

Decoding Reed-Muller and Polar Codes by Successive Factor Graph Permutations

Seyyed Ali Hashemi*, Nghia Doan*, Marco Mondelli[†], Warren J. Gross*

*Department of Electrical and Computer Engineering, McGill University, Montréal, Québec, Canada

[†]Department of Electrical Engineering, Stanford University, California, USA

Email: seyyed.hashemi@mail.mcgill.ca, nghia.doan@mail.mcgill.ca, mondelli@stanford.edu, warren.gross@mcgill.ca

Abstract—Reed-Muller (RM) and polar codes are a class of capacity-achieving channel coding schemes with the same factor graph representation. Low-complexity decoding algorithms fall short in providing a good error-correction performance for RM and polar codes. Using the symmetric group of RM and polar codes, the specific decoding algorithm can be carried out on multiple permutations of the factor graph to boost the error-correction performance. However, this approach results in high decoding complexity. In this paper, we first derive the total number of factor graph permutations on which the decoding can be performed. We further propose a successive permutation (SP) scheme which finds the permutations on the fly, thus the decoding always progresses on a single factor graph permutation. We show that SP can be used to improve the error-correction performance of RM and polar codes under successive-cancellation (SC) and SC list (SCL) decoding, while keeping the memory requirements of the decoders unaltered. Our results for RM and polar codes of length 128 and rate 0.5 show that when SP is used and at a target frame error rate of 10^{-4} , up to 0.5 dB and 0.1 dB improvement can be achieved for RM and polar codes respectively.

Index Terms—Reed-Muller Codes, Polar Codes, Factor Graph Permutations.

I. INTRODUCTION

The next generation of wireless communications standard (5G) requires channel coding techniques for the control channel which are of short length. Polar codes [1] are a class of channel coding techniques that were recently selected for the enhanced mobile broadband (eMBB) control channel of 5G [2]. Initially, polar codes attracted a lot of research attention because of their capacity achieving property at infinite code length, with the low-complexity successive-cancellation (SC) decoding. However, for the 5G control channel applications with short code lengths, SC falls short in providing a reasonable error-correction performance. A SC list (SCL) decoder can reduce the error-correction performance gap between SC and maximum a posteriori (MAP) decoder [3]. However, the error-correction performance of polar codes under MAP decoding is not satisfactory.

Reed-Muller (RM) codes [4], [5] are similar to polar codes in the sense that the generator matrices of both codes are constructed by selecting rows from a Hadamard matrix. The row selection of polar codes minimizes the error probability under SC decoding, while the row selection of RM codes maximizes the minimum distance. As a result, polar codes outperform RM codes under SC decoding and RM codes outperform polar codes under MAP decoding. Recently, it was shown that RM

codes achieve the capacity of a binary erasure channel (BEC) under MAP decoding [6]. Since MAP decoding is practically intractable, sub-optimal decoding algorithms such as SC [7] and SCL [8] are used to decode RM codes. However, SC decoding provides a poor error-correction performance when used to decode RM codes and SCL decoding requires a large list size to achieve a desirable error rate.

The symmetric group of RM and polar codes were used to improve their error-correction performance. In particular, a number of permutations on the factor graph representation of polar and RM codes is selected and the decoding algorithm is performed on them [9], [10]. It was shown in [8] that using the factor graph permutations of RM codes helps boost the error-correction performance of SCL decoding. Moreover, [10] showed that for a polar code of length $N = 2^n$, there are $n!$ factor graph permutations on which the decoding can be performed. To limit the complexity of the decoder, it was suggested in [10] and [11] that only the cyclic shifts or random permutations are selected for decoding. However, the error-correction performance of decoding polar codes under the aforementioned schemes deteriorated significantly in comparison with using all the factor graph permutations. In [12], a method was derived to carefully select the factor graph permutations in order to improve the error-correction performance of polar codes. However, all of these techniques require multiple decoding attempts on different factor graph permutations, which adversely affects the decoder complexity.

In this paper, we first show that by considering the permutations on polar code partitions [13], the number of factor graph permutations is larger than $n!$. We then use these permutations to propose a successive permutation (SP) scheme which can find the suitable permutations during the decoding process on the fly. We show that this approach can significantly improve the performance of RM codes under SC decoding with low complexity overhead. We further apply the proposed SP scheme to the SCL decoder for RM codes and show that SP can help SCL decoder to approach MAP decoding performance with a smaller list size than the SCL decoder alone. We then devise a method to apply SP on SC and SCL decoding of polar codes to improve their error-correction performance. Since the SP scheme enables the decoder to progress on a single factor graph, the memory requirements of the underlying decoder remains unchanged. Our results show that for a code of length 128 and rate 0.5 at a target frame error

rate (FER) of 10^{-4} , SP results in up to 0.5 dB improvement for SC and SCL decoding of RM codes, and up to 0.1 dB improvement for SC and SCL decoding of polar codes.

II. PRELIMINARIES

A. Construction of Polar and RM Codes

A polar code of length $N = 2^n$ with K information bits is denoted as $\mathcal{P}(N, K)$. Similarly, a RM code of length $N = 2^n$ and dimension K is represented as $\mathcal{RM}(N, K)$. Let us denote by $R = \frac{K}{N}$ the rate of the code. The encoding process of polar and RM codes includes a matrix multiplication as

$$\mathbf{x} = \mathbf{u} \mathbf{G}^{\otimes n}, \quad (1)$$

where $\mathbf{x} = \{x_0, x_1, \dots, x_{N-1}\}$ is the sequence of coded bits, $\mathbf{u} = \{u_0, u_1, \dots, u_{N-1}\}$ is the sequence of input bits, and $\mathbf{G}^{\otimes n}$ is the n -th Kronecker power of the matrix $\mathbf{G} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$.

Construction of polar and RM codes relies on the selection of K rows of matrix $\mathbf{G}^{\otimes n}$ to build the generator matrix. This process is equivalent to assigning information bits to the corresponding K elements in \mathbf{u} and setting the remaining $N - K$ bits to 0. In this paper, we denote by the indices in the vector \mathbf{u} which carry information as the information set \mathcal{A} and we call the remaining bit indices as the frozen set \mathcal{A}^c .

In case of polar codes, the selection of information set \mathcal{A} is performed by using the polarization property. It was shown in [1] that as the code length goes to infinity, the bit-channels polarize in the sense that some of them become completely noisy and the others become completely noiseless. The fraction of noiseless bit-channels tends to the capacity of the channel. For finite practical code lengths, the bit-channels are divided into two sets \mathcal{A} and \mathcal{A}^c depending on the number of information bits. It should be noted that the polarization process is dependent on the channel over which the data transmission takes place.

In case of RM codes, the information set \mathcal{A} is generated by selecting the rows of $\mathbf{G}^{\otimes n}$ with largest Hamming weights. For a RM code with minimum distance 2^m , the rows with minimum Hamming weight 2^m are selected. Therefore, the code rate for RM codes is constrained to

$$R_{\text{RM}} = \frac{\sum_{i=0}^m \binom{n}{i}}{N}, \quad (2)$$

where $0 \leq m \leq n$. The advantage of RM codes over polar codes is that the construction of RM codes is channel-independent.

After \mathbf{u} is populated, the matrix multiplication in (1) is performed to generate \mathbf{x} . Subsequently, \mathbf{x} is modulated and transmitted through the channel. In this paper, we consider binary phase-shift keying (BPSK) modulation and binary additive white Gaussian noise (AWGN) channel model with variance σ^2 . Therefore, the received vector $\mathbf{y} = \{y_0, y_1, \dots, y_{N-1}\}$ can be translated into logarithmic likelihood ratio (LLR) domain as $\boldsymbol{\alpha} = \frac{2}{\sigma^2} \mathbf{y}$, where $\boldsymbol{\alpha} = \{\alpha_0, \alpha_1, \dots, \alpha_{N-1}\}$.

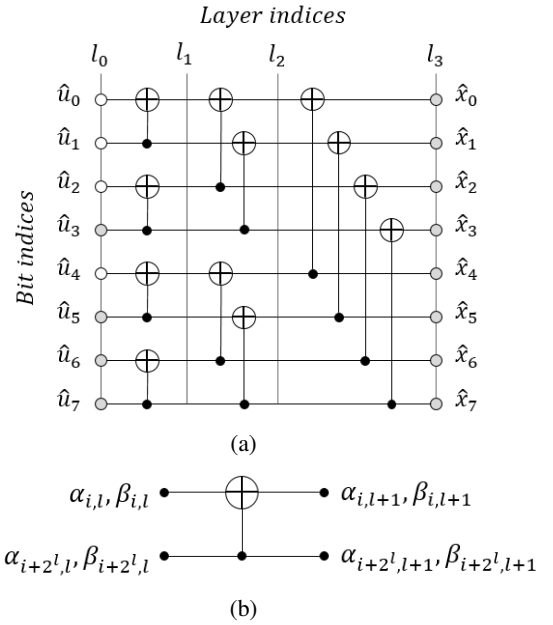


Fig. 1: (a) Factor graph representation of a polar (RM) code with $N = 8$, $K = 4$, and $\{u_0, u_1, u_2, u_4\} \in \mathcal{A}^c$, (b) a processing element (PE).

B. Decoding of Polar and RM Codes

Decoding of polar and RM codes can be represented on a factor graph as shown in Fig. 1a for a code of length $N = 8$ with $K = 4$. The messages are propagated through the processing elements (PEs) as shown in Fig. 1b where $\beta_{i,l}$ denotes a left-to-right message, and $\alpha_{i,l}$ denotes a right-to-left message of the i -th bit index at layer l of the factor graph.

1) *Successive-Cancellation Decoding*: In SC decoding, the right-to-left messages $\alpha_{i,l}$ are soft LLR values and the left-to-right messages $\beta_{i,l}$ are hard decision bits. The messages in the factor graph are updated as

$$\alpha_{i,l} = f(\alpha_{i,l+1}, \alpha_{i+2^l,l+1}), \quad (3)$$

$$\alpha_{i+2^l,l} = g(\alpha_{i,l+1}, \alpha_{i+2^l,l+1}, \beta_{i,l}), \quad (4)$$

$$\beta_{i,l+1} = \beta_{i,l} \oplus \beta_{i+2^l,l}, \quad (5)$$

$$\beta_{i+2^l,l+1} = \beta_{i+2^l,l}, \quad (6)$$

where

$$f(a, b) = 2 \operatorname{arctanh} \left(\tanh \left(\frac{a}{2} \right) \tanh \left(\frac{b}{2} \right) \right) \approx \operatorname{sgn}(a) \operatorname{sgn}(b) \min(|a|, |b|), \quad (7)$$

$$g(a, b, s) = b + (1 - 2s)a, \quad (8)$$

and \oplus is the bitwise XOR operation. SC decoding is initialized by setting $\alpha_{i,n} = y_i$ and the decoding schedule is such that the bits are decoded one by one from u_0 to u_{N-1} . Therefore, the function f is given priority in SC decoding. At layer 0, the elements of \mathbf{u} are estimated as

$$\hat{u}_i = \begin{cases} 0, & \text{if } i \in \mathcal{A}^c \text{ or } \alpha_{i,0} \geq 0, \\ 1, & \text{otherwise.} \end{cases} \quad (9)$$

2) *Successive-Cancellation List Decoding*: SCL decoding improves the error-correction performance of SC decoding by running multiple SC decoders in parallel. Unlike SC, instead of using (9) to estimate \mathbf{u} , each bit is estimated considering both its possible values 0 and 1. Therefore, at each bit estimation, the number of candidates doubles. In order to limit the exponential growth in the complexity of SCL, at each bit estimation a set of L candidates are allowed to survive based on a path metric which is calculated as [14], [15]

$$\text{PM}_{i_\ell} = \sum_{j=0}^i \ln \left(1 + e^{-(1-2\hat{u}_{j_\ell})\alpha_{j,0_\ell}} \right), \quad (10)$$

$$\approx \frac{1}{2} \sum_{j=0}^i \text{sgn}(\alpha_{j,0_\ell}) \alpha_{j,0_\ell} - (1 - 2\hat{u}_{j_\ell}) \alpha_{j,0_\ell}, \quad (11)$$

where ℓ is the path index and \hat{u}_{j_ℓ} is the estimate of bit j at path ℓ .

3) *Factor Graph Permutations*: Factor graph permutations are a way to provide multiple representations of a single code. It was observed in [10] that there exists $n!$ different ways to represent a polar code by permuting the layers in its factor graph. In [12], the permutations in the factor graph layers were translated into permutations in the bit indices. Each permutation is then decoded with a decoding algorithm and the codeword resulted from the decoding of the permutation with the best metric is selected as the output.

III. SUCCESSIVE FACTOR GRAPH PERMUTATIONS

In this section, we first show that the number of permutations in the factor graph representation of polar and RM codes is higher than $n!$. Based on this observation, we introduce the SP scheme.

Consider the setting introduced in [12] in which the bit indices are permuted instead of the layers in the factor graph representation. Let $\mathbf{b}_i = \{b_{n-1_i}, \dots, b_{0_i}\}$ be the binary expansion of the index i and let $\pi_n : \{0, \dots, n-1\} \rightarrow \{0, \dots, n-1\}$ be a permutation of n elements. We show a permutation in the bits of \mathbf{b}_i as $\mathbf{b}_{\pi_n(i)} = \{b_{\pi_n(n-1)_i}, \dots, b_{\pi_n(0)_i}\}$. Clearly, since each index can be represented by n bits, there are $n!$ permutations available. However, polar and RM codes can be represented as the concatenation of polar and RM sub-codes and it was shown in [13] that each sub-code can be decoded independently. We use this idea to permute each sub-code independently. This is depicted for a polar (RM) code with $N = 8$ and $K = 4$ in Fig. 2. At each layer l in the factor graph representation of polar and RM codes, there are 2^{n-l} polar or RM codes of length 2^l . Therefore, the total number of permutations for a polar or RM code can be calculated as

$$\prod_{l=0}^{n-1} (n-l)^{(2^l)}, \quad (12)$$

which quickly grows with n . For example, for a code of length 32, (12) results in 1658880, while for a code of length 128, the result is more than 1.9×10^{27} . Since the memory requirements and the computational complexity of permutation decoding

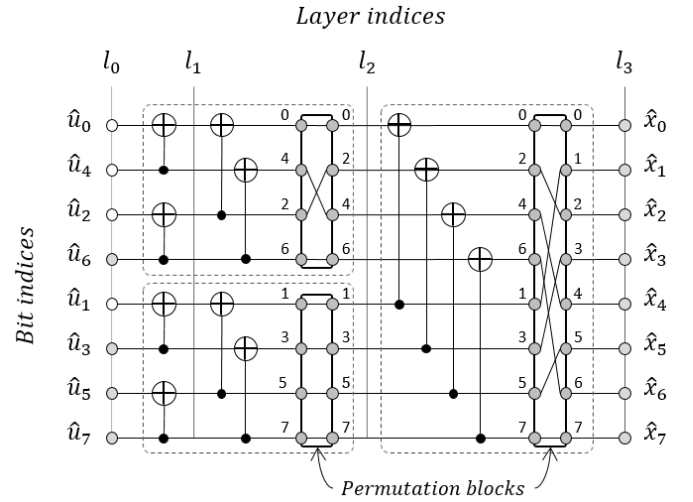


Fig. 2: Permutations on a factor graph representation of a polar (RM) code with $N = 8$, $K = 4$, and $\{u_0, u_1, u_2, u_4\} \in \mathcal{A}^c$.

grows linearly with the number of permutations over which the decoding is performed, decoding over all of the above permutations is impossible.

It should be noted that if all the sub-codes in layer l are permuted similarly, then the total number of permutations reduces to $n!$. In fact, one can represent the permutations in the bit indices as a nested permutation of cyclic shifts at each layer. At layer l in the factor graph, there are l cyclic shift permutations, therefore, the total number of permutations can be calculated as (12). In order to limit the complexity of running multiple decoding attempts on different permutations, we devise a method to induce the best permutations during the course of decoding on the fly.

A. Successive Permutation for SC Decoding

SC decoding is serial in nature where the bits are decoded one by one. The success of SC decoding relies on the correct estimation of bits in the early stages of the decoding process. This is due to the fact that if a bit is estimated incorrectly, there is no way to correct it. The decoding process follows the calculation of LLR values at each decoding layer using f and g functions as in (7) and (8), with priority given to f function. Therefore, it is crucial for SC decoding that the f function provides reliable LLR values for the intermediate layers. Note that the reliability of a LLR value can be measured by taking the absolute value of it. Therefore, we can measure the reliability of a vector of LLR values α of length N as

$$\mathcal{R}(\alpha) = \sum_{i=0}^{N-1} |\alpha_i|. \quad (13)$$

A larger $\mathcal{R}(\alpha)$ indicates a more reliable LLR vector.

In order to find the most reliable LLR vector resulting from the permutations, we successively permute the layers, considering only the cyclic shifts. We then pick the permutation with which the f function results in the most reliable LLR values.

Let us consider the vector of LLR values resulting from the f function at layer l as $\alpha_l^f = \{\alpha_{0,l}, \alpha_{1,l}, \dots, \alpha_{2^l-1,l}\}$. The reliability of α_l^f can be computed as

$$\mathcal{R}(\alpha_l^f) = \sum_{i=0}^{2^l-1} |\alpha_{i,l}| = \sum_{i=0}^{2^l-1} |f(\alpha_{i,l+1}, \alpha_{i+2^l,l+1})|. \quad (14)$$

Now consider all the cyclic shift permutations π_{l+1} at layer $l+1$. We find the permutation P_{l+1} at layer $l+1$ which results in the largest $\mathcal{R}(\alpha_l^f)$ in accordance with

$$P_{l+1} = \arg \max_{p_{l+1} \in \pi_{l+1}} \sum_{i=0}^{2^l-1} |f(\alpha_{p_{l+1}(i),l+1}, \alpha_{p_{l+1}(i+2^l),l+1})|. \quad (15)$$

The process is successively repeated at each layer whenever the f function calculation is performed. The permutation is then fixed at that layer and is used whenever the corresponding g function calculation is conducted. Note that at each layer l , there are only l cyclic shift permutations to be checked in (15). Since only one permutation is selected at each layer, SP algorithm does not change the memory requirements of SC. However, it will add to the computational complexity of SC by a factor of n , in order to perform (15). Moreover, if a parallel implementation of (15) is considered, the latency of SC in terms of the number of time steps [1] remains unchanged, while for a serial implementation, the latency of SC is increased by a factor of n . We now analyse the effect of SP for SC decoding in case of RM and polar codes.

1) *RM Codes*: The rate of RM codes is limited to those in (2). In fact, (2) guarantees that all of the rows in $\mathbf{G}^{\otimes n}$ which have the same Hamming weight are either included in the information set \mathcal{A} or the frozen set \mathcal{A}^c . On the other hand, the permutations in each layer only permutes the bit indices which correspond to the same Hamming weight. Therefore in RM codes, permutations do not change the location of information and frozen bits as seen by the SC decoder. In other words, while the order of the bits that are decoded by SC is changed, the frozen/information bit pattern remains unaltered. This specifically results in significant improvement in error-correction performance when SP is used with SC decoding on RM codes. This is due to the fact that while the code remains unchanged, the bits which are more likely to be decoded correctly are decoded in the earlier stages of the SC decoding process.

2) *Polar Codes*: Unlike RM codes, polar codes are constructed based on reliabilities of bit indices under SC decoding. In other words, the frozen/information bit pattern of polar codes is optimized to result in the best error-correction performance under SC decoding for a specific channel condition. Since the bit indices of information bits are not selected based on their Hamming weights, the permutations may result in a different frozen/information bit pattern which may not be optimized for the specific channel on which the transmission takes place. This in turn may result in error-correction performance loss when SP is used along with SC. In order to resolve this issue, we only apply SP on the sub-codes of polar

codes which are RM codes. For decoding the rest of the code, we use the conventional SC decoding. This will guarantee that the frozen/information bit pattern remains unchanged in case of polar codes.

B. Successive Permutation for SCL decoding

SCL decoding with list size L can be considered as L different SC decoders which are running in parallel and at each bit estimation, L candidates are allowed to survive. In order to use SP with SCL decoding, we apply SP to each of the L SC decoders in a SCL decoder. More formally, consider the vector of LLR values resulting from the f function at layer l for the ℓ -th path in the SCL decoder as $\alpha_{l\ell}^f = \{\alpha_{0,l\ell}, \alpha_{1,l\ell}, \dots, \alpha_{2^l-1,l\ell}\}$. For each path ℓ , we find the permutation $P_{l+1\ell}$ at layer $l+1$ which results in the largest $\mathcal{R}(\alpha_{l\ell}^f)$ in accordance with

$$P_{l+1\ell} = \arg \max_{p_{l+1\ell} \in \pi_{l+1\ell}} \sum_{i=0}^{2^l-1} |f(\alpha_{p_{l+1\ell}(i),l+1\ell}, \alpha_{p_{l+1\ell}(i+2^l),l+1\ell})|, \quad (16)$$

where $\pi_{l+1\ell}$ represents all the cyclic shift permutations for path ℓ at layer $l+1$. Consider after SP, bit i at path ℓ is permuted to a new position $\pi_{l+1\ell}^\ell(i)$. The path metric associated with the ℓ -th path when SP is applied to SCL decoding can be calculated as

$$\begin{aligned} \text{PM}_{i\ell} &= \sum_{j=0}^i \ln \left(1 + e^{-(1-2\hat{u}_{\pi_{l+1\ell}^\ell(j)})\alpha_{\pi_{l+1\ell}^\ell(j),0}} \right), \\ &\approx \frac{1}{2} \sum_{j=0}^i \text{sgn}(\alpha_{\pi_{l+1\ell}^\ell(j),0})\alpha_{\pi_{l+1\ell}^\ell(j),0} - (1-2\hat{u}_{\pi_{l+1\ell}^\ell(j)})\alpha_{\pi_{l+1\ell}^\ell(j),0}, \end{aligned} \quad (17)$$

where ℓ is the path index and $\hat{u}_{j\ell}$ is the estimate of bit j at path ℓ .

We will use the same considerations when using SP for SCL decoding in case of polar and RM codes, i.e., since permutations do not change the frozen/information bit pattern in RM codes, we directly use SP in SCL decoding for RM codes and only apply SP on the sub-codes of polar codes which are RM codes.

IV. RESULTS

In this section, we evaluate the application of SP on SC and SCL decoding for RM and polar codes. The simulations in this section are carried out for $\mathcal{RM}(128, 64)$ and $\mathcal{P}(128, 64)$, where the polar code is constructed for $\text{SNR} = 6$ dB. The results are reported based on FER. The SC decoder with SP is denoted as SPSC and the SCL decoder with SP is denoted as SPSCSCL. SCL(L) and SPSCSCL(L) denote the SCL and SPSCSCL decoders with list size L respectively.

A. RM Codes

Fig. 3 shows the effect of applying SP on SC and SCL decoding when RM codes are used. It can be seen that SPSC outperforms SC by around 0.5 dB at a target FER of 10^{-4} . In addition, SP brings significant improvements when applied to SCL decoding. For example, SPSCSCL(4) slightly outperforms

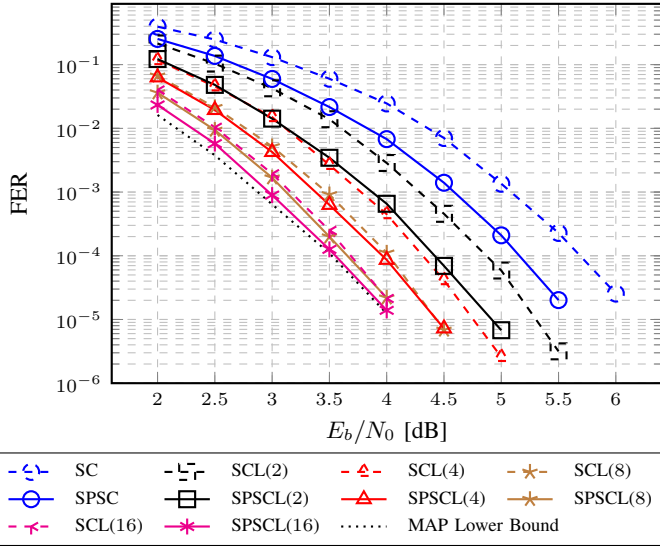


Fig. 3: FER performance of $\mathcal{RM}(128, 64)$ under SC, SPSC, SCL, and SPSC decoding.

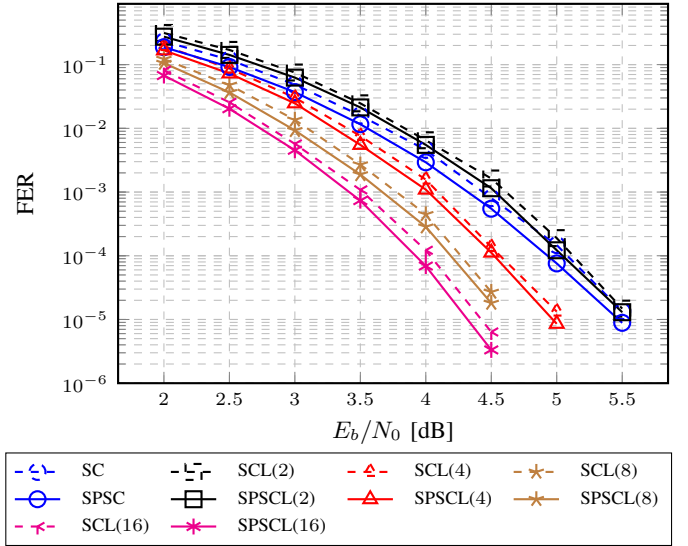


Fig. 4: FER performance of $\mathcal{P}(128, 64)$ under SC, SPSC, SCL, and SPSC decoding.

SCL(8) at a target FER of 10^{-4} and SPSC(16) provides an FER performance which is less than 0.05 dB away from the MAP lower bound at FER of 10^{-4} .

B. Polar Codes

Fig. 4 illustrates the effect of applying SP on SC and SCL decoding when polar codes are used. In case of SCL and SPSC decoders, an 11-bit cyclic redundancy check (CRC) is used and 53 information bits are present to ensure the rate that the decoder sees remains at $\frac{1}{2}$. It can be seen that the FER performance improvement brought by applying SP is around 0.1 dB at a target FER of 10^{-4} . This improvement is smaller than the case of RM codes since we only apply SP on the sub-codes of polar codes which are RM codes. In fact for polar codes, SP is more beneficial when the polar code has many RM sub-codes. Therefore, the performance of SP for polar codes is dependent on the polar codes construction.

V. CONCLUSION

In this paper, we first derived the total number of factor graph permutations for Reed-Muller (RM) and polar codes. We then proposed successive factor graph permutations for RM and polar codes which can find the suitable factor graph permutations on the fly. We showed that the proposed successive permutation (SP) scheme significantly improves the error-correction performance of successive-cancellation (SC) and SC list (SCL) decoders. Our results for RM and polar codes of length 128 and rate 0.5 show that by using SP, the error-correction performance of RM and polar codes under SC and SCL decoding can be improved by up to 0.5 dB and 0.1 dB respectively, at a target frame error rate of 10^{-4} . Moreover, SP enables RM code of length 128 and rate 0.5 to approach within 0.05 dB of the maximum a posteriori (MAP) lower bound with SCL decoder of list size 16.

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] 3GPP, "Final report of 3GPP TSG RAN WG1 #87 v1.0.0," http://www.3gpp.org/ftp/tsg_ran/WG1_RL1/TSGR1_87/Report/Final_Minutes_report_RAN1%2387_v100.zip, Nov. 2016, Reno, USA.
- [3] I. Tal and A. Vardy, "List decoding of polar codes," *IEEE Trans. Inf. Theory*, vol. 61, no. 5, pp. 2213–2226, Mar. 2015.
- [4] D. E. Muller, "Application of boolean algebra to switching circuit design and to error detection," *Trans. of the IRE Prof. Group on Electron. Comput.*, vol. EC-3, no. 3, pp. 6–12, Sep. 1954.
- [5] I. Reed, "A class of multiple-error-correcting codes and the decoding scheme," *Trans. of the IRE Prof. Group on Inf. Theory*, vol. 4, no. 4, pp. 38–49, Sep. 1954.
- [6] S. Kudekar, S. Kumar, M. Mondelli, H. D. Pfister, E. Şaşoğlu, and R. L. Urbanke, "Reed-Muller codes achieve capacity on erasure channels," *IEEE Trans. Inf. Theory*, vol. 63, no. 7, pp. 4298–4316, Jul. 2017.
- [7] I. Dumer, "Recursive decoding and its performance for low-rate Reed-Muller codes," *IEEE Trans. Inf. Theory*, vol. 50, no. 5, pp. 811–823, May 2004.
- [8] 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, Mar. 2006.
- [9] J. D. Key, T. McDonough, and V. Mavron, "Reed-Muller codes and permutation decoding," *Discrete Mathematics*, vol. 310, no. 22, pp. 3114–3119, 2010, combinatorics 2008.
- [10] S. B. Korada, "Polar codes for channel and source coding," Ph.D. dissertation, EPFL, Lausanne, Switzerland, 2009.
- [11] A. Elkelesh, M. Ebada, S. Cammerer, and S. ten Brink, "Belief propagation decoding of polar codes on permuted factor graphs," in *IEEE Wireless Commun. and Networking Conf. (WCNC)*, Apr. 2018, pp. 1–6.
- [12] N. Doan, S. A. Hashemi, M. Mondelli, and W. J. Gross, "On the Decoding of Polar Codes on Permuted Factor Graphs," *ArXiv e-prints*, Jun. 2018. [Online]. Available: <https://arxiv.org/abs/1806.11195>
- [13] S. A. Hashemi, M. Mondelli, S. H. Hassani, C. Condo, R. L. Urbanke, and W. J. Gross, "Decoder partitioning: Towards practical list decoding of polar codes," *IEEE Trans. Commun.*, vol. 66, no. 9, pp. 3749–3759, Sep. 2018.
- [14] 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.
- [15] S. A. Hashemi, C. Condo, and W. J. Gross, "A fast polar code list decoder architecture based on sphere decoding," *IEEE Trans. Circuits Syst. I*, vol. 63, no. 12, pp. 2368–2380, Dec. 2016.