sign function, which equals 1 when $x > 0$, and otherwise, $-1$. $\boldsymbol{T}_3$'s coding schemes are defined as Eq. (4),

$$\begin{aligned} x_0 &= u_0 \oplus u_1 \oplus u_2, \\ x_1 &= u_1, \\ x_2 &= u_2. \end{aligned} \tag{4}$$

SSC decoding was proposed to exclude redundant calculation of conventional SC decoding. The basic idea is to perform parallel decoder on SC decoding tree. Identifying special nodes from binary tree of polar codes, and deducing corresponding equivalent formulas, which saves the time-step of traversing from the special nodes to the leaf nodes. In general, Rate-1 and Rate-0 nodes are widely used in SSC decoding. Rate-1 node represent special node whose corresponding leaf nodes all carry information bits. On the contrary, Rate-0's leaf nodes are all frozen bits. According to SSC proposed in [9], a Rate-1 node $v$ of binary tree decoder's candidate $\boldsymbol{\beta}_v$ can be calculated via

$$\boldsymbol{\beta}_v = \mathrm{h}(\boldsymbol{\alpha}_v), \tag{5}$$

$\boldsymbol{\alpha}_v$ is the input LLRs of local decoder $v$. As for Rate-0 node, its candidate code $\boldsymbol{\beta}_v = 0$. At the top of decoding tree, the corresponding codeword $\boldsymbol{x}$ is calculated via $\boldsymbol{\beta}_v \boldsymbol{G}$.

### 2.3 SCL and SSCL Decoding

SCL decoding significantly improves the error-correction performance of polar codes at short and medium code length. It considers and stores both possible values 0 and 1, at each step a information bit is estimated. By introducing path metric, it maintains $L$ paths of possible candidates with the most possibility. $\mathrm{PM}_{i,l}$ is the $i$-th step of $l$-th path's path metric value, defined as

$$\mathrm{PM}_{i,l} = \mathrm{PM}_{i-1,l} + \ln(1 + e^{-(1-2\hat{u}_{i,l})\alpha_{i,l}}). \tag{6}$$

Every $i$-th step estimating bit produces $2L$ new candidates, half of which with larger PM values will be discarded. Equation (7) is a Hardware Friendly (HWF) version calculation of Eq. (6), reducing calculation complexity at the expense of decoding performance.

$$\mathrm{PM}_{i,l} = \begin{cases} \mathrm{PM}_{i-1,l}, & \text{if } \hat{u}_{i,l} = \frac{1}{2}(1 - \mathrm{sng}(\alpha_{i,l})), \\ \mathrm{PM}_{i-1,l} + |\alpha_{i,l}|, & \text{otherwise.} \end{cases} \tag{7}$$

SSCL performs simplified method on list decoding of polar codes. The key is to prove and induct PM calculation can still be deduced and performed on special nodes proposed in SSC decoding. [14] offered relevant proofs and defined Rate-1 nodes' $\mathrm{PM}_{v,l}$ is calculated via

$$\mathrm{PM}_{v,l} = \begin{cases} \sum_{i=0}^{N_v-1} \ln(1 + e^{-(1-2\beta_{v(i),l})\alpha_{v(i),l}}), & \text{Exact,} \\ \frac{1}{2}\sum_{i=0}^{N_v-1} \mathrm{sgn}(\alpha_{v(i),l})\alpha_{v(i),l} & \text{HWF.} \\ -(1-2\beta_{v(i),l})\alpha_{v(i),l}, & \end{cases} \tag{8}$$

And the PM value of Rate-0 node is calculated through

$$\mathrm{PM}_{v,l} = \sum_{i=0}^{N_v-1} \ln(1 + e^{-\alpha_{v(i),l}}). \tag{9}$$

## 3 Generalized Simplified Decoding Algorithms

In this paper, we only focus on Rate-1 and Rate-0 which have much simpler representations under MK polar coding scheme. We extend the simplification conclusion of size-2 based polar codes to MK. In order to reduce the calculation complexity of MK polar codes decoding, the generalized conditions for kernels who could be adopted to perform simplified SC and SCL are explored in our proposed algorithms. At the end of each algorithm, the time-step reduction effects are discussed separately.

### 3.1 Generalized SSC Decoding for MK Polar Codes

In this section, we first derive the equivalent expression of $\boldsymbol{T}_3$ kernel's Rate-1 decoder. Then we propose the sufficient conditions and proof for kernels of MK polar codes who are suitable for generalized SSC decoding. The corresponding time-step analysis is mentioned at the end.

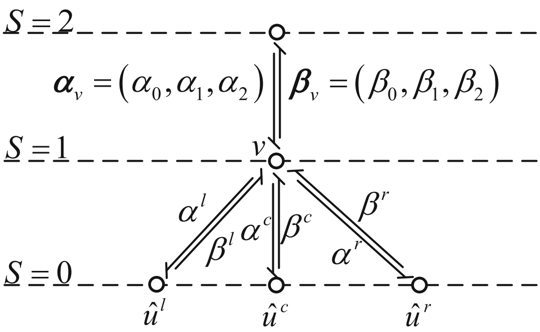


**Fig. 2.** SC decoding tree which $\boldsymbol{G} = \boldsymbol{T}_3$.

**Rate-1 Nodes.** Figure 2 depicts the decoding tree defined by $\boldsymbol{T}_3$ kernel. Stage $S = 0$ represents leaf nodes of decoding trees. Their corresponding input messages are represented as $\alpha^l$, $\alpha^c$ and $\alpha^r$, which are relative to their farther node's input message $\boldsymbol{\alpha}_v$. For $\boldsymbol{T}_3$ defined Tanner graph, according to Eq. (4), we have

$$\begin{aligned} \beta^l &= \beta_0 \oplus \beta_1 \oplus \beta_2, \\ \beta^c &= \beta_1 = \hat{u}^l \oplus \beta_0 \oplus \beta_2, \\ \beta^r &= \beta_2 = \hat{u}^l \oplus \hat{u}^c \oplus \beta_0. \end{aligned} \tag{10}$$

$\hat{u}^l$, $\hat{u}^c$ and $\hat{u}^r$ are the estimation bits. In Fig. 2, $\beta^l$, $\beta^c$ and $\beta^r$ represent the output candidate code of their corresponding nodes. For leaf nodes, $\beta^l = \hat{u}^l$ and etc. Therefore, the corresponding soft messages are calculated as

$$\begin{aligned} \alpha^l &= \alpha_0 \boxplus \alpha_1 \boxplus \alpha_2, \\ \alpha^c &= \alpha_1 + (-1)^{\hat{u}^l} \alpha_0 \boxplus \alpha_2, \\ \alpha^r &= \alpha_2 + (-1)^{\hat{u}^l \oplus \hat{u}^c} \alpha_0. \end{aligned} \tag{11}$$

Now, assuming that $\alpha_i \neq 0$ and performing hard decision function on both sides of the equal sign of Eq. (11), we have

$$\mathrm{h}(\alpha^l) = \mathrm{h}(\alpha_0) \oplus \mathrm{h}(\alpha_1) \oplus \mathrm{h}(\alpha_2),$$

$$\begin{aligned}\mathrm{h}(\alpha^c) &= \mathrm{h}\left(\alpha_1 + (1 - 2\mathrm{h}(\alpha^l))(\alpha_0 \boxplus \alpha_2)\right) \\ &= \mathrm{h}(\alpha_1 + (1 - 2(\mathrm{h}(\alpha_0) \oplus \mathrm{h}(\alpha_1) \oplus \mathrm{h}(\alpha_2)))(\alpha_0 \boxplus \alpha_2)) \\ &= \mathrm{h}(\alpha_1 + (1 - 2(\mathrm{h}(\alpha_0 \boxplus \alpha_2) \oplus \mathrm{h}(\alpha_1)))(\alpha_0 \boxplus \alpha_2)) \\ &= \mathrm{h}(\alpha_1),\end{aligned}$$

$$\begin{aligned}\mathrm{h}(\alpha^r) &= \mathrm{h}(\alpha_2 + (1 - 2\mathrm{h}(\alpha^l \boxplus \alpha^c))\alpha_0) \\ &= \mathrm{h}(\alpha_2 + (1 - 2(\mathrm{h}(\alpha^l) \oplus \mathrm{h}(\alpha^c)))\alpha_0) \\ &= \mathrm{h}(\alpha_2 + (1 - 2(\mathrm{h}(\alpha_0) \oplus \mathrm{h}(\alpha_2)))\alpha_0) \\ &= \mathrm{h}(\alpha_2).\end{aligned}$$

According to that and Eq. (4), we have

$$\begin{aligned}\beta_0 &= \beta^l \oplus \beta^c \oplus \beta^r = \mathrm{h}(\alpha^l) \oplus \mathrm{h}(\alpha^c) \oplus \mathrm{h}(\alpha^r) \\ &= \mathrm{h}(\alpha_0) \oplus \mathrm{h}(\alpha_1) \oplus \mathrm{h}(\alpha_2) \oplus \mathrm{h}(\alpha_1) \oplus \mathrm{h}(\alpha_2) \\ &= \mathrm{h}(\alpha_0), \\ \beta_1 &= \beta^c = \mathrm{h}(\alpha^c) = \mathrm{h}(\alpha_1), \\ \beta_2 &= \beta^r = \mathrm{h}(\alpha^r) = \mathrm{h}(\alpha_2).\end{aligned} \tag{12}$$

Equation (12) indicates $\boldsymbol{\beta}_{\boldsymbol{T}_3} = \mathrm{h}(\boldsymbol{\alpha}_{\boldsymbol{T}_3})$, where $\boldsymbol{\beta}_{\boldsymbol{T}_3}$ is $\boldsymbol{T}_3$ defined $N_v = 3$ length decoder's output candidate and $\boldsymbol{\alpha}_{\boldsymbol{T}_3}$ is the relative input message vector.

Here, we propose Theorem 1.

**Theorem 1.** *If $\boldsymbol{T}_k \in \mathcal{T}_v$ and $\boldsymbol{\beta}_{\boldsymbol{T}_k} = h(\boldsymbol{\alpha}_{\boldsymbol{T}_k})$, decoder $v$ is a Rate-1 node. Then $\boldsymbol{\beta}_v = h(\boldsymbol{\alpha}_v)$.*

*Proof (Proof of Theorem 1).*

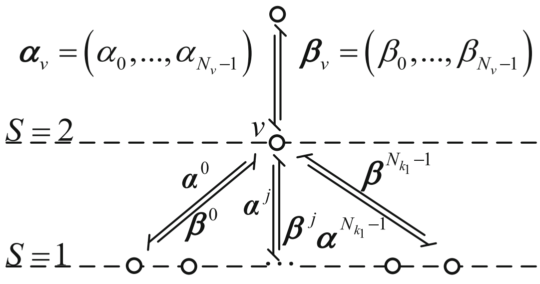


**Fig. 3.** SC decoding tree which $\boldsymbol{G} = \boldsymbol{T}_{k_0} \otimes \boldsymbol{T}_{k_1}$.

Figure 3 shows stage 2's corresponding soft massage and candidate code updates. $\boldsymbol{\alpha}^i$ and $\boldsymbol{\beta}^i$ are node $v$'s $i$-th son's corresponding input and output vectors.

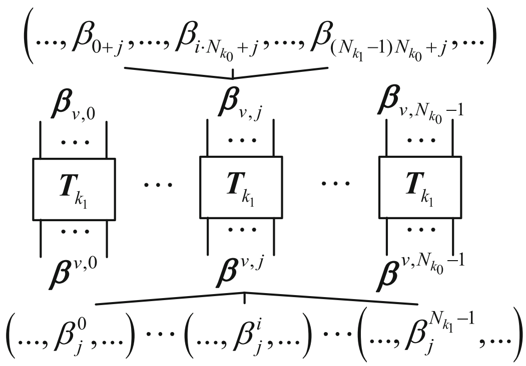

**Fig. 4.** SC decoding tree's stage 1 equivalent decoders.

By rearranging stage 2's input and output candidates, we divide node $v$ into $N_{k_0}$ encoders. And the $j$-th encoder's input code is denoted as $\boldsymbol{\beta}^{v,j}$, where $\boldsymbol{\beta}^{v,j} = \{\beta_j^0, ..., \beta_j^i, ...\beta_j^{N_{k_1}-1}\}$ and $0 \leq j < N_{k_0}$. And the corresponding output code is $\boldsymbol{\beta}_{v,j}$, where $\boldsymbol{\beta}_{v,j} = \{\beta_{0+j}, ..., \beta_{i \cdot N_{k_0}+j}, ..., \beta_{(N_{k_1}-1)N_{k_0}+j}\}$, $0 \leq j < N_{k_0}$, which is rearranged by $\boldsymbol{\beta}_v$.

According to polar coding scheme, we have $\boldsymbol{\beta}_{v,j} = \boldsymbol{\beta}^{v,j}\boldsymbol{T}_{k_1}$. And we suppose $\boldsymbol{T}_{k_0}$ satisfies $\boldsymbol{\beta}_{\boldsymbol{T}_{k_0}} = \mathrm{h}(\boldsymbol{\alpha}_{\boldsymbol{T}_{k_0}})$. That indicates that $\boldsymbol{\beta}^{v,j} = \mathrm{h}(\boldsymbol{\alpha}^{v,j})$, which is the same with $\hat{u} = \mathrm{h}(\alpha)$. Hence, we have $\boldsymbol{\beta}_{v,j} = \mathrm{h}(\boldsymbol{\alpha}_{v,j})$. It should be noted that the rearrangement of $v$'s input and output does not change the relative order of corresponding vectors. Then, we have $\boldsymbol{\beta}_v = \mathrm{h}(\boldsymbol{\alpha}_v)$. For $S = 2$, the theorem is proven. From Fig. 3, it's easy to see that the proof is still hold when $S > 2$. Therefore, Theorem 1 is proven for MK polar codes' Rate-1 node with any stage.

**Rate-0 Nodes.** A Rate-0 node is decoder whose corresponding leaf nodes are frozen bits. Take $\boldsymbol{u}_v$ as the corresponding information vector of node $v$, where $\boldsymbol{u}_v = 0$. For local decoder $v$, we know that $\boldsymbol{\beta}_v = \boldsymbol{u}_v\boldsymbol{G}$. Hence, $\boldsymbol{\beta}_v = 0$, which is irrelevant to $\boldsymbol{G}$. In a word, the output candidate for a Rate-0 node of MK polar codes is always 0, no matter what stage and transition matrix it is.

**Time-Step Analysis of Generalized SSC.** Here, we analyze time-step spending of MK polar codes SSC decoding. The original SC decoding algorithm [1] takes $2N_v - 1$ time steps to decode $N_v$ length code, which contains one step to calculate the corresponding likelihood ratios. As for local decoder $v$ who is not root node, it costs $2N_v - 2$ time steps for $\boldsymbol{T}_2$ kernel SC decoding. Follow the same rule, MK polar code SC decoding takes $\sum_{j=0}^{S-1}\prod_{i=0}^{j} N_{\boldsymbol{T}_{k_i}}$ time steps. Due to the deletion of the complete tree traversal, SSC decoding for Rate-1 and Rate-0 nodes only need 1 time step.

### 3.2 Generalized SSCL for MK Polar Codes

To collect the conditions for kernels of MK polar codes which could perform generalized SSCL decoding, we study $\boldsymbol{T}_3$ as example. The exact and hardware friendly calculation formulas of PMs are derived. Then we provide the sufficient conditions and proofs for generalized simplified list decoding of MK polar codes. Time-step requirements are also discussed.

**Rate-1 Nodes.** The $\boldsymbol{T}_2$ defined polar codes' list decoding simplification has been proved in [14]. In this paper, we attempt to generalize this simplification to $\boldsymbol{T}_3$ and other kernels who satisfied certain conditions. For simplicity consideration, we use $\eta$ to represent $1-2\beta$.

For Rate-1 polar code of length $N_v = 3$ and $\boldsymbol{G} = \boldsymbol{T}_3$, based on Eq. (11), we have

$$\begin{aligned} \alpha^l &= \alpha_0 \boxplus \alpha_1 \boxplus \alpha_2, \\ \alpha^c &= \alpha_1 + \eta^l(\alpha_0 \boxplus \alpha_2), \\ \alpha^r &= \alpha_2 + \eta^l\eta^c\alpha_0. \end{aligned} \tag{13}$$

And analyze the relationship defined by Eq. (4), we have

$$\begin{aligned} \eta^l &= \eta_0\eta_1\eta_2, \\ \eta^c &= \eta_1, \\ \eta^r &= \eta_2. \end{aligned} \tag{14}$$

Take it to Eq. (13), we have

$$\begin{aligned} \alpha^l &= \alpha_0 \boxplus \alpha_1 \boxplus \alpha_2, \\ \alpha^c &= \alpha_1 + \eta_0\eta_1\eta_2(\alpha_0 \boxplus \alpha_2), \\ \alpha^r &= \alpha_2 + \eta_0\eta_2\alpha_0. \end{aligned} \tag{15}$$

Now, with Eq. (14) and (15), we build relationship of input soft messages and output candidate codes between a Rate-1 node and its descendants nodes. Now, we take Eq. (14) and (15) into (8), the path metric associated with $\boldsymbol{T}_3$ is

$$\begin{aligned} \mathrm{PM}_{\boldsymbol{T}_3} &= \ln\left(e^{-\eta_0\eta_1\eta_2(\alpha_0\boxplus\alpha_1\boxplus\alpha_2)}+1\right) \\ &\quad + \ln\left(e^{-\eta_1(\alpha_1+\eta_0\eta_1\eta_2(\alpha_0\boxplus\alpha_2))}+1\right) \\ &\quad + \ln\left(e^{-\eta_2(\alpha_2+\eta_0\eta_2\alpha_0)}+1\right) \\ &= \ln\left(\left(\frac{e^{\alpha_0}+e^{\alpha_1}+e^{\alpha_2}+e^{\alpha_0+\alpha_1+\alpha_2}}{1+e^{\alpha_0+\alpha_1}+e^{\alpha_0+\alpha_2}+e^{\alpha_1+\alpha_2}}\right)^{-\eta_0\eta_1\eta_2}+1\right) \\ &\quad + \ln\left(e^{-\eta_1\alpha_1}\left(\frac{1+e^{\alpha_0+\alpha_2}}{e^{\alpha_0}+e^{\alpha_2}}\right)^{-\eta_0\eta_2}+1\right) \\ &\quad + \ln\left(e^{-\eta_0\alpha_0-\eta_2\alpha_2}+1\right). \end{aligned} \tag{16}$$

For the sake of brevity, we introduce $\varDelta$ to replace $\ln((e^{\alpha_0}+1)(e^{\alpha_1}+1)(e^{\alpha_2}+1))$, after expand and simplification Eq. (16) we have

$$\mathrm{PM}_{\boldsymbol{T}_3} = \begin{cases} \Delta - \alpha_0 - \alpha_1 - \alpha_2, & \text{when } \eta_0, \eta_1, \eta_2 = 1, 1, 1, \\ \Delta - \alpha_0 - \alpha_1, & \text{when } \eta_0, \eta_1, \eta_2 = 1, 1, -1, \\ \Delta - \alpha_0 - \alpha_2, & \text{when } \eta_0, \eta_1, \eta_2 = 1, -1, 1, \\ \Delta - \alpha_0, & \text{when } \eta_0, \eta_1, \eta_2 = 1, -1, -1, \\ \Delta - \alpha_1 - \alpha_2, & \text{when } \eta_0, \eta_1, \eta_2 = -1, 1, 1, \\ \Delta - \alpha_1, & \text{when } \eta_0, \eta_1, \eta_2 = -1, 1, -1, \\ \Delta - \alpha_2, & \text{when } \eta_0, \eta_1, \eta_2 = -1, -1, 1, \\ \Delta, & \text{when } \eta_0, \eta_1, \eta_2 = -1, -1, -1, \end{cases} \\ = \ln(1 + e^{-\eta_0\alpha_0}) + \ln(1 + e^{-\eta_1\alpha_1}) + \ln(1 + e^{-\eta_2\alpha_2}) \\ = \sum_{i=0}^{N_{\boldsymbol{T}_3}-1} \ln(1 + e^{-\eta_i\alpha_i}). \tag{17}$$

Furthermore, we will introduce the HWF version path metric calculation's simplification of MK polar codes list decoding. For simplicity, we introduce $s_i = \mathrm{sgn}(\alpha_i)$. And based on the character of sign function, we have $\mathrm{sgn}(x)x = |x|$. Take the approximation of Eq. (13) and (14), we have

$$\begin{aligned} \alpha^l &= s_0 s_1 s_2 \min(|\alpha_0|, |\alpha_1|, |\alpha_2|), \\ \alpha^c &= \alpha_1 + \eta^l s_0 s_1 \min(|\alpha_0|, |\alpha_2|) \\ &= \alpha_1 + \eta_0\eta_2 s_0 s_2 \min(|\alpha_0|, |\alpha_2|), \\ \alpha^r &= \alpha_2 + \eta^l\eta^c\alpha_0 = \alpha_2 + \eta_0\eta_2\alpha_0. \end{aligned} \tag{18}$$

According to Eq. (8)'s HWF version, we have $\boldsymbol{T}_3$ kernel's corresponding path metric formulation Eq. (19).

$$\begin{aligned} \mathrm{PM}_{\boldsymbol{T}_3} = \tfrac{1}{2} (\ & \mathrm{sgn}(\alpha^l)\alpha^l - \eta^l\alpha^l + \mathrm{sgn}(\alpha^c)\alpha^c - \eta^c\alpha^c \\ & + \mathrm{sgn}(\alpha^r)\alpha^r - \eta^r\alpha^r\ ). \end{aligned} \tag{19}$$

From Eq. (19), substituting Eq. (18), the path metric associated with $\boldsymbol{T}_3$ would be computed as

$$\begin{aligned} 2\mathrm{PM}_{\boldsymbol{T}_3} = & \underbrace{(1 - \eta_0\eta_1\eta_2 s_0 s_1 s_2)\min(|\alpha_0|, |\alpha_1|, |\alpha_2|)}_{p_0} \\ & + \underbrace{|\alpha_1 + \eta_0\eta_1\eta_2 s_0 s_2 \min(|\alpha_0|, |\alpha_2|)|}_{p_1} - \eta_1\alpha_1 \\ & \underbrace{-\eta_0\eta_2 s_0 s_2 \min(|\alpha_0|, |\alpha_2|) + |\alpha_2 + \eta_0\eta_2\alpha_0|}_{p_2} \\ & - \eta_2\alpha_2 - \eta_0\alpha_0. \end{aligned}$$

Then, we analyze this equation with several situations:

First, we calculate $p_0 + p_1$. When $\eta_0\eta_1\eta_2 s_0 s_1 s_2 = 1$, we have

$$p_0 + p_1 = |\alpha_1 + s_1 \min(|\alpha_0|, |\alpha_2|)| = |\alpha_1| + \min(|\alpha_0|, |\alpha_2|).$$

When $\eta_0\eta_1\eta_2 s_0 s_1 s_2 = -1$, we have

$$p_0 + p_1 = 2\min(|\alpha_0|, |\alpha_1|, |\alpha_2|) + |\alpha_1 - s_1 \min(|\alpha_0|, |\alpha_2|)|\,.$$

In this situation, when $|\alpha_1| \leq \min(|\alpha_0|, |\alpha_2|)$,

$$\begin{aligned} p_0 + p_1 &= 2|\alpha_1| + |\min(|\alpha_0|, |\alpha_2|) - |\alpha_1|| \\ &= |\alpha_1| + \min(|\alpha_0|, |\alpha_2|). \end{aligned}$$

Otherwise, we have

$$\begin{aligned} p_0 + p_1 &= 2\min(|\alpha_0|, |\alpha_2|) + ||\alpha_1| - \min(|\alpha_0|, |\alpha_2|)| \\ &= |\alpha_1| + \min(|\alpha_0|, |\alpha_2|). \end{aligned}$$

Therefore, we have

$$p_0 + p_1 = |\alpha_1| + \min(|\alpha_0|, |\alpha_2|).$$

Then we focus on the calculation of $p_0 + p_1 + p_2$. When $\eta_0\eta_2 s_0 s_2 = 1$,

$$\begin{aligned} p_2 &= -\min(|\alpha_0|, |\alpha_2|) + |s_2|\alpha_2| + \eta_0\eta_2 s_0|\alpha_0|| \\ &= -\min(|\alpha_0|, |\alpha_2|) + |s_2|\alpha_2| + s_2|\alpha_0|| \\ &= -\min(|\alpha_0|, |\alpha_2|) + |\alpha_2| + |\alpha_0|. \end{aligned}$$

Hence, we have

$$p_0 + p_1 + p_2 = |\alpha_0| + |\alpha_1| + |\alpha_2|.$$

When $\eta_0\eta_2 s_0 s_2 = -1$,

$$\begin{aligned} p_2 &= \min(|\alpha_0|, |\alpha_2|) + |s_2|\alpha_2| - s_2|\alpha_0|| \\ &= \min(|\alpha_0|, |\alpha_2|) + ||\alpha_2| - |\alpha_0||. \end{aligned}$$

If $|\alpha_2| \geq |\alpha_0|$,

$$p_0 + p_1 + p_2 = 2|\alpha_0| + |\alpha_2| - |\alpha_0| + |\alpha_1|.$$

Else, we have

$$p_0 + p_1 + p_2 = 2|\alpha_2| - |\alpha_2| + |\alpha_0| + |\alpha_1|.$$

Finally, we have

$$\begin{aligned} \mathrm{PM}_{\boldsymbol{T}_3} &= \tfrac{1}{2}(|\alpha_0| + |\alpha_1| + |\alpha_2| - \eta_0\alpha_0 - \eta_1\alpha_1 - \eta_2\alpha_2) \\ &= \textstyle\sum_{i=0}^{N_{\boldsymbol{T}_3}-1}(\mathrm{sgn}(\alpha_i)\alpha_i - \eta_i\alpha_i). \end{aligned} \tag{20}$$

Before we introduce Theorem 2, we define function $pm(\eta, \alpha)$, which is

$$\mathrm{pm}(\eta, \alpha) = \begin{cases} \ln(1 + e^{-\eta\alpha}), & \text{Exact}, \\ \frac{1}{2}\left(\mathrm{sgn}(\alpha)\alpha - \eta\alpha\right), & \text{HWF}. \end{cases} \tag{21}$$

**Theorem 2.** *If* $\forall \boldsymbol{T}_k \in \mathcal{T}_v$ *all satisfy* $PM_{\boldsymbol{T}_k} = \sum_{i=0}^{N_k-1} pm(\eta_i, \alpha_i)$, $v$ *is a Rate-1 node. Then* $PM_v = \sum_{i=0}^{N_v-1} pm(\eta_i, \alpha_i)$.

*Proof (Proof of Theorem 2).* Figure 5 shows stage 2's path metric updating procedure.

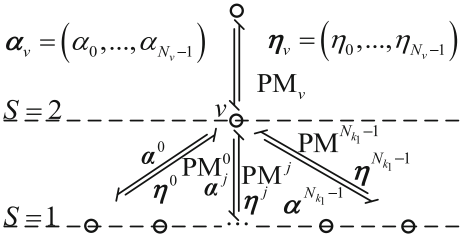


**Fig. 5.** Path metrics of SC decoding tree

Node $v$'s path metric is represented as $\mathrm{PM}_v$. The corresponding candidate is replaced by $\boldsymbol{\eta}_v$ who satisfies $\boldsymbol{\eta}_v = 1 - 2\boldsymbol{\beta}_v$. $\mathrm{PM}^i$ is node $v$'s $i$-th son's corresponding path metric and vector $\boldsymbol{\eta}^i$ is the corresponding output candidate. The path metric at a node $v$ is the summation of the path metrics calculated at its sons. Hence, we have

$$\begin{aligned}\mathrm{PM}_v &= \textstyle\sum_{i=0}^{N_{k_1}-1} \mathrm{PM}^i \\ &= \textstyle\sum_{i=0}^{N_{k_1}-1} \sum_{j=0}^{N_{k_0}-1} \mathrm{pm}(\eta_j^i, \alpha_j^i) \\ &= \textstyle\sum_{j=0}^{N_{k_0}-1} \sum_{i=0}^{N_{k_1}-1} \mathrm{pm}(\eta_j^i, \alpha_j^i).\end{aligned} \tag{22}$$

According to Fig. 4, we know that $\boldsymbol{\beta}_{v,j} = \boldsymbol{\beta}^{v,j}\boldsymbol{T}_{k_1}$. Hence, we have $\mathrm{PM}_{v,j} = \sum_{i=0}^{N_{k_1}-1} \mathrm{pm}(\eta_j^i, \alpha_j^i)$, where $\mathrm{PM}_{v,j}$ is the $j$-th PM value of the new rearranged decoder. $\boldsymbol{T}_{k_1}$ defines the new decoder and satisfies above conditions which means $\mathrm{PM}_{v,j} = \sum_{i=0}^{N_{k_1}-1} \mathrm{pm}(\eta_{i \cdot N_{k_0}+j}, \alpha_{i \cdot N_{k_0}+j})$. Then we have

$$\begin{aligned}\mathrm{PM}_v &= \textstyle\sum_{j=0}^{N_{k_0}-1} \sum_{i=0}^{N_{k_1}-1} \mathrm{pm}(\eta_{i \cdot N_{k_0}+j}, \alpha_{i \cdot N_{k_0}+j}) \\ &= \textstyle\sum_{i=0}^{N_{k_0}N_{k_1}-1} \mathrm{pm}(\eta_i, \alpha_i).\end{aligned} \tag{23}$$

For $S = 2$, the theorem is proven. From Fig. 5, it's easy to see that the proof is still hold when $S > 2$. Hence, Theorem 2 is valid for any stage of tree decoder.

### Rate-0 Nodes

**Theorem 3.** *The PM value of a Rate-0 node can be calculated as:*

$$PM_v = \begin{cases} \sum_{i=0}^{N_v-1} \ln(1 + e^{-\alpha_i}), \ \textit{Exact}, \\ \sum_{i=0}^{N_v-1} \frac{1}{2}\left(|\alpha_i| - \alpha_i\right), \ \textit{HWF}, \end{cases} \tag{24}$$

*where $\alpha_i$ is the input soft message of the Rate-0 node tree.*

*Proof (Proof of Theorem 3).* To proof this theorem, it worth to note that Rate-0 node is only a special case of Rate-1 node. That means decoder already know the output candidate of Rate-0 node tree without traversal the whole tree from top to the leaves. Then we revisit the calculation of Rate-1 decoder fast list decoding path metric Eq. (21). If we take Rate-0 as a normal Rate-1 node. Due to the prior knowledge, $\boldsymbol{\eta}_v$ equals to 1 at this situation. Hence, we have $\mathrm{PM}_v = \sum_{i=0}^{N_v-1} \ln(1 + e^{-\alpha_i})$ in the exact calculation of path metric. And for the approximation version, $\mathrm{PM}_v = \sum_{i=0}^{N_v-1} \frac{1}{2}\left(|\alpha_i| - \alpha_i\right)$.

**Time-Step Analysis of Generalized SSCL.** The conventional SCL needs $3N_v - 2$ to decode a $N_v$ length code. According to [14] and [15], a Rate-1 node needs $\min(L-1, N_v)$ time steps and a Rate-0 node takes one step for $\boldsymbol{T}_2$ kernel based polar codes. When performing conventional SCL decoding algorithm on MK polar codes, a $N_v$ length decoder will cost $\sum_{j=0}^{S-1} \prod_{i=0}^{j} N_{\boldsymbol{T}_{k_i}} + N_v$ time steps. In this paper, the generalize SSCL-MK reduces the necessary time steps to $\min(L-1, N_v)$ and 1.

## 4 Numerical Results

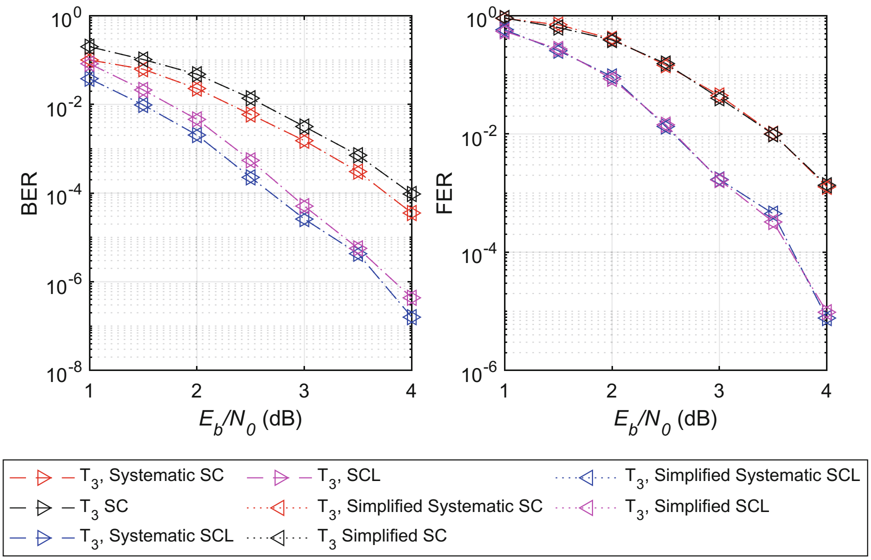


**Fig. 6.** Bit Error Rate (BER) and Frame Error Rate (FER) curves for $\boldsymbol{T}_3$ kernel MK polar codes with coding rate 0.5.

In this section, we demonstrate our proposed generalized simplified SC and SCL MK decoding algorithms with the conventional decoding algorithms. We assume binary phase shift key transmission over the additive white Gaussian noise channel. MK polar codes in Fig. 6 and 7 are build by the method proposed in [18]. $\boldsymbol{T}_3$ kernel MK polar codes in Fig. 6 are 729 length. $\boldsymbol{T}_2$ and $\boldsymbol{T}_3$ kernels MK polar codes are 648 length. Systematic and unsystematic coding methods are used in $\boldsymbol{T}_3$ kernel MK polar codes. SCL method is performed with list size 8. The results in Fig. 6 and 7 show our proposed generalized simplified decoding holds the error-correction performance of conventional SC and SCL methods.

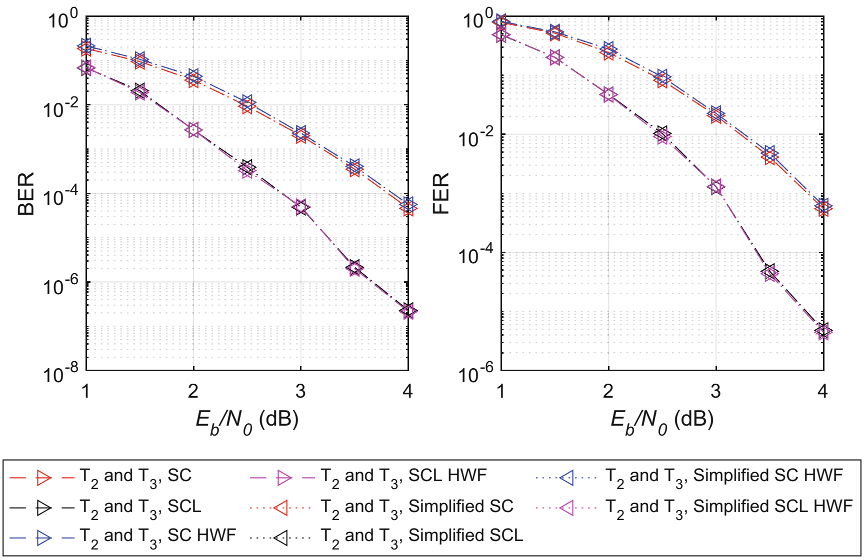


**Fig. 7.** Bit Error Rate (BER) and Frame Error Rate (FER) curves for $\boldsymbol{T}_2$ and $\boldsymbol{T}_3$ kernels MK polar codes with coding rate 0.5.

## 5 Conclusion

In this paper, we have explored generalized SSC and SSCL algorithm for MK polar codes. We provide the sufficient conditions for kernels who are suitable for simplification of SC and SCL calculations. We proved that these simplifications do not incur error-correction performance loss which means that Arikan proposed kernel $\boldsymbol{T}_2$ is a special case of multi-kernel polar codes. This conclusion is valid for any kernel-based polar codes decoding with both exact and approximate calculations. The proposed method saves time step requirements of Rate-1 and Rate-0 nodes from $\sum_{j=0}^{S-1}\prod_{i=0}^{j} N_{\boldsymbol{T}_{k_i}} + N_v$ to $\min(L-1, N_v)$ and 1, which would remarkably reduce the time complexity of conventional SCL.

## References


1. Arikan, E.: Channel polarization: a method for constructing capacity-achieving codes for symmetric binary-input memoryless channels. IEEE Trans. Inf. Theory **55**(7), 3051–3073 (2009)
2. Bioglio, V., Gabry, F., Land, I.: Low-complexity puncturing and shortening of polar codes. In: IEEE Wireless Communications and Networking Conference Workshops (WCNCW). IEEE 2017, pp. 1–6 (2017)
3. El-Khamy, M., Lin, H.-P., Lee, J., Kang, I.: Circular buffer rate-matched polar codes. IEEE Trans. Commun. **66**(2), 493–506 (2017)
4. Korada, S.B., Şaşoğlu, E., Urbanke, R.: Polar codes: characterization of exponent, bounds, and constructions. IEEE Trans. Inf. Theory **56**(12), 6253–6264 (2010)
5. Gabry, F., Bioglio, V., Land, I., Belfiore, J.-C.: Multi-kernel construction of polar codes. In: 2017 IEEE International Conference on Communications Workshops (ICC Workshops), pp. 761–765. IEEE (2017)
6. Bioglio, V., Gabry, F., Land, I., Belfiore, J.-C.: Minimum-distance based construction of multi-kernel polar codes. In: GLOBECOM 2017–2017 IEEE Global Communications Conference, pp. 1–6. IEEE (2017)
7. Bioglio, V., Gabry, F., Land, I., Belfiore, J.-C.: Multi-kernel polar codes: concept and design principles. IEEE Trans. Commun. **68**(9), 5350–5362 (2020)
8. Xia, C., Tsui, C.-Y., Fan, Y.: Construction of multi-kernel polar codes with kernel substitution. IEEE Wirel. Commun. Lett. **9**(11), 1879–1883 (2020)
9. Alamdar-Yazdi, A., Kschischang, F.R.: A simplified successive-cancellation decoder for polar codes. IEEE Commun. Lett. **15**(12), 1378–1380 (2011)
10. Sarkis, G., Giard, P., Vardy, A., Thibeault, C., Gross, W.J.: Fast polar decoders: algorithm and implementation. IEEE J. Sel. Areas Commun. **32**(5), 946–957 (2014)
11. Hanif, M., Ardakani, M.: Fast successive-cancellation decoding of polar codes: identification and decoding of new nodes. IEEE Commun. Lett. **21**(11), 2360–2363 (2017)
12. Condo, C., Bioglio, V., Land, I.: Generalized fast decoding of polar codes. In: IEEE Global Communications Conference (GLOBECOM) 2018, pp. 1–6 (2018)
13. Tal, I., Vardy, A.: List decoding of polar codes. IEEE Trans. Inf. Theory **61**(5), 2213–2226 (2015)
14. Hashemi, S.A., Condo, C., Gross, W.J.: Simplified successive-cancellation list decoding of polar codes. In: IEEE International Symposium on Information Theory (ISIT), pp. 815–819. IEEE (2016)
15. Hashemi, S.A., Condo, C., Gross, W.J.: Fast simplified successive-cancellation list decoding of polar codes. In: IEEE Wireless Communications and Networking Conference Workshops (WCNCW), IEEE, pp. 1–6 (2017)
16. Bioglio, V., Land, I.: On the marginalization of polarizing kernels. In: IEEE 10th International Symposium on Turbo Codes & Iterative Information Processing (ISTC), pp. 1–5 (2018)
17. Cavatassi, A., Tonnellier, T., Gross, W.J.: Fast decoding of multi-kernel polar codes. In: IEEE Wireless Communications and Networking Conference (WCNC), pp. 1–6. IEEE (2019)
18. Trifonov, P.: Efficient design and decoding of polar codes. IEEE Trans. Commun. **60**(11), 3221–3227 (2012)