Decoding Reed-Muller Codes with Successive Factor-Graph Permutations

Journal:	*IEEE Transactions on Communications*
Manuscript ID	TCOM-TPS-21-1085
Manuscript Type:	Transactions Paper Submissions
Date Submitted by the Author:	06-Sep-2021
Complete List of Authors:	Doan, Nghia; McGill University, Electrical & Computer Engineering Hashemi, Seyyed Ali; Stanford University, Electrical Engineering Mondelli, Marco; IST Austria, Gross, Warren; McGill, ECE
Keyword:	Reed-Muller codes, Permutations

Decoding Reed-Muller Codes with Successive Factor-Graph Permutations

Nghia Doan, Seyyed Ali Hashemi, Marco Mondelli, and Warren J. Gross

Abstract

This paper presents a novel successive factor-graph permutation (SFP) scheme that significantly improves the error-correction performance of Reed-Muller (RM) codes under successive-cancellation list (SCL) decoding. In particular, we perform maximum-likelihood decoding on the symmetry group of RM codes to carefully select a good factor-graph permutation on the fly. We further propose an SFP-aided fast SCL decoding that significantly reduces the decoding latency while preserving the error-correction performance of the code. The simulation results show that for the third and fourth order RM codes of length 256, the proposed decoder reduces up to 85% of the memory consumption, 78% of the decoding latency, and more than 99% of the computational complexity of the state-of-the-art recursive projection-aggregation decoder at the frame error rate of 10^{-3}.

Index Terms

Reed-Muller codes, polar codes, 5G, factor-graph permutations.

I. Introduction

Reed-Muller (RM) codes are a class of error-correction codes discovered by Muller [1] and Reed [2], which were proven to achieve the capacity of erasure channels thanks to their large symmetry group [3]. RM codes are similar to polar codes under the factor-graph representation of the codes. The main difference between RM and polar codes is that RM codes are constructed to maximize the minimum distance among all the codewords [1], [2], while polar codes are constructed to minimize the error probability under successive-cancellation (SC) decoding [4]–[7] or SC list (SCL) decoding [8], [9]. An advantage of RM codes over polar codes is that the code construction of RM codes is channel independent, which does not impose additional complexity when the codes are constructed for different communication mediums, even under variable channel conditions.

In addition, it was shown in [10], [11] that with maximum likelihood (ML) decoding, RM codes achieve a better error-correction performance when compared with polar codes. However, ML decoding is generally impractical due to its exponential complexity. As a consequence, RM codes are often decoded using sub-optimal decoders in practice, e.g., SC [10], [12] and SCL decoding [13]–[17]. Recently, a recursive projection-aggregation (RPA) decoding algorithm has been introduced to decode RM codes, whose error-correction performance is close to that of ML decoding [18]. However, RPA decoding in general is of higher complexity when compared with SCL decoding [19]. Thus, it is important to improve the error-correction performance of RM codes under the low-complexity SC-based decoding algorithms.

The error-correction performance of RM codes under various decoding algorithms can be significantly improved by utilizing their rich symmetry group [13], [20]–[22]. Specifically, a list of factor-graph permutations of RM codes is selected and the decoding algorithm is performed on them. As RM and polar codes share the same factor-graph representation, it was observed in [23]–[26] that the error-correction performance of polar codes is also improved by running the decoding algorithms on a list of their factor-graph permutations. However, the decoders proposed in [13], [20]–[26] need to run the same decoding algorithm on different permuted factor graphs, which increases the decoding complexity and memory usage of the decoders. A successive factor-graph permutation (SFP) scheme was introduced in [16] to address this issue. In particular, during the course of SCL decoding, the SFP technique recursively selects a cyclic factor-graph permutation of a RM code to maximize the channel reliability of its constituent codes [16]. It was shown that the error-correction performance of the SFP-aided SCL (SFP-SCL) decoder with a list size of L ($L \geq 2$) is similar to that of a conventional SCL decoder with a list size of $2L$ [16]. However, the factor-graph selection scheme in [16] assumes that all the constituent codes of a RM code are of maximum order. Consequently, the SFP-SCL decoder does not provide significant error-correction performance gains for low-order RM codes when a relatively small list size is used.

In this paper, we propose an improved SFP-SCL decoding algorithm that performs low-complexity ML decoding on the symmetry group of RM codes to carefully select the factor-graph permutations on the fly. In addition, we propose an SFP-aided fast SCL (SFP-FSCL) decoding algorithm that has a significantly higher speed than SFP-SCL decoding. We show that the proposed SFP-FSCL decoder does not incur any error-correction performance loss compared to the proposed SFP-SCL decoding algorithm. The simulation results show that for a RM code

of length 512, the proposed SFP-FSCL decoder with a practical list size $L = 8$ [15], [17], [27] obtains an error-correction performance gain of 1.12 dB for code order $r = 2$, 0.68 dB for $r = 3$, and 0.42 dB for $r = 4$, at the target frame error rate (FER) of 10^{-4} when compared with the factor-graph permutation selection scheme introduced in [16]. Furthermore, for the third and fourth order RM codes of length 256, the proposed decoder reduces 99% of the computational complexity, up to 78% of the decoding latency, and 85% of the memory consumption when compared to those of the RPA decoding algorithm at a similar error-correction performance.

The remainder of this paper is organized as follows. Section II introduces the backgrounds on RM codes and their factor graph-based decoding algorithms. Section III provides details of the proposed decoding techniques, followed by a performance evaluation considering the error-correction performance, decoding latency, and memory requirement of the proposed decoders presented in Section IV. Finally, concluding remarks are drawn in Section V.

II. Preliminaries

A. Reed-Muller Codes

A RM code is specified by a pair of integers $0 \leq r \leq m$ and is denoted as $\mathcal{RM}(r, m)$, where r is the order of the code. $\mathcal{RM}(r, m)$ has a code length $N = 2^m$ with $K = \sum_{i=0}^{r} \binom{m}{i}$ information bits, and a minimum distance $d = 2^{m-r}$. A RM code is constructed by applying a linear transformation to the binary message word $\boldsymbol{u} = \{u_0, u_1, \ldots, u_{N-1}\}$ as $\boldsymbol{x} = \boldsymbol{u}\boldsymbol{G}^{\otimes m}$ where $\boldsymbol{x} = \{x_0, x_1, \ldots, x_{N-1}\}$ is the codeword and $\boldsymbol{G}^{\otimes m}$ is the m-th Kronecker power of the matrix $\boldsymbol{G} = \left[\begin{smallmatrix} 1 & 0 \\ 1 & 1 \end{smallmatrix}\right]$ [28]. The element u_i of $\boldsymbol{u}$ is fixed to 0 if the weight of the i-th row of $\boldsymbol{G}^{\otimes m}$, denoted as w_i, is smaller than d. Formally, $u_i = 0 \ \forall i \in \mathcal{F}$, where $\mathcal{F} = \{i | 0 \leq i < N, w_i < d\}$. In addition, we denote by $\mathcal{I}$ the set of information bits, i.e., $\mathcal{I} = \{i | 0 \leq i < N, w_i \geq d\}$, and $\mathcal{I}$ and $\mathcal{F}$ are known to both the encoder and the decoder.

In this paper, the codeword $\boldsymbol{x}$ is modulated using binary phase-shift keying (BPSK) modulation, and an additive white Gaussian noise (AWGN) channel model is considered. Therefore, the soft vector of the transmitted codeword received by the decoder is given as $\boldsymbol{y} = (\mathbf{1} - 2\boldsymbol{x}) + \boldsymbol{z}$, where $\mathbf{1}$ is an all-one vector of size N, and $\boldsymbol{z} \in \mathbb{R}^N$ is a Gaussian noise vector with variance σ^2 and zero mean. In the log-likelihood ratio (LLR) domain, the LLR vector of the transmitted codeword is given as $\boldsymbol{\alpha}_m = \ln \frac{Pr(\boldsymbol{x}=0|\boldsymbol{y})}{Pr(\boldsymbol{x}=1|\boldsymbol{y})} = \frac{2\boldsymbol{y}}{\sigma^2}$. Fig. 1(a) illustrates the encoding process of $\mathcal{RM}(1, 3)$ using the factor-graph representation of the code, where $N = 8$, $K = 4$, and $\mathcal{I} = \{3, 5, 6, 7\}$. It was shown in [29] that for RM codes of order $r = 1$, the ML decoding

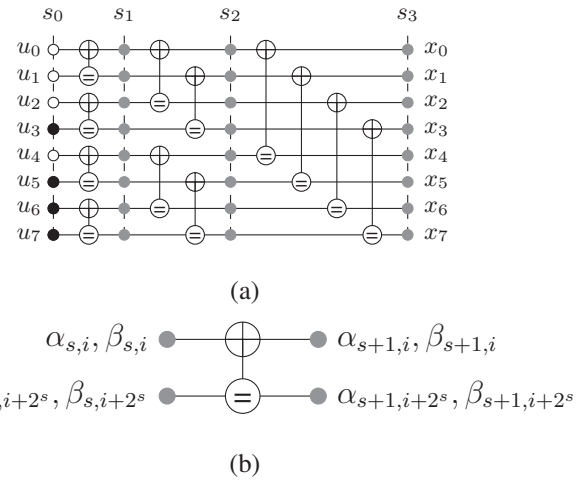


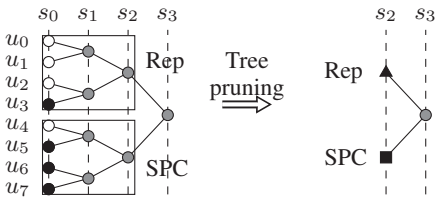
Fig. 1: (a) Factor-graph representation of $\mathcal{RM}(1,3)$, and (b) a PE.

algorithm can be efficiently implemented by utilizing a fast Hadarmard transform (FHT). In the next sections, we summarize various state-of-the-art decoding algorithms used to decode RM codes of order $r > 1$.

B. Successive-Cancellation and Successive-Cancellation List Decoding

SC decoding is executed on the factor-graph representation of the code [10]. To obtain the message word, the soft LLR values and the hard bit estimations are propagated through all the processing elements (PEs), which are depicted in Fig. 1(b). Each PE performs the following computations: $\alpha_{s,i} = f(\alpha_{s+1,i}, \alpha_{s+1,i+2^s})$ and $\alpha_{s,i+2^s} = g(\alpha_{s+1,i}, \alpha_{s+1,i+2^s}, \beta_{s,i})$, where $\alpha_{s,i}$ and $\beta_{s,i}$ are the soft LLR value and the hard-bit estimation at the s-th stage and the i-th bit, respectively. The min-sum approximation formulations of f and g are $f(a,b) = \min(|a|,|b|)\operatorname{sgn}(a)\operatorname{sgn}(b)$, and $g(a,b,c) = b+(1-2c)a$. The soft LLR values at the m-th stage are initialized to $\boldsymbol{\alpha}_m$ and the hard-bit estimation of an information bit at the 0-th stage is obtained as $\hat{u}_i = \beta_{0,i} = \frac{1-\operatorname{sgn}(\alpha_{0,i})}{2}$, $\forall i \in \mathcal{I}$. The hard-bit values of the PE are then computed as $\beta_{s+1,i} = \beta_{s,i} \oplus \beta_{s,i+2^s}$ and $\beta_{s+1,i+2^s} = \beta_{s,i+2^s}$.

A recursive list decoding algorithm was proposed in [13] to decode RM codes by maintaining L most probable SC decoding paths at the same time. It is noteworthy that the SC decoding algorithm in [10] is a special case of the recursive list algorithm proposed in [13], when the list size L is set to 1. Under SCL decoding, the estimation of a message bit $\hat{u}_i$ $(i \in \mathcal{I})$ is considered to be both 0 and 1, i.e., a path split. Thus, the number of candidate codewords (decoding paths)



(a) (b)

Fig. 2: (a) Full binary tree representation of $\mathcal{RM}(1,3)$ illustrated in Fig. 1(a), and (b) the pruned binary tree representation of the same RM code. The left and right child nodes of the parent node at stage s_3 are Rep and SPC, respectively.

doubles after each information bit is estimated. To prevent the exponential growth of the number of decoding paths, a path metric is utilized to select the L most probable decoding paths after each information bit is decoded. In the LLR domain, the low-complexity path metric can be obtained as [15]

$$\text{PM}_l = \begin{cases} \text{PM}_l + \left|\alpha_{0,i_l}\right| & \text{if } \hat{u}_i \neq \frac{1-\text{sgn}(\alpha_{0,i_l})}{2}, \\ \text{PM}_l & \text{otherwise,} \end{cases} \tag{1}$$

where α_{0,i_l} denotes the soft value of the i-th bit at stage 0 of the l-th path, and initially $\text{PM}_l = 0$ $\forall l$. At the end of the decoding process, only the path that has the smallest path metric is selected as the decoding output.

C. Fast Successive-Cancellation List Decoding

SC and SCL decoding can also be illustrated on a binary tree representation of the code [12], [30], [31]. Fig. 2(a) shows a full binary tree representation of $\mathcal{RM}(1,3)$, whose factor graph is depicted in Fig. 1(a). In [17], [27], the authors proposed FSCL decoding algorithms for various special sub polar codes, which are also RM codes. The FSCL decoder preserves the error-correction performance of SCL decoding and completely removes the need to traverse the child nodes of the special nodes. Therefore, the decoding latency of FSCL decoding is significantly smaller than SCL decoding.

Consider a parent node ν located at the s-th stage $(s > 0)$ of the binary tree, which is a RM code specified by a pair of parameters (r_ν, m_ν) with $m_\nu = s$. There are N_ν LLR values and N_ν hard decisions associated with this node, where $N_\nu = 2^{m_\nu}$. Let $\boldsymbol{\alpha}_{\nu_l}$ and $\boldsymbol{\beta}_{\nu_l}$ be the soft and hard

values associated with the parent node ν of the l-th decoding path, respectively. $\boldsymbol{\alpha}_{\nu_l}$ and $\boldsymbol{\beta}_{\nu_l}$ are defined as

$$\begin{cases} \boldsymbol{\alpha}_{\nu_l} = \{\alpha_{s,i_{\min_{\nu_l}}}, \ldots, \alpha_{s,i_{\max_{\nu_l}}}\}, \\ \boldsymbol{\beta}_{\nu_l} = \{\beta_{s,i_{\min_{\nu_l}}}, \ldots, \beta_{s,i_{\max_{\nu_l}}}\}, \end{cases}$$

where $i_{\min_{\nu_l}}$ and $i_{\max_{\nu_l}}$ are the bit indices such that $0 \leq i_{\min_{\nu_l}} < i_{\max_{\nu_l}} \leq N-1$ and $i_{\max_{\nu_l}} - i_{\min_{\nu_l}} = N_\nu - 1$. The hard-decision values of ν in the bipolar form are denoted as $\boldsymbol{\eta}_{\nu_l} = \{\eta_{s,i_{\min_{\nu_l}}}, \ldots, \eta_{s,i_{\max_{\nu_l}}}\}$, where $\eta_{s,i} = 1 - 2\beta_{s,i}$, $i_{\min_{\nu_l}} \leq i \leq i_{\max_{\nu_l}}$.

In this paper, $\mathcal{RM}(0, m_\nu)$, i.e., repetition (Rep) codes, and $\mathcal{RM}(m_\nu - 1, m_\nu)$, i.e., single parity check (SPC) codes are used in FSCL decoding. The decoding operations in FSCL decoding for Rep and SPC nodes are summarized as follows.

1) $\mathcal{RM}(0, m_\nu)$ (Rep): All the leaf nodes of a Rep node are frozen bits, except for $\beta_{0,i_{\max_\nu}}$. The path metric of the l-th decoding path is calculated as [27]

$$\text{PM}_l = \text{PM}_l + \sum_{i=i_{\min_{\nu_l}}}^{i_{\max_{\nu_l}}} \frac{|\alpha_{s,i}| - \eta_{s,i_{\max_{\nu_l}}} \alpha_{s,i}}{2}, \tag{2}$$

where $\eta_{s,i_{\max_{\nu_l}}}$ denotes the bit estimate of the information bit in the Rep node.

2) $\mathcal{RM}(m_\nu - 1, m_\nu)$ (SPC): All the leaf nodes of an SPC node are information bits, except for $\beta_{0,i_{\min_\nu}}$. Let us assume that the elements of $\boldsymbol{\alpha}_\nu$ corresponding to the SPC nodes are sorted in the following order: $\left|\alpha_{s,i_{\min_\nu}}\right| \leq \ldots \leq \left|\alpha_{s,i_{\max_\nu}}\right|$. The parity check sum of the l-th path is first obtained as [17], [27]

$$p_l = \bigoplus_{i=i_{\min_{\nu_l}}}^{i_{\max_{\nu_l}}} \frac{1 - \text{sgn}(\alpha_{s,i})}{2}. \tag{3}$$

The path metric is then updated as [17], [27]

$$\text{PM}_l = \text{PM}_l + p_l \left|\alpha_{s,i_{\min_l}}\right|. \tag{4}$$

The decoding continues with τ path splits, where $\tau = \min(L, N_\nu)$, which is defined as the minimum number of path splits that allows FSCL decoding to preserve the error-correction performance of the conventional SCL decoding algorithm [27]. In each new path split, the path metric is then updated as [17], [27]

$$\text{PM}_l = \begin{cases} \text{PM}_l + |\alpha_{s,i}| + (1 - 2p_l)\left|\alpha_{s,i_{\min_l}}\right| \\ \qquad\qquad \text{if } \eta_{s,i} \neq \text{sgn}(\alpha_{s,i}), \\ \text{PM}_l \qquad\qquad \text{otherwise}, \end{cases} \tag{5}$$

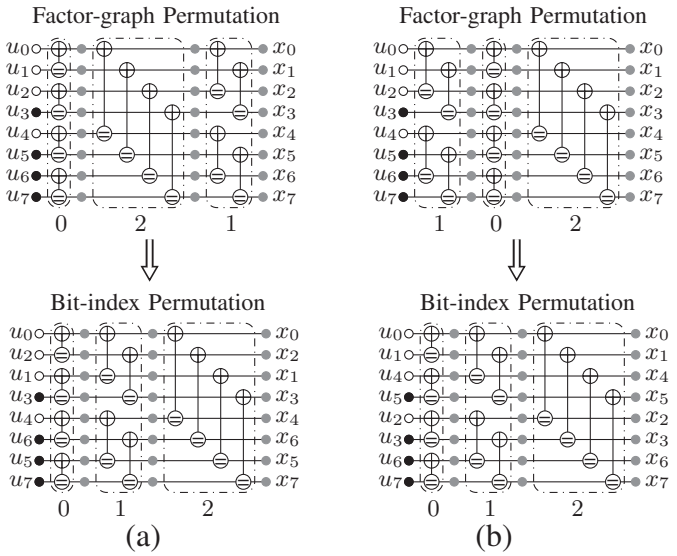


Fig. 3: The equivalent bit-index permutations of the factor-graph permutations (a) $\{0, 2, 1\}$ and (b) $\{1, 0, 2\}$, under the original factor-graph representation with the PE layers indexed as $\{0, 1, 2\}$.

where $i_{\min_{\nu_l}} < i \leq i_{\min_{\nu_l}} + \tau$. The parity check sum is then updated after each path split as [17]

$$p_l = \begin{cases} 1 \oplus p_l & \text{if } \eta_{s,i_l} \neq \operatorname{sgn}(\alpha_{s,i_l}), \\ p_l & \text{otherwise.} \end{cases} \tag{6}$$

When all the bits are estimated, the hard decision of the least reliable bit is updated to maintain the parity check condition of the SPC node, which is given as [17], [27]

$$\beta_{s,i_{\min_{\nu_l}}} = \bigoplus_{i=i_{\min_{\nu_l}}+1}^{i_{\max_{\nu_l}}} \beta_{s,i}. \tag{7}$$

Fig. 2(b) shows an example of the pruned binary tree representation of $\mathcal{RM}(1, 3)$ depicted in Fig. 2(a), where FSCL decoding operations are applied to the Rep and SPC nodes. In addition, with $L = 1$, the FSCL and SCL decoding algorithms revert to the Fast-SC (FSC) [31] and SC [10] decoding algorithms, respectively. Note that the recursive list algorithm in [13] performs fast decoding for the Rep nodes, while the decoding of the information bits of higher-order RM codes follow the SCL decoding operations [14], [15].

D. Successive Factor-Graph Permutations of SCL Decoding

A factor-graph permutation is constructed by permuting the PE stages of the RM code's factor graph [11]. It was shown in [13], [25] that there is a one-to-one mapping between the

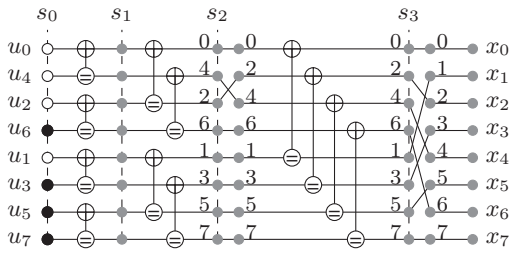


Fig. 4: Bit-index permutation on the original factor graph of $\mathcal{RM}(1,3)$ using the SFP technique in [16].

factor-graph permutation and the bit-index permutation of the original factor-graph. Thus, this decoding technique allows the use of a single decoder architecture while the bit indices of the codeword and message word are permuted. Fig. 3 illustrates examples of the factor-graph permutations of $\mathcal{RM}(1,3)$ whose original factor-graph representation is presented in Fig. 1(a), where the PE layers of the original factor graph in Fig. 1(a) are indexed as $\{0,1,2\}$. Fig. 3 also illustrates the conversion from the factor-graph permutations to the bit-index permutations proposed in [13], [25], which is formally defined as follows. Let $\boldsymbol{b}_i = \{b_{m-1_i}, \ldots, b_{0_i}\}$ be a binary expansion of the bit index i, and $\pi : \{0, \ldots, m-1\} \rightarrow \{0, \ldots, m-1\}$ be a permutation of the PE layers of $\mathcal{RM}(r,m)$. The permuted bit index of i given π in the binary expansion is $\boldsymbol{b}_{\pi(i)} = \{b_{\pi(m-1)_i}, \ldots, b_{\pi(0)_i}\}$, where $0 \leq i < 2^m$ [13], [25].

The SFP scheme proposed in [16] tackles the problem associated with the decoders in [13], [20], [21], [23]–[26], which require multiple code permutations during the course of decoding that significantly increase the memory and computational complexity. The SFP technique carefully selects a single code permutation on the fly to significantly improve the error probability of SCL decoding. Consider a parent node ν located at the s-th stage of the l-th decoding path, whose left-child node is denoted as λ. Let $\mathcal{P}_s$ be a set of factor-graph permutations of a RM code of length 2^s. Given a permutation $\pi \in \mathcal{P}_s$, the f functions when applied to the permuted LLR values associated with ν generate an LLR vector corresponding to its left-child node λ, denoted as $\boldsymbol{\alpha}^{(\lambda)} = \{\alpha_{s-1,i_{\min_\lambda}}, \ldots, \alpha_{s-1,i_{\max_\lambda}}\}$[1]. In [16], the factor-graph permutation π^* of ν is selected to maximize the channel reliabilities corresponding to λ, which allows for a better estimation of

[1]In the rest of the paper, as the permutation selection scheme is applied independently for each decoding path, we drop the subscript l for clarity.

λ under SC and SCL decoding. The selection criteria of π^* is given as [16]

$$\pi^* = \underset{\pi \in \mathcal{P}_s}{\arg\max} \sum_{i=i_{\min_\lambda}}^{i_{\max_\lambda}} |\alpha_{s-1,i}|. \tag{8}$$

The SFP technique considers cyclic factor-graph permutations to be included in the set $\mathcal{P}_s$ [16]. For instance, the cyclic permutations of $\mathcal{RM}(1,3)$ are $\{0,1,2\}$, $\{2,0,1\}$, and $\{1,2,0\}$. Fig. 4 shows an example of the SFP technique applied to $\mathcal{RM}(1,3)$, where the permuted factor graphs are transformed to the permuted bit indices as in [13], [25]. In Fig. 4, let π^* be the factor-graph permutation applied to the bit indices $I = \{0,2,4,6\}$ at stage s_2, corresponding to $\mathcal{RM}(0,2)$. The equivalent bit-index permutation of π^* is $\{0,1,2,3\} \rightarrow \{0,2,1,3\}$. Thus, $I = \{0,2,4,6\} \xrightarrow{\pi^*} I_{\pi^*} = \{0,4,2,6\}$, where I_{π^*} is the resulting permuted bit indices of I. On the other hand, the best factor-graph permutation selected for the $\mathcal{RM}(1,2)$ code at stage s_2 is the original permutation.

III. Improved Successive Factor-Graph Permutations for SC-Based Decoding

The selection criteria in (8), which is used in [16], is an oversimplification that does not take into account the existing parity constraints in the code. In fact, it treats all the constituent RM codes λ as Rate-1 codes. This oversimplification becomes inaccurate, especially for low-order RM codes, as the number of information bits is significantly smaller than $N_\lambda = 2^{s-1}$. Consequently, the error probability of the SFP scheme in [16] for SCL decoding on low-order RM codes is not satisfactory, especially when a small to moderate list size is used.

To tackle this issue, we propose an accurate SFP scheme that selects the best factor-graph permutation π^* by performing ML decoding on the symmetry group of RM codes. The proposed selection criteria is given as

$$\pi^* = \underset{\pi \in \mathcal{P}_s}{\arg\max}\, M_\pi \left(\boldsymbol{\alpha}^{(\lambda)}\right), \tag{9}$$

where $M_\pi \left(\boldsymbol{\alpha}^{(\lambda)}\right)$ is the factor-graph permutation metric of π when π is applied to the parent node ν that is calculated as

$$M_\pi \left(\boldsymbol{\alpha}^{(\lambda)}\right) = \max_{\forall \boldsymbol{\eta}_\lambda} \sum_{i=i_{\min_\lambda}}^{i_{\max_\lambda}} \eta_{s-1,i} \alpha_{s-1,i}, \tag{10}$$

with $\boldsymbol{\eta}_\lambda$ being the hard decisions of a valid codeword corresponding to λ. It can be observed that if λ is a Rate-1 code, (9) reverts to (8) as $\eta_{s-1,i}$ is set to $\mathrm{sgn}(\alpha_{s-1,i})$ to maximize the likelihood of $\boldsymbol{\eta}_\lambda$ and $\boldsymbol{\alpha}_\lambda$. The elements of $\boldsymbol{\eta}_\lambda$ can be calculated by performing ML decoding on λ. However,

ML decoding is generally of high complexity. Therefore, in this paper we derive $M_\pi\left(\boldsymbol{\alpha}^{(\lambda)}\right)$ for special cases of λ for which the ML decoding operations can be realized with low complexity. Similar to [16], we only consider the cyclic permutations in $\mathcal{P}_s$ to significantly reduce the total number of factor-graph permutations.

Since ML decoding can be efficiently implemented for RM codes of order 0 or 1, the permutation metric $M_\pi\left(\boldsymbol{\alpha}^{(\lambda)}\right)$ can be efficiently calculated if λ is a zero-order RM code (Rep code) or a first-order RM code. When λ is of order 2 or higher, we propose to simplify the computation of $M_\pi\left(\boldsymbol{\alpha}^{(\lambda)}\right)$ by using (8). The calculation of $M_\pi\left(\boldsymbol{\alpha}^{(\lambda)}\right)$ for the considered special nodes in this paper is summarized as follows.

1) $\mathcal{RM}(0, s-1)$ (Rep): Let $a = \eta_{s-1,i_{\min_\lambda}} = \ldots = \eta_{s-1,i_{\max_\lambda}}$ be the hard-decision value of the information bit of the Rep node λ in the bipolar form, where $a \in \{-1, 1\}$. The metric $M_\pi\left(\boldsymbol{\alpha}^{(\lambda)}\right)$ is calculated as

$$M_\pi\left(\boldsymbol{\alpha}^{(\lambda)}\right) = \max_{a\in\{-1,1\}} \sum_{i=i_{\min_\lambda}}^{i_{\max_\lambda}} a\alpha_{s-1,i} = \left|\sum_{i=i_{\min_\lambda}}^{i_{\max_\lambda}} \alpha_{s-1,i}\right|. \tag{11}$$

2) $\mathcal{RM}(1, s-1)$: The metric $M_\pi\left(\boldsymbol{\alpha}^{(\lambda)}\right)$ is the likelihood of the best decoding path of λ given $\boldsymbol{\alpha}^{(\lambda)}$, which can be calculated efficiently using FHT decoding [29].

3) $\mathcal{RM}(r_\lambda, s-1)$ ($r_\lambda \geq 2$): We use (8) to calculate $M_\pi\left(\boldsymbol{\alpha}^{(\lambda)}\right)$ for constituent RM codes of order $r_\lambda \geq 2$ as

$$M_\pi\left(\boldsymbol{\alpha}^{(\lambda)}\right) = \sum_{i=i_{\min_\lambda}}^{i_{\max_\lambda}} |\alpha_{s-1,i}|. \tag{12}$$

Note that FHT decoding performs ML decoding for λ, which is carried out during the factor graph selection of a second-order RM code. Since SC decoding progresses by following a single decoding path, if ν is a second-order RM code, the output of FHT decoding for the first-order RM subcode λ is selected as the decoding result. On the other hand, under SCL decoding, the first-order RM subcode λ is decoded using the proposed SFP-SCL decoding algorithm.

We now show that when SCL decoding is considered, using the proposed SFP scheme for a Rep or SPC code is redundant. In fact, applying SCL decoding to decode the Rep and SPC codes on the original permutation of the codes preserves the error-correction performance of the proposed SFP-SCL decoder. Thus, we propose SFP-FSCL decoding that uses FSCL decoding [27] on the original permutation of Rep and SPC nodes and applies the proposed SFP scheme only when ν is neither a Rep nor an SPC node. The following theorem proves that the error-correction performance of SFP-SCL decoding under any SFP setting is the same as that of SCL

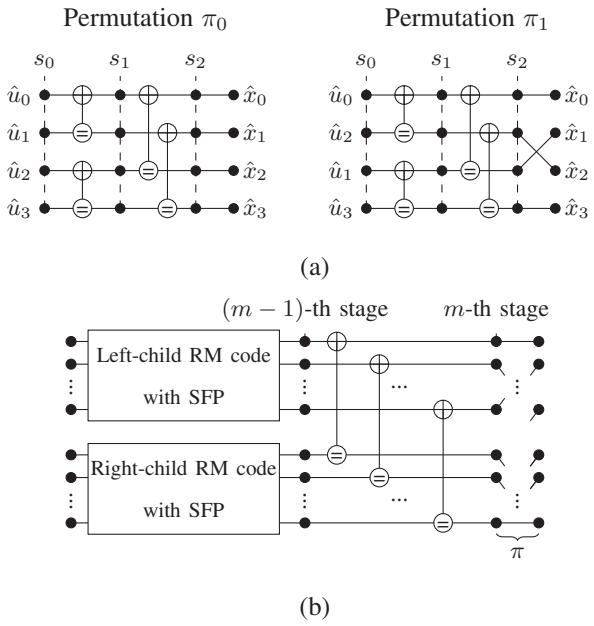


Fig. 5: (a) Bit-index permutations of a RM code of length 4, and (b) a generalized SFP setting of $\mathcal{RM}(r, m)$.

decoding on the original permutation for special RM codes: $\mathcal{RM}(0, m)$ (Rep), $\mathcal{RM}(m, m)$ (Rate-1), or $\mathcal{RM}(m-1, m)$ (SPC).

Theorem 1. *Let ν be a special RM code of size $N_\nu = 2^m$ $(N_\nu \geq 4)$. Applying SCL decoding on the original permutation of ν yields the exact same results as applying the proposed SFP-SCL decoding to ν.*

To prove Theorem 1, we use the following lemma, which states that the error-correction performance of SCL decoding when applied to the original permutation of ν is exactly the same as that of SCL decoding when applied to any permuted codeword of ν, when ν is a Rep, SPC, or Rate-1 code.

Lemma 1. *Let ν be a Rep, SPC, or Rate-1 code, and let π be a permutation applied to ν. SCL decoding on the permuted codeword of ν has the exact same result as SCL decoding on the original permutation of ν.*

Proof of Lemma 1. We use the equivalency of FSCL and SCL decoders on Rep, SPC, and Rate-1 nodes [27] to prove Lemma 1. Any permutation π applied to ν does not change the ranking

of the magnitude of LLR values corresponding to ν. Thus, if ν is either a Rep, or an SPC, or a Rate-1 node, the path splitting order under FSCL decoding remains unchanged for any permutation π. Consequently, FSCL decoding provides the exact same decoding result on ν given any permutation π, including the original permutation. By using the equivalency of FSCL and SCL decoders on Rep, SPC, and Rate-1 nodes [27], it directly holds that applying FSCL decoding to decode the permuted codeword of ν yields the exact same result as applying SCL decoding to decode the same permuted codeword of ν. This proves Lemma 1. □

The SFP scheme recursively selects a permutation for ν and for all of its descendants. We now prove Theorem 1 by extending the proof of Lemma 1 to the case where permutations are selected recursively for ν and all of its descendants.

Proof of Theorem 1. We prove Theorem 1 by induction on m. For a special node ν of length $N_\nu = 4$ ($m = 2$), there are only two different possible bit-index permutations as shown in Fig. 5a that occur at ν and not at its descendants. Therefore, the result in Lemma 1 proves that when $N_\nu = 4$, SCL decoding on the original factor-graph permutation yields the same result as SFP-SCL decoding.

Let us now assume the theorem is true for a special code of length 2^{m-1}. We show that the theorem is also true for a special code of length 2^m. Note that a special node of length 2^m is composed of two descendants of length 2^{m-1}, where each layer undergoes a separate bit-index permutation in the SFP scheme as shown in Fig. 5b. For each special node we have:

- $\mathcal{RM}(0, m)$: Since the LLR value of the only information bit in $\mathcal{RM}(0, m)$ is calculated by taking the sum of all the LLR values at ν, the SFP scheme does not change it. Therefore, performing SCL decoding on the original permutation of ν yields the exact same results as the SFP-SCL decoder.
- $\mathcal{RM}(m, m)$: Let us assume that Theorem 1 holds for $\mathcal{RM}(m-1, m-1)$ with $m \geq 3$. SFP-SCL decoding on $\mathcal{RM}(m, m)$ is equivalent to recursively applying SFP-SCL decoding on the left-child $\mathcal{RM}(m-1, m-1)$ code and then to the right-child $\mathcal{RM}(m-1, m-1)$ code. Based on the induction hypothesis, one can select the original permutation for the left-child and right-child $\mathcal{RM}(m-1, m-1)$ codes without altering the decoding results. Therefore, the SFP scheme applied to $\mathcal{RM}(m, m)$ will only contain the bit-index permutations at ν. By the result of Lemma 1, SCL decoding on the original permutation is equivalent to SFP-SCL decoding for $\mathcal{RM}(m, m)$.

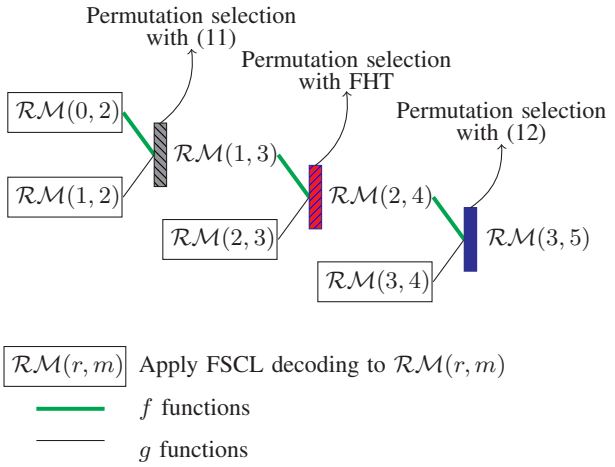


Fig. 6: An example of the proposed SFP-FSCL decoder for $\mathcal{RM}(3,5)$ with a list size $L \geq 2$.

- $\mathcal{RM}(m-1,m)$: Let us assume that Theorem 1 holds for $\mathcal{RM}(m-2,m-1)$ with $m \geq 3$. For $\mathcal{RM}(m-1,m)$, the left-child is a $\mathcal{RM}(m-2,m-1)$ and the right child is a $\mathcal{RM}(m-1,m-1)$. By the induction hypothesis for $\mathcal{RM}(m-2,m-1)$ and the proof of Theorem 1 for $\mathcal{RM}(m-1,m-1)$, one can select the original permutation for the SFP schemes applied to $\mathcal{RM}(m-2,m-1)$ and $\mathcal{RM}(m-1,m-1)$ without altering the decoding results of the proposed SFP-SCL decoder. Therefore, the SFP scheme for $\mathcal{RM}(m-1,m)$ only has permutations at ν and not at its descendants. By the result in Lemma 1, one can select the original permutation at ν without altering the decoding result of the SFP-SCL decoder when applied to $\mathcal{RM}(m-1,m)$. Thus, SCL decoding on the original permutation is equivalent to SFP-SCL decoding for $\mathcal{RM}(m-1,m)$.

This concludes the proof of Theorem 1. □

Fig. 6 illustrates an example of the proposed SFP-FSCL decoding of $\mathcal{RM}(3,5)$ with a list size $L \geq 2$. Since the left-child node of $\mathcal{RM}(3,5)$ is a RM code of order 2, the factor-graph permutation of $\mathcal{RM}(3,5)$ is selected using the selection criteria in (12). Then, the LLR values associated with $\mathcal{RM}(2,4)$ are obtained with the f functions. Since the left-child node of $\mathcal{RM}(2,4)$ is a first-order RM code, FHT [29] is used to select the permutation in $\mathcal{RM}(2,4)$. The LLR values of the first-order RM code are calculated using the f function and the permutation in $\mathcal{RM}(1,3)$ is selected based on (11) to maximize the probability that the Rep node $\mathcal{RM}(0,2)$ is correctly decoded. Based on Theorem 1, $\mathcal{RM}(0,2)$ can be directly decoded under FSCL

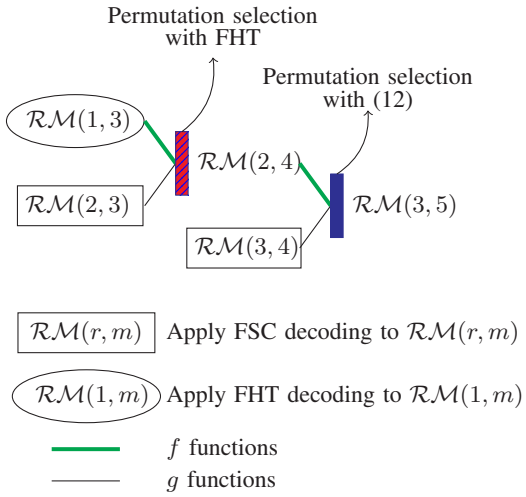


Fig. 7: An example of the proposed SFP-FSC decoder for $\mathcal{RM}(3,5)$.

decoding without degrading the error-correction performance of the proposed SFP-SCL decoder on $\mathcal{RM}(0,2)$. Finally, as all the right-child RM codes are SPC codes, FSCL decoding is used to decode them based on Theorem 1. Fig. 7 shows an example of the proposed SFP-FSC decoding of $\mathcal{RM}(3,5)$, where the first-order RM code uses the decoding result of FHT decoding.

IV. Performance Evaluation

In this section, we compare the performance of the proposed decoders and other SC-based and RPA-based decoders of RM codes in terms of FER, decoding latency, computational complexity, and memory consumption.

A. Quantitative Complexity Analysis

In this paper, we consider a sequential implementation of the factor-graph permutation selection schemes in (9). This ensures the same memory consumption as the SC-based decoders to store the LLR values [16]. However, the SFP scheme requires an additional mQ memory bits to store the permutation metric $M_\pi\left(\boldsymbol{\alpha}^{(\lambda)}\right)$, where m is the maximum number of cyclic permutations and Q is the number of quantization bits. Throughout this paper we use $Q = 32$ for all the considered decoders. Note that the FHT operations compute and store the new LLR values directly to $\boldsymbol{\alpha}^{(\lambda)}$, thus no extra memory is needed under FHT decoding. Table I summarizes the memory requirements of the SC-based RM decoders considered in this paper.

TABLE I: Memory requirements in terms of the number of bits required by various SC-based RM decoders.

Decoding Algorithm	Memory Requirement
FSC [31]	$(2N-1)Q+N$
FSCL [17], [27]	$N(L+1)Q+2NL$
SFP-SC [16] & SFP-FSC (proposed)	$(2N-1)Q+mQ+N$
SFP-SCL [16] & SFP-FSCL (proposed)	$N(L+1)Q+mQ+2NL$

The decoding latency of the SC-based decoders is computed by counting the number of time steps required by all the decoding operations with the following assumptions. We assume that there is no resource constraint. Thus, all concurrent operations in $f(\cdot)$ and $g(\cdot)$ functions, the path metric computations, and the computations of $M_\pi\left(\boldsymbol{\alpha}^{(\lambda)}\right)$ in (11) and (12) require one time step [17], [27]. The permutation metric $M_\pi\left(\boldsymbol{\alpha}^{(\lambda)}\right)$ obtained from FHT requires s time steps for a first-order RM code located at the s-th stage [29]. Furthermore, the factor-graph permutation selection in (9) requires s time steps because each cyclic permutation is evaluated sequentially. In addition, the hard decisions obtained from the LLR values and binary operations are computed instantaneously [15], [17], [27]. Finally, we assume that the number of time steps required by a merge sort algorithm to sort 2^s absolute LLR values of an SPC node is s, and the number of time steps required for the sorting of $2L$ path metrics at each path splitting is $\log_2(2L)$ [32, Chapter 2].

To calculate the computational complexities of the decoders considered in this paper, we count the number of floating point operations, namely, the number of additions, subtractions, and comparisons, required during the course of decoding. Note that the merge sort algorithm requires $N\log_2 N$ comparisons to sort an array of length N [32, Chapter 2].

We use the same methodology to quantify the computational complexity and decoding latency of the RPA decoder [18] and the sparse RPA (SRPA) decoder proposed in [19]. In particular, we consider a fully-parallel implementation of the RPA and the SRPA decoders, in which all the operations that can be carried out concurrently are executed at the same time. The SRPA decoding algorithm runs two fully-parallel RPA decoders with each decoder using a quarter of the code projections at each recursion step [19]. Thus, the SRPA decoder effectively reduces 50% of

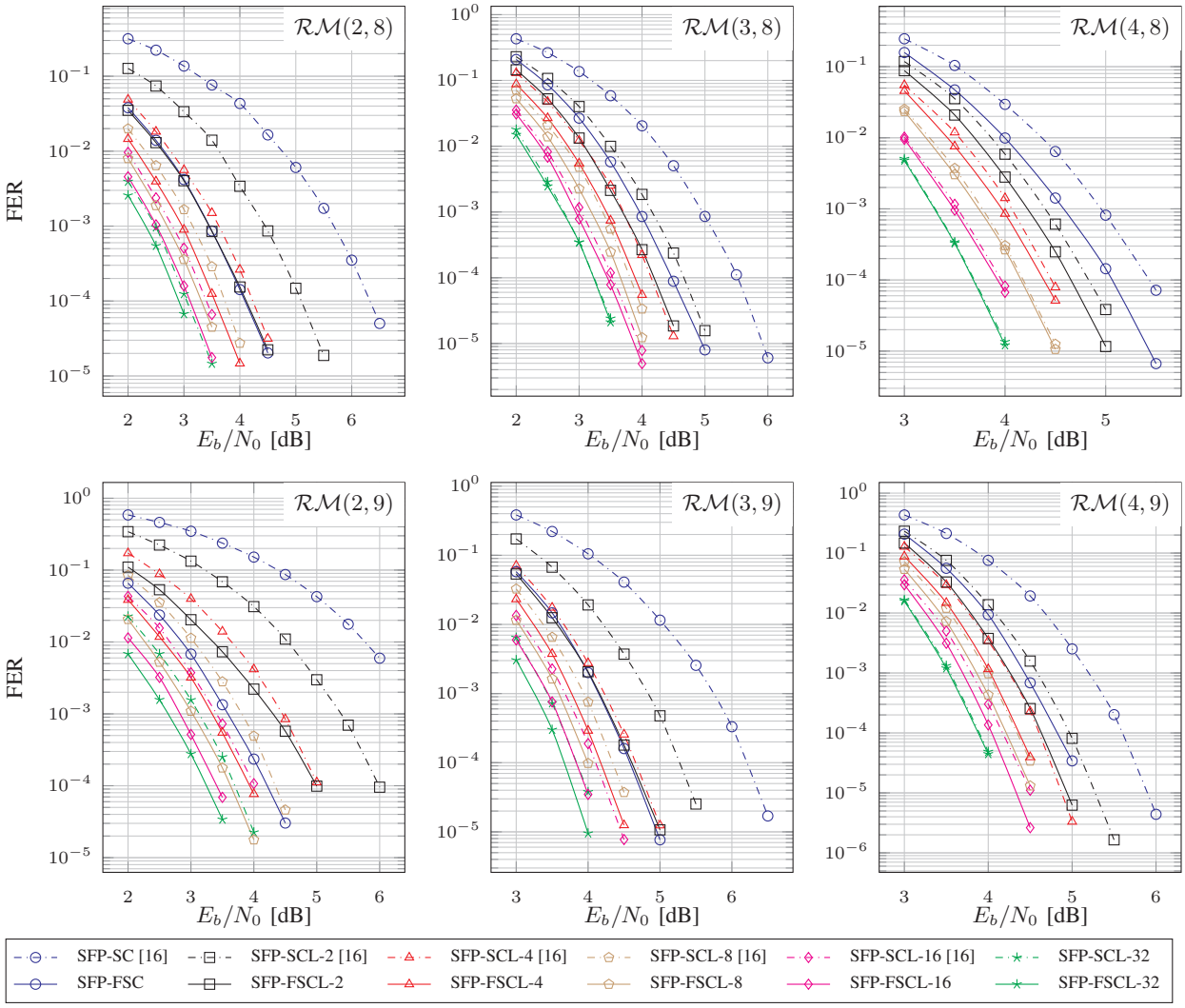


Fig. 8: Error-correction performance of the proposed SFP-FSCL and the SFP-SCL decoders.

the total number of projections used by the conventional RPA algorithm [18]. This configuration incurs negligible error-correction performance loss with respect to the conventional RPA decoder in [18] for the second and third order RM codes of size 256.

B. Comparison with SFP-SCL and FSCL Decoding

Fig. 8 illustrates the FER of the proposed SFP-FSC decoder and SFP-FSCL decoder with list size L (SFP-FSCL-L) for $L \in \{2, 4, 8, 16, 32\}$. The FER of the SFP-SC decoder and SFP-SCL decoder with list size L (SFP-SCL-L) in [16] are also plotted for comparison. It can be observed in Fig. 8 that the FER of the proposed decoders is better than or similar to that of the decoders

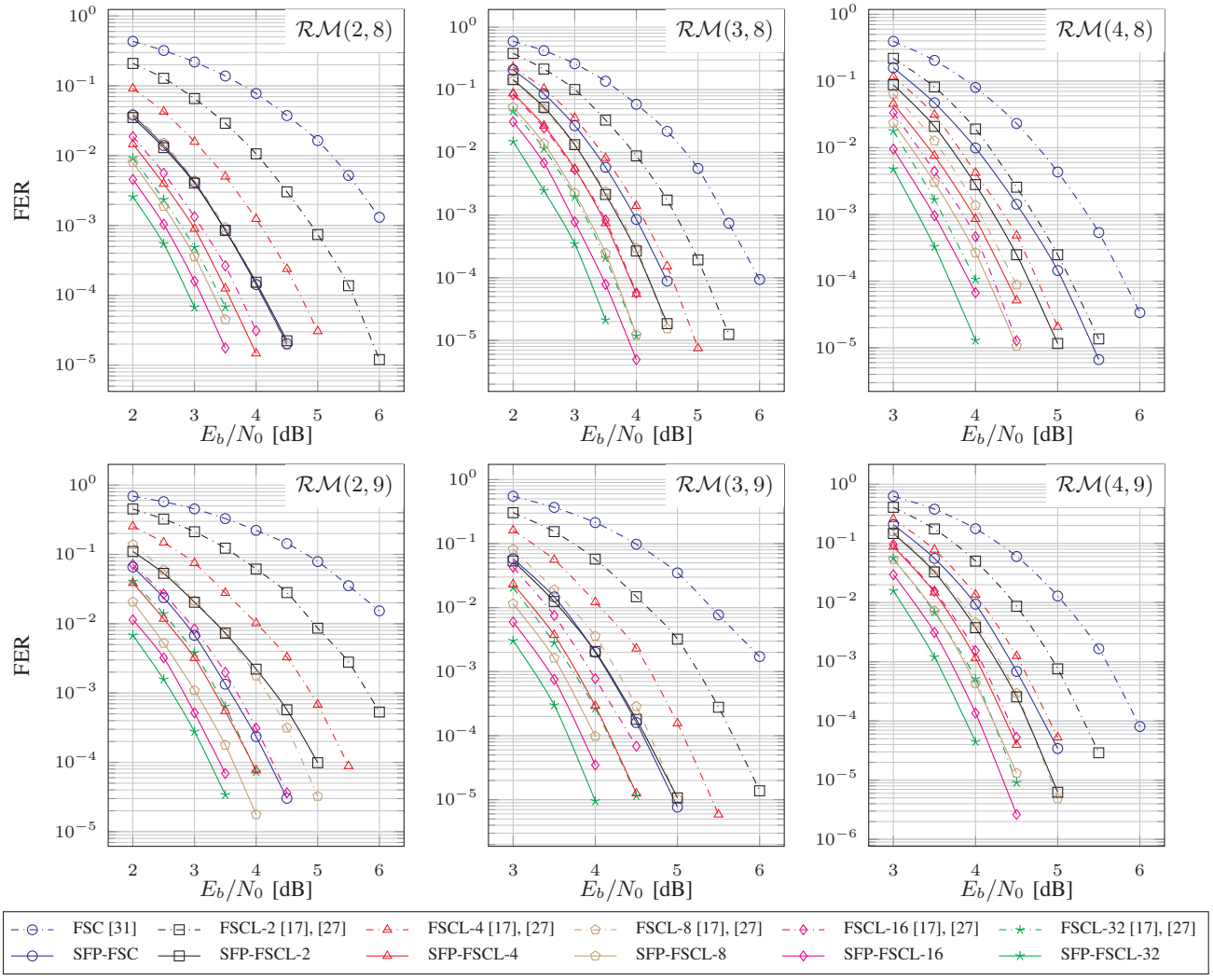


Fig. 9: Error-correction performance of the proposed SFP-FSCL and the FSCL decoders.

of [16] for all the considered RM codes at the same list size. In addition, the FER gains of the proposed decoders are more significant for RM codes of higher lengths and lower code orders, especially with a small list size. For instance, for the second-order RM code of length 512, when compared with the SFP-SC decoder at the target FER of 10^{-4}, the proposed SFP-FSC decoder provides an error-correction performance gain of more than 2 dB. In addition, for the third and fourth order RM codes of length 512, the error-correction performance gains of 1.25 dB and 0.62 dB are recorded for the SFP-FSC decoder also compared with the SFP-SC decoder at the target FER of 10^{-4}, respectively.

In Fig. 9, we compare the FER of the SFP-FSC and SFP-FSCL decoders with those of the

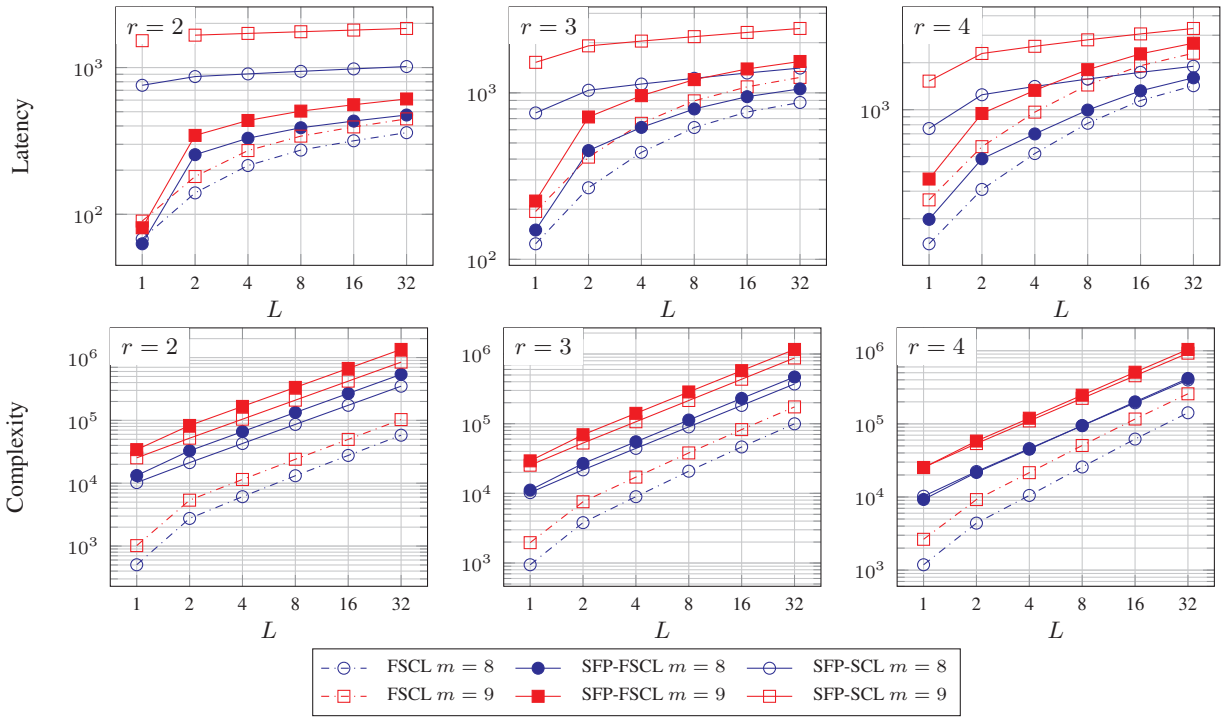


Fig. 10: Decoding latency and computational complexity of the SC-based decoding algorithms considered in Fig. 8 and Fig. 9.

FSC decoder and FSCL decoder with list size L (FSCL-L). It can be seen that to obtain the FER performance of the FSCL decoder with list size 32, the proposed SFP-FSCL decoder only needs a list size of 4 for $\mathcal{RM}(2, 9)$ and $\mathcal{RM}(3, 9)$, a list size of 8 for $\mathcal{RM}(2, 8)$, $\mathcal{RM}(3, 8)$, and $\mathcal{RM}(4, 8)$, and a list size of 16 for $\mathcal{RM}(4, 8)$. Note that for $\mathcal{RM}(2, 9)$ and $\mathcal{RM}(3, 9)$, the SFP-FSC decoder has a similar FER performance to that of the FSCL-16 and FSCL-8 decoders, respectively.

Fig. 10 shows the decoding latency and the computational complexity of the SC-based decoders considered in Fig. 8 and Fig. 9 as a function of the list size L while the memory requirements of the SC-based decoders are provided in Table II. Note that $L = 1$ implies the FSC, SFP-SC, and SFP-FSC decoders. As observed in Fig. 10, the proposed SFP-FSCL decoders have a significantly lower decoding latency than the SFP-SCL decoders for all the considered RM codes with the same list size. This confirms that decoding the special Rep and SPC nodes without tree traversal results in significant latency savings. In addition, the redundant factor-graph permutation selections in SFP-SCL decoder for the special Rep and SPC nodes significantly increases the

TABLE II: Memory requirements in KB of various SC-based decoders considered in Fig. 8 and Fig. 9.

m	8						9					
L	1	2	4	8	16	32	1	2	4	8	16	32
FSCL [17], [27]	2.03	3.12	5.25	9.50	18.00	35.00	4.06	6.25	10.50	19.00	36.00	70.00
SFP-SCL [16] SFP-FSCL	2.06	3.16	5.28	9.53	18.03	35.03	4.09	6.29	10.54	19.04	36.04	70.04

computational complexity and decoding latency, especially for codes of large lengths.

It can be observed from Fig. 9, Fig. 10, and Table II that at the same list size, the proposed SFP-FSCL decoder outperforms the FER of the FSCL decoder at the cost of increased decoding latency and computational complexity, while having a relatively similar memory consumption. Note that for $\mathcal{RM}(2,8)$ and $\mathcal{RM}(2,9)$, the number of time steps required by the SFP-FSC decoder is less than that of the FSC decoder. This is because the first-order RM sub-codes are directly estimated using the FHT operations under SFP-FSC decoding, obviating the need to traverse the tree. It can also be seen in Fig. 10 that at the same list size, the proposed SFP-FSCL decoder requires a higher computational complexity than the SFP-SCL and FSCL decoders, which is a result of the proposed factor-graph permutation selection scheme.

C. Comparison with RPA and SRPA Decoding

Fig. 11 illustrates the error-correction performance, decoding latency, and computational complexity of the proposed SFP-FSCL decoder compared with RPA-based decoding [18], [19] for the RM codes of length 256 with $r \in \{2,3,4\}$. The list size L of the SFP-FSCL decoder is set to 64 for $\mathcal{RM}(2,8)$, 256 for $\mathcal{RM}(3,8)$, and 128 for $\mathcal{RM}(4,8)$, to have a similar FER performance to the RPA decoder. In addition, the lower bound of the FER performance under ML decoding is also plotted as a reference. The memory consumption in KB of the decoders considered in Fig. 11 is summarized in Table III.

It can be observed from Fig. 11 that at a similar FER performance, the decoding latency of the proposed decoder is significantly smaller than that of the RPA decoder. In addition, the proposed decoder requires a computational complexity that is several orders of magnitude lower

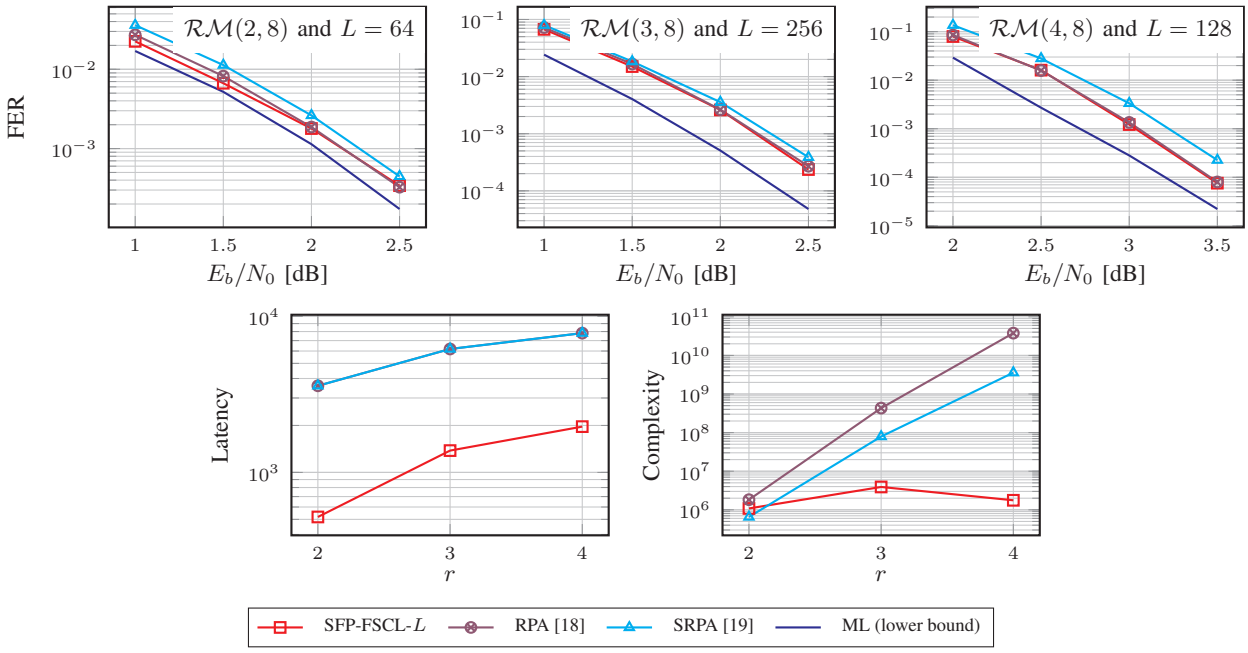


Fig. 11: The error correction performance, decoding latency, and computational complexity of the RPA and the proposed decoders.

TABLE III: Memory requirements in KB of the decoders considered in Fig. 11.

	SFP-FSCL-L	SRPA [19]	RPA [18]
$\mathcal{RM}(2,8)$	69.03 ($L = 64$)	69.80	135.55
$\mathcal{RM}(3,8)$	273.03 ($L = 256$)	281.45	556.80
$\mathcal{RM}(4,8)$	137.03 ($L = 128$)	465.15	922.18

than that of the RPA decoder for $\mathcal{RM}(3,8)$ and $\mathcal{RM}(4,8)$. As seen in Table III, the proposed decoder requires lower memory consumption than the RPA decoder for all the considered RM codes of length 256 that reaches up to 85% for $\mathcal{RM}(4,8)$.

Compared to the SRPA decoder, the proposed decoder obtains an error-correction performance gain of 0.25 dB at the FER of 10^{-3} for $\mathcal{RM}(4,8)$. Moreover, the proposed decoder reduces more than 95% of the computational complexity of the SRPA decoder for $\mathcal{RM}(3,8)$ and 99% of the computational complexity of the SRPA decoder for $\mathcal{RM}(4,8)$. In addition, a memory saving of up to 71% is recorded for the proposed decoder in comparison with SRPA decoder for $\mathcal{RM}(4,8)$. For $\mathcal{RM}(2,8)$, the SRPA decoder requires 60% of the computational complexity of the proposed decoder at a similar error-correction performance. However, this comes at the

cost of significantly higher decoding latency. Note that the RPA and the SRPA decoders have the same decoding latency since a fully parallel implementation is considered for both of them.

V. Conclusion

In this paper, we proposed a novel successive factor-graph permutation (SFP) selection scheme to significantly improve the error-correction performance of Reed-Muller (RM) codes under fast successive-cancellation (FSC) and FSC list (FSCL) decoding. We performed low-complexity maximum likelihood (ML) decoding on the rich symmetry group of RM codes to select a good cyclic factor-graph permutation of the code on the fly. We showed that for a RM code of length 512 with list size $L = 8$, the proposed SFP-FSCL decoder provides error-correction performance gains of $\{1.12, 0.68, 0.42\}$ dB for code orders $r \in \{2, 3, 4\}$, at the target frame error rate (FER) of 10^{-4} when compared to the factor-graph permutation selection scheme in [16]. In addition, for RM codes of length 512, the proposed SFP-FSCL decoder with $L = 8$ reduces up to 71% of the decoding latency while incurring a maximum overhead of 37% in the computational complexity of the SFP-SCL decoder introduced in [16]. For the third and fourth order RM codes of length 256, the proposed decoder has similar error-correction performance to the state-of-the-art recursive projection-aggregation decoder, while reducing up to 85% of the memory consumption, up to 78% of the decoding latency, and more than 99% of the computational complexity. Future research includes developing a factor-graph permutation selection method that allows error-correction performance improvement over the SFP scheme that only considers cyclic permutations.

References

[1] D. E. Muller, "Application of boolean algebra to switching circuit design and to error detection," *Transactions of the I.R.E. Professional Group on Electronic Computers*, vol. EC-3, no. 3, pp. 6–12, 1954.

[2] I. Reed, "A class of multiple-error-correcting codes and the decoding scheme," *Transactions of the IRE Professional Group on Information Theory*, vol. 4, no. 4, pp. 38–49, 1954.

[3] S. Kudekar, S. Kumar, M. Mondelli, H. D. Pfister, E. aolu, and R. L. Urbanke, "Reedmuller codes achieve capacity on erasure channels," *IEEE Trans. Inf. Theory*, vol. 63, no. 7, pp. 4298–4316, 2017.

[4] R. Pedarsani, S. H. Hassani, I. Tal, and E. Telatar, "On the construction of polar codes," in *IEEE Int. Symp. on Inf. Theory*, 2011, pp. 11–15.

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

[6] I. Tal and A. Vardy, "How to construct polar codes," *IEEE Trans. Inf. Theory*, vol. 59, no. 10, pp. 6562–6582, 2013.

[7] M. Mondelli, S. H. Hassani, and R. Urbanke, "Construction of polar codes with sublinear complexity," in *IEEE Int. Symp. on Inf. Theory*, 2017, pp. 1853–1857.

[8] L. Huang, H. Zhang, R. Li, Y. Ge, and J. Wang, "Reinforcement learning for nested polar code construction," *IEEE Global Commun. Conf.*, pp. 1–6, 2019.

[9] Y. Liao, S. A. Hashemi, J. Cioffi, and A. Goldsmith, "Construction of polar codes with reinforcement learning," *IEEE Global Commun. Conf.*, pp. 1–6, 2020.

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

[11] N. Hussami, S. B. Korada, and R. Urbanke, "Performance of polar codes for channel and source coding," in *IEEE Int. Symp. on Inf. Theory*, 2009, pp. 1488–1492.

[12] G. Schnabl and M. Bossert, "Soft-decision decoding of reed-muller codes as generalized multiple concatenated codes," *IEEE Transactions on Information Theory*, vol. 41, no. 1, pp. 304–308, 1995.

[13] I. Dumer and K. Shabunov, "Soft-decision decoding of reed-muller codes: recursive lists," *IEEE Trans. Inf. Theory*, vol. 52, no. 3, pp. 1260–1266, 2006.

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

[15] A. Balatsoukas-Stimming, M. B. Parizi, and A. Burg, "LLR-based successive cancellation list decoding of polar codes," *IEEE Trans. Signal Process.*, vol. 63, no. 19, pp. 5165–5179, Oct. 2015.

[16] S. A. Hashemi, N. Doan, M. Mondelli, and W. J. Gross, "Decoding reed-muller and polar codes by successive factor graph permutations," in *2018 IEEE 10th International Symposium on Turbo Codes Iterative Information Processing (ISTC)*, 2018, pp. 1–5.

[17] M. H. Ardakani, M. Hanif, M. Ardakani, and C. Tellambura, "Fast successive-cancellation-based decoders of polar codes," *IEEE Trans. Commun.*, vol. 67, no. 7, pp. 4562–4574, 2019.

[18] M. Ye and E. Abbe, "Recursive projection-aggregation decoding of reed-muller codes," *IEEE Trans. Inf. Theory*, vol. 66, no. 8, pp. 4948–4965, 2020.

[19] D. Fathollahi, N. Farsad, S. A. Hashemi, and M. Mondelli, "Sparse multi-decoder recursive projection aggregation for reed-muller codes," in *IEEE Int. Symp. on Inf. Theory*, 2021. [Online]. Available: https://arxiv.org/abs/2011.12882

[20] J. D. Key, T. P. McDonough, and V. C. Mavron, "Reed–muller codes and permutation decoding," *Discrete mathematics*, vol. 310, no. 22, pp. 3114–3119, 2010.

[21] M. Kamenev, Y. Kameneva, O. Kurmaev, and A. Maevskiy, "A new permutation decoding method for reed-muller codes," in *IEEE Int. Symp. on Inf. Theory*, 2019, pp. 26–30.

[22] M. Geiselhart, A. Elkelesh, M. Ebada, S. Cammerer, and S. Ten Brink, "Automorphism ensemble decoding of reedmuller codes," *IEEE Trans. Commun.*, pp. 1–1, 2021.

[23] S. B. Korada, "Polar codes for channel and source coding," Ph.D. dissertation, EPFL, Lausanne, Switzerland, 2009.

[24] A. Elkelesh, M. Ebada, S. Cammerer, and S. ten Brink, "Belief propagation list decoding of polar codes," *IEEE Commun. Letters*, vol. 22, no. 8, pp. 1536–1539, 2018.

[25] N. Doan, S. A. Hashemi, M. Mondelli, and W. J. Gross, "On the decoding of polar codes on permuted factor graphs," *IEEE Global Commun. Conf.*, pp. 1–6, Dec 2018.

[26] N. Doan, S. A. Hashemi, and W. J. Gross, "Decoding polar codes with reinforcement learning," *IEEE Global Commun. Conf.*, pp. 1–6, 2020.

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

[28] E. Arikan, "A survey of reed-muller codes from polar coding perspective," in *IEEE Inf. Theory Work. on Inf. Theory*, 2010, pp. 1–5.

[29] Y. Be'ery and J. Snyders, "Optimal soft decision block decoders based on fast hadamard transform," *IEEE Trans. Inf. Theory*, vol. 32, no. 3, pp. 355–364, 1986.

[30] A. Alamdar-Yazdi and F. R. Kschischang, "A simplified successive-cancellation decoder for polar codes," *IEEE Commun. Lett.*, vol. 15, no. 12, pp. 1378–1380, October 2011.

[31] G. Sarkis, P. Giard, A. Vardy, C. Thibeault, and W. J. Gross, "Fast polar decoders: Algorithm and implementation," *IEEE J. Sel. Areas Commun.*, vol. 32, no. 5, pp. 946–957, April 2014.

[32] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to algorithms*. MIT press, 2009.