Relaxed Decoding of Polar Codes with Large Kernels

Peter Trifonov, *Member, IEEE*

***Abstract*—A simplified decoding method for polar codes with large kernels is presented. The proposed approach consists in decoder-side substitution of some submatrices of the polarizing transformation with matrices, which admit simpler evaluation of log-likelihood ratios. This approach enables complexity reduction for the successive cancellation, list and sequential decoding algorithms.**

***Index Terms*—Polar codes, large kernels, fast decoding.**

I. Introduction

Polar codes are a novel class of error correcting codes, which was already adopted for use in 5G systems [1]. Although polar codes achieve the capacity of many kinds of communication channels, the performance of Arikan polar codes improves very slowly with their length [2], i.e. these codes have very high scaling exponent. Polar codes with large kernels were shown to have scaling exponent, which converges to its optimal value 2 with kernel dimension growing to infinity [3], [4]. Several constructions of polarization kernels are available [5], [6], [7], [8]. For some of these kernels, very efficient processing techniques, i.e. algorithms for computing the LLRs arising in the successive cancellation (SC) decoder, are available [9], [10]. These algorithms enable list SC decoding [11] of the corresponding polar codes with lower complexity compared to Arikan polar (sub)codes with the same performance.

In this paper, we present a method for further reduction of decoding complexity of polar codes with large kernels. The idea is to replace at the decoder side some parts of the polarizing transformation with matrices, which admit simpler evaluation of LLRs arising in the SC algorithm.

II. Background

A. Polar codes

An $(n=l^m,k)$ polar code is a linear block code generated by k rows of matrix $G_m=M^{(l,m)}K^{\otimes m}$, where $\otimes m$ is m-fold Kronecker product of matrix with itself, K is a $l\times l$ polarization kernel, and $M^{(l,m)}$ is a digit-reversal permutation matrix, corresponding to mapping $\sum_{i=0}^{m-1}t_il^i\rightarrow\sum_{i=0}^{m-1}t_{m-1-i}l^i$, $t_i\in[l]$, where $[l]=\{0,\ldots,l-1\}$. The encoding scheme is given by $c_0^{n-1}=u_0^{n-1}G_m$, where $u_i,i\in\mathcal{F}$ are set to some pre-defined values, e.g. zero (frozen symbols), $|\mathcal{F}|=n-k$, and the remaining values u_i are set to the payload data.

P. Trifonov is with ITMO University, Saint-Petersburg 197101, Russia. E-mail: pvtrifonov@itmo.ru. This work was supported by the Government of Russian Federation (grant 08-08).

The approximate LLRs were defined in [9], [12] as

$$\mathbf{S}_m^{(i)}(u_0^{i-1},y_0^{n-1})=\ln\frac{\mathbf{W}_m^{(i)}(u_0^{i-1}.0|y_0^{n-1})}{\mathbf{W}_m^{(i)}(u_0^{i-1}.1|y_0^{n-1})},\tag{1}$$

where

$$\mathbf{W}_m^{(sl+t)}(u_0^{sl+t}|y_0^{n-1})=\max_{u_{sl+t+1}^{l(s+1)-1}}\prod_{j=0}^{l-1}\mathbf{W}_{m-1}^{(s)}(\theta_K[u_0^{l(s+1)-1},j]|y_{j\frac{n}{l}}^{(j+1)\frac{n}{l}-1})\tag{2}$$

$t\in[s]$, $\theta_K[u_0^{(s+1)l-1},j]_r=(u_{lr}^{l(r+1)-1}K)_j, r\in[s+1]$, and $\mathbf{W}_0^{(0)}(u|y)$ is the probability of symbol u being sent over the channel given y at its output. In particular, for $m=1$ one has

$$\mathbf{W}_1^{(t)}(u_0^t|y_0^{l-1})=\max_{u_{t+1}^{l-1}}\prod_{j=0}^{l-1}\mathbf{W}_0^{(0)}((u_0^{l-1}K)_j|y_j).\tag{3}$$

Assuming that all symbols u_i take equiprobable binary values, one can see that computing this expression reduces to maximum likelihood decoding in a coset of the code generated by rows $t+1,\ldots,l-1$ of kernel K, while the particular coset representative is given by a linear combination of its remaining rows.

For the special case of the iterated Arikan kernel $\mathbb{A}_\tau=F_1^{\otimes\tau}$, where $F_1=\begin{pmatrix}1&0\\1&1\end{pmatrix}$, one has $\mathbf{S}_1^{(i)}(u_0^{i-1},y_0^{l-1})=\mathbb{S}_\tau^{(i)}(u_0^{i-1},y_0^{l-1})$, where

$$\mathbb{S}_\tau^{(2i)}(u_0^{2i-1},y_0^{l-1})=a\boxplus b,\tag{4}$$

$$\mathbb{S}_\tau^{(2i+1)}(u_0^{2i},y_0^{l-1})=(-1)^{u_{2i}}a+b,\tag{5}$$

$a=\mathbb{S}_{\tau-1}^{(i)}(u_{0,e}^{2i-1}\oplus u_{0,o}^{2i-1},y_{0,e}^{l-1})$, $b=\mathbb{S}_{\tau-1}^{(i)}(u_{0,o}^{2i-1},y_{0,o}^{l-1})$,

$$a\boxplus b=\operatorname{sgn}(a)\operatorname{sgn}(b)\min(|a|,|b|),$$

and $\mathbb{S}_0^{(0)}(y)=\mathbf{S}_0^{(0)}(y)$.

At the receiver side, one can successively estimate the input symbols of the polarizing transformation as

$$\widehat{u}_i=\begin{cases}1, & \mathbf{S}_m^{(i)}(\widehat{u}_0^{i-1}|y_0^{n-1})<0, i\notin\mathcal{F},\\ 0, & \mathbf{S}_m^{(i)}(\widehat{u}_0^{i-1}|y_0^{n-1})\geq 0, i\notin\mathcal{F},\\ 0 & i\in\mathcal{F}.\end{cases}\tag{6}$$

This is known as the successive cancellation (SC) decoding algorithm. Its complexity is $O(n\log_l n)$ operations of computing (2). Although efficient algorithms for computing $\mathbf{S}_m^{(i)}$ are available for some kernels, they are still more complex compared to the case of Arikan kernel.

B. Generalized concatenated codes

A generalized concatenated code (GCC) [13], [14] over $\mathbb{F}_2$ is defined using a family of nested inner (n, k_i, d_i) codes $\mathcal{C}_i : \mathcal{C}_0 \supset \mathcal{C}_1 \supset \cdots \supset \mathcal{C}_{\nu-1}$, and a family of outer $(\mathbf{N}, \mathbf{K}_i, \mathbf{D}_i)$ codes $\mathbf{C}_i$, where the i-th outer code is defined over $\mathbb{F}_{2^{k_i - k_{i+1}}}, i \in [\nu], k_\nu = 0$. It is assumed here that $k_i = k_{i+1} + 1, \nu = n$. Let $\mathcal{G}$ be a $n \times n$ matrix, such that its rows $i, \ldots, n-1$ generate code $\mathcal{C}_i$. GCC encoding is performed as follows. First, partition a data vector into n blocks of size $\mathbf{K}_i, i \in [n]$. Second, encode these blocks with codes $\mathbf{C}_i$ to obtain codewords $(\tilde{c}_{i,0}, \ldots, \tilde{c}_{i,\mathbf{N}-1})$. Finally, multiply vectors $(\tilde{c}_{0,j}, \ldots, \tilde{c}_{n-1,j}), j \in [\mathbf{N}]$, by $\mathcal{G}$ to obtain a GCC codeword $(c_{0,0}, \ldots, c_{n-1,0}, c_{0,1}, \ldots, c_{n-1,\mathbf{N}-1})$. A GCC generator matrix can be obtained as

$$G = \begin{pmatrix} G^{(0)} \otimes \mathcal{G}_{0,-} \\ G^{(1)} \otimes \mathcal{G}_{1,-} \\ \vdots \\ G^{(n-1)} \otimes \mathcal{G}_{n-1,-} \end{pmatrix},$$

where $G^{(i)}$ is a generator matrix of $\mathbf{C}_i$, and $\mathcal{G}_{i,-}$ denotes the i-th row of $\mathcal{G}$. It is possible to show that this encoding method results in a $(\mathbf{N}n, \sum_{i=0}^{n-1} \mathbf{K}_i, \geq \min_i d_i \mathbf{D}_i)$ linear block code.

Polar codes are known to be instances of generalized concatenated codes. Indeed, a polar code given by a polarizing transformation $G_m = M^{(l,m)} K^{\otimes m}$, where K is a kernel of dimension l, and frozen set $\mathcal{F}$ can be considered as a generalized concatenated code with inner $(l^{m-\mu}, l^{m-\mu} - i), i \in [l^{m-\mu}]$ polar codes, given by the polarizing transformation $G_{m-\mu}$ and frozen sets $\mathcal{F}_i = \{0, \ldots, i-1\}$, and outer polar codes of length $\mathbf{N} = l^\mu$ with frozen sets $\hat{\mathcal{F}}_i = \{j \in [l^\mu] | i l^\mu + j \in \mathcal{F}\}$, so that $\mathbf{K}_i = l^\mu - |\hat{\mathcal{F}}_i|$. A generator matrix for the i-th outer code can be obtained by striking out rows of G_μ with indices in $\hat{\mathcal{F}}_i$.

C. Relaxed polar codes

It was suggested in [15] to stop applying the Arikan polarizing transformation to bit subchannels, as soon as they become sufficiently good or sufficiently bad, so that all the symbols which are transmitted over these subchannels are obtained exclusively from unfrozen or frozen symbols, respectively. That is, if for some $j, t \in \mathbb{N}$ one has $j2^t + s \in \mathcal{F}$ or $j2^t + s \notin \mathcal{F}$ for all $s \in [2^t]$, then t last steps of the Arikan polarizing transformation are not applied to $u_{j2^t}^{(j+1)2^t-1}$, and the corresponding submatrix $F_1^{\otimes t}$ of $G_m = M^{(2,m)} F_1^{\otimes m}$, is replaced with the identity matrix. This results in relaxed polar codes, which admit simpler encoding and decoding procedures. In this paper we generalize this idea to the case of large kernels.

III. Relaxed decoding

Polar codes are known to have frozen set configurations containing large blocks of adjacent frozen or unfrozen symbols. We propose to simplify the associated decoder computations, by replacing at the receiver side the corresponding pieces of the polarizing transformation with simpler ones. This approach is similar to relaxed polar codes [15], except that we do not require any changes at the transmitter side. The replacements need to be done in such way, so that they do not affect the performance. The specific replacement rules are discussed below.

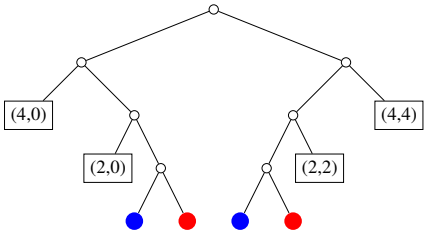

Fig. 1: Decomposition tree for $(16, 8)$ Arikan polar code.

A. Identity replacement

Consider a GCC representation of the original polar code with outer codes of length $\mathbf{N} = l^\mu$. If for some $i \in [l^{m-\mu}]$ one has $\mathbf{K}_i = 0$ or $\mathbf{K}_i = \mathbf{N}$, then such outer code $\mathbf{C}_i$ can be considered as a polar code with a "kernel" given by $l^\mu \times l^\mu$ identity matrix[1]. Instead of computing LLRs $\mathbf{S}_m^{(il^\mu+j)}, j \in [l^\mu]$, one can directly make decisions on the symbols $\widetilde{c}_{i,j}, j \in [\mathbf{N}]$, of these codewords. In the case of $\mathbf{K}_i = 0$ one can unconditionally set $\widetilde{c}_{i,j} = 0$, while for $\mathbf{K}_i = \mathbf{N}$ the values of $\widetilde{c}_{i,j}$ can be obtained as hard decisions based on the sign bits of the corresponding LLRs $\mathbf{S}_{m-\mu}^{(i)}$ from an intermediate layer of the polarizing transformation.

These replacements can be done simultaneously for different values of μ. Their application results in a decoding algorithm similar to [16], where simplified processing of rate 0 and rate 1 nodes is used.

Example 1. *Consider $(16, 8)$ polar code with Arikan kernel F_1 and frozen set $\mathcal{F} = \{0, 1, 2, 3, 4, 5, 6, 8\}$. For $\mu = 2$, one obtains that outer codes $\mathbf{C}_0$ and $\mathbf{C}_3$ are trivial $(4, 0)$ and $(4, 4)$ codes, respectively. These can be considered as polar codes with "kernel"* $\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$. *For $\mu = 1$, one obtains that $\mathbf{C}_0, \mathbf{C}_1, \mathbf{C}_2$ are also trivial $(2, 0)$ codes, while $\mathbf{C}_5, \mathbf{C}_6, \mathbf{C}_7$ are $(2, 2)$ codes. Figure 1 shows the code decomposition tree with the trivial codes identified for $\mu = 2$ and $\mu = 1$. The nodes shown in blue and red correspond to frozen and unfrozen symbols, i.e. $(1, 0)$ and $(1, 1)$ codes, respectively.*

B. Arikan replacement

Here we consider the codes based on kernels of dimension $l = 2^\tau$.

Many of the large kernels, which were published together with efficient processing algorithms, are somewhat similar to the Arikan matrix [9], [10], and several of the corresponding LLRs are computed using the Arikan recursion (4)–(5). In

[1]Identity matrix does not provide any polarization, but can be still treated in the framework of SC decoding.

this section we show that in some cases the instances of such kernels can be replaced with the Arikan matrix $\mathbb{A}_\tau = F_1^{\otimes\tau}$ of appropriate dimension. This matrix can be considered as a polarization kernel itself, and its processing can be implemented using the equations (4)–(5).

Lemma 1. *Let K be an invertible $l \times l$ matrix, where $l = 2^\tau$. If the rows of K with indices in some set $\mathbf{I}_K \subset [l]$ are obtained by an invertible linear transformation of the rows of $\mathbb{A}_\tau$ with indices in some set $\mathbf{I}' \subset [l]$, then rows of K with indices in $\mathbf{I}_K$ and rows of $\mathbb{A}_\tau$ with indices in $\mathbf{I}'$ generate the same code.*

Proof. The statement is obvious. □

Lemma 2. *Let K be an invertible $l \times l$ matrix, where $l = 2^\tau$. If the columns of K^{-1} with indices in some set $\mathbf{F}_K \subset [l]$ are obtained by an invertible linear transformation of the columns of $\mathbb{A}_\tau^{-1}$ with indices in some set $\mathbf{F}' \subset [l]$, then rows of K with indices in $\mathbf{I}_K = [l] \setminus \mathbf{F}_K$ and rows of $\mathbb{A}_\tau$ with indices in $\mathbf{I}'_K = [l] \setminus \mathbf{F}'$ generate the same code.*

Proof. Let codes $\mathcal{C}_1$ and $\mathcal{C}_2$ be generated by rows of K and $\mathbb{A}_\tau$ with indices in $\mathbf{I}_K$ and $\mathbf{I}'_K$, respectively. The codewords of these codes can be represented as $c' = x_0^{l-1}K, c' \in C_1$ and $c'' = y_0^{l-1}\mathbb{A}, c'' \in C_2$, where $x_i = 0, i \notin \mathbf{I}_K$ and $y_j = 0, j \notin \mathbf{I}'_K$. Hence, $(c'K^{-1})_i = 0, i \in \mathbf{F}_K$ and $(c'\mathbb{A}_\tau^{-1})_j = 0, j \in \mathbf{F}'_K$. This implies that the transposed check matrices H_1^T and H_2^T of these codes are given by the columns of K^{-1} and $\mathbb{A}_\tau^{-1}$ with indices in $\mathbf{F}_K$ and $\mathbf{F}'_K$, respectively. Since $H_1 = QH_2$ for some invertible matrix Q, these check matrices define the same code. □

For the sake of simplicity, we assume that the linear transformation mentioned in the above lemmas is the trivial one, i.e. certain rows (columns) of K (K^{-1}) are equal to the rows (columns) of $\mathbb{A}_\tau$, and these rows (columns) are located in the same positions. Equation (3) implies that at phase t of the SC algorithm one needs to find the most probable codewords of the code (and its coset) generated by rows $t+1, \dots, l-1$ of kernel $K, 0 \le t < l$. Then SC algorithm selects the most probable value of u_t and proceeds with $u_{t+1}, \dots, u_{l-1}$.

Consider the case when symbols $u_0, \dots, u_{f-1}$ are frozen, and the remaining symbols $u_f, \dots, u_{l-1}$ are unfrozen. If the codes generated by rows $f, \dots, l-1$ of matrices K and $\mathbb{A}_\tau$ are identical, then the most probable codeword selected by the SC algorithm at phase f would be also selected on phases $f+1, \dots, l-1$. This allows one to use simpler LLR calculation techniques available for kernel $\mathbb{A}_\tau$ to fully construct such codeword.

This leads to the following replacement rule. We represent the polar code as a GCC with $\mu = 1$, and check all outer codes $\mathbf{C}_i, 0 \le i < l^{m-1}$ with the corresponding frozen sets $\hat{\mathcal{F}}_i$. If

1) $[l] \setminus \hat{\mathcal{F}}_i \subset \mathbf{I}_K$, where $\mathbf{I}_K$ is given in Lemma 1, or
2) $\hat{\mathcal{F}}_i \subset \mathbf{F}_K$, where $\mathbf{F}_K$ is given in Lemma 2,

then the corresponding instance of K in the polarizing transformation can be replaced with $\mathbb{A}_\tau$, and $\mathbf{C}_i$ can be considered as an Arikan polar code with frozen set $\hat{\mathcal{F}}_i$.

Example 2. *Consider the case of $\mu = 1$ and kernel K shown in Figure 2. It can be seen that rows $11, \dots, 15$ of K and columns $0, \dots, 4$ of K^{-1} (shown in red) are identical to those of $\mathbb{A}_4 = \mathbb{A}_4^{-1}$. This means that for a $(256, 220)$ polar code with frozen set $\mathcal{F} = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14; 16, 17, 18, 19, 20, 21, 22; 32, 33, 34, 36, 37; 48; 64, 65, 66, 68, 69; 80, 81, 82\}$ one can consider outer codes $\mathbf{C}_0, \mathbf{C}_3, \mathbf{C}_5, \dots, \mathbf{C}_{15}$ of length 16 as Arikan polar codes, i.e. the corresponding instances of K in the polarizing transformation can be replaced with $\mathbb{A}_4$. Furthermore, codes $\mathbf{C}_6, \dots \mathbf{C}_{15}$ are generated by identity matrix, i.e. the corresponding instances of K in the polarizing transformation can be replaced with the identity matrix.*

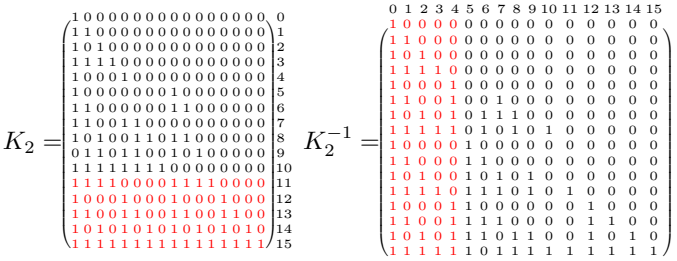

Fig. 2: A 16×16 kernel with rate of polarization 0.51828 and scaling exponent 3.45 [10].

These replacement rules can be applied recursively. To do this, let us consider a GCC, where the inner polar codes of length l are generated by matrix K. If the i-th outer code has rate 0, then one can assume that the i-th input symbol of the inner polar code is fully frozen. Similarly, if the i-th outer code has rate 1, then one can assume that the i-th input symbol of the polar code is fully unfrozen. If the set of fully frozen symbols is a subset of $\mathbf{F}_K$ and the remaining symbols are fully unfrozen, or if the set of fully unfrozen symbols $\widetilde{\mathcal{I}}$ is a subset of $\mathbf{I}_K$ and the remaining symbols are fully frozen, then the inner codes can be assumed to be generated by $\mathbb{A}_\tau$. Indeed, in this case all outer codes have rate 0 or 1, and the generator matrix of the whole GCC is given by $I \otimes K_{\widetilde{\mathcal{I}}}$, where $K_{\widetilde{\mathcal{I}}}$ is a submatrix of K with rows given by $K_{\widetilde{\mathcal{I}}}$. The latter matrix can be transformed by row operations to $\mathbb{A}_{\tau,\widetilde{\mathcal{I}}}$. This means that the same GCC can be generated by $I \otimes \mathbb{A}_{\tau,\widetilde{\mathcal{I}}}$.

Application of the proposed approach results in a different decision vector $\hat{u}_0^{n-1}$ (but the same recovered codeword in the case of successful decoding) of the SC algorithm compared to a straightforward implementation. One can avoid any extra data transformations if systematic encoding is used.

IV. Numeric results

Table I presents the number of summation and comparison operations for SC decoding of some polar codes with Arikan kernel F_1 and large kernels given in [10], as well as the relative reduction of the complexity provided by the proposed approach. It can be seen that most of the complexity reduction is achieved by the identity replacement transformation. By combining it with Arikan replacement, one obtains up to 39% gain in terms of complexity.

TABLE I: SC decoding complexity

Code	G_N	Straightforward		Identity		Identity and Arikan		
		Summ.	Comp.	Summ.	Comp.	Summ.	Comp.	Gain
$(1024, 256)$	$K_3^{\otimes 2}$	200576	232704	144164	167256	129301	150500	0.35
$(1024, 512)$	$K_3^{\otimes 2}$	200576	232704	159834	185436	132913	155480	0.33
$(1024, 768)$	$K_3^{\otimes 2}$	200576	232704	153566	178164	129924	152788	0.35
$(4096, 1024)$	$K_1^{\otimes 3}$	193536	168192	127714	113377	119670	106832	0.37
$(4096, 2048)$	$K_1^{\otimes 3}$	193536	168192	143414	130613	133242	122198	0.29
$(4096, 3072)$	$K_1^{\otimes 3}$	193536	168192	138924	130110	130379	123004	0.30
$(4096, 1024)$	$K_2^{\otimes 3}$	72960	67584	47193	46072	46850	46072	0.34
$(4096, 2048)$	$K_2^{\otimes 3}$	72960	67584	48932	50963	52827	55112	0.23
$(4096, 3072)$	$K_2^{\otimes 3}$	72960	67584	52033	57488	51849	57488	0.22
$(1024, 256)$	$F_1^{\otimes 10}$	5120	5120	3022	3588	3022	3588	0.35
$(1024, 512)$	$F_1^{\otimes 10}$	5120	5120	3340	4452	3340	4452	0.24
$(1024, 768)$	$F_1^{\otimes 10}$	5120	5120	3205	4676	3205	4676	0.23
$(4096, 1024)$	$F_1^{\otimes 12}$	24576	24576	13599	16082	13599	16082	0.39
$(4096, 2048)$	$F_1^{\otimes 12}$	24576	24576	15805	20292	15805	20292	0.27
$(4096, 3072)$	$F_1^{\otimes 12}$	24576	24576	15087	21682	15087	21682	0.25

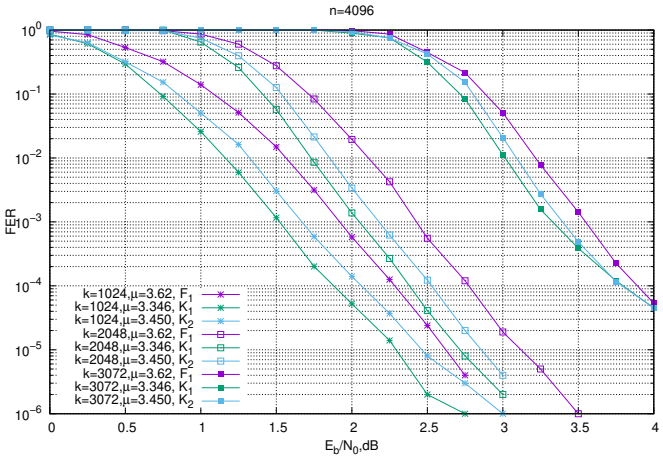

Fig. 3: SC performance of the codes considered in Table I

The performance of the considered codes under SC decoding is reported in Figure 3. As it may be expected, codes based on the kernels with lower scaling exponent provide better performance.

Figure 4 presents the performance of $(4096, 2048)$ polar (sub)codes based on kernel K_2 [10] with conventional and relaxed decoding. It can be seen that employing relaxed decoding does not incur any performance loss. Due to better distance properties, polar subcodes do outperform polar codes.

V. Conclusions

In this paper a simplified decoding method for polar codes with large kernels was suggested. It consists in selective replacement of the instances of large kernel in the polarizing transformation with simpler matrices. The proposed approach does not require any changes at the encoder side and does not affect the decoder performance. It can be also used together with list [11] and sequential [17] decoding algorithms.

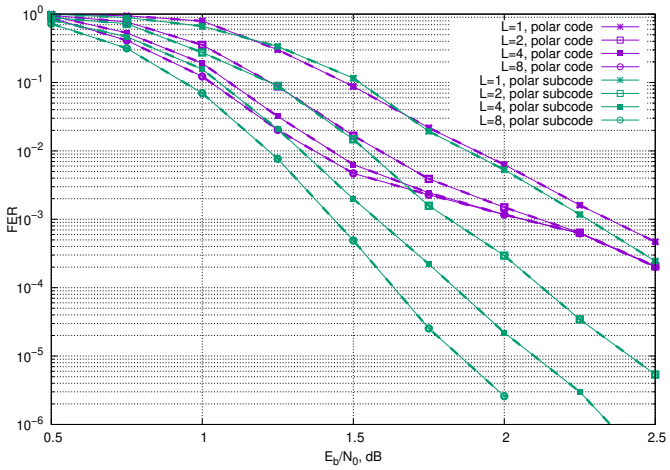

Fig. 4: Performance of list SC decoding with conventional (solid lines) and relaxed (dashed lines) decoding

References

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

[2] M. Mondelli, R. Urbanke, and S. H. Hassani, "Unified scaling of polar codes: Error exponent,scaling exponent, moderate deviations, and error floors," in *Proceedings of IEEE International Symposium on Information Theory*, 2015.

[3] S. B. Korada, E. Sasoglu, and R. Urbanke, "Polar codes: Characterization of exponent, bounds, and constructions," *IEEE Transactions on Information Theory*, vol. 56, no. 12, pp. 6253–6264, December 2010.

[4] A. Fazeli, S. H. Hassani, M. Mondelli, and A. Vardy, "Binary linear codes with optimal scaling: Polar codes with large kernels," in *Proceedings of IEEE Information Theory Workshop*, 2018.

[5] N. Presman, O. Shapira, S. Litsyn, T. Etzion, and A. Vardy, "Binary polarization kernels from code decompositions," *IEEE Transactions On Information Theory*, vol. 61, no. 5, May 2015.

[6] H.-P. Lin, S. Lin, and K. A. Abdel-Ghaffar, "Linear and nonlinear binary kernels of polar codes of small dimensions with maximum exponents," *IEEE Transactions On Information Theory*, vol. 61, no. 10, October 2015.

[7] G. Trofimiuk and P. Trifonov, "Construction of binary polarization kernels for low complexity window processing," in *Proceedings of IEEE Information Theory Workshop*, 2019.

[8] F. Abbasi and E. Viterbo, "Large kernel polar codes with efficient window decoding," *IEEE Transactions On Vehicular Technology*, vol. 69, no. 11, November 2020.

[9] G. Trofimiuk and P. Trifonov, "Efficient decoding of polar codes with some 16×16 kernels," in *Proceedings of IEEE Information Theory Workshop*, 2018.

[10] G. Trofimiuk and P. Trifonov, "Reduced complexity window processing of binary polarization kernels," in *Proceedings of IEEE International Symposium on Information Theory*, Paris, France, July 2019.

[11] I. Tal and A. Vardy, "List decoding of polar codes," *IEEE Transactions On Information Theory*, vol. 61, no. 5, pp. 2213–2226, May 2015.

[12] V. Miloslavskaya and P. Trifonov, "Sequential decoding of polar codes with arbitrary binary kernel," in *Proceedings of IEEE Information Theory Workshop*. Hobart, Australia: IEEE, 2014, pp. 377–381.

[13] E. Blokh and V. Zyablov, "Coding of generalized concatenated codes," *Problems of Information Transmission*, vol. 10, no. 3, pp. 45–50, 1974.

[14] V. Zyablov, S. Shavgulidze, and M. Bossert, "An introduction to generalized concatenated codes," *European transactions on telecommunications*, vol. 10, no. 6, pp. 609–622, 1999.

[15] M. El-Khamy, H. Mahdavifar, G. Feygin, J. Lee, and I. Kang, "Relaxed polar codes," *IEEE Transactions On Information Theory*, vol. 63, no. 4, April 2017.

[16] G. Sarkis, P. Giard, A. Vardy, C. Thibeault, and W. Gross, "Fast polar decoders: Algorithm and implementation," *IEEE Journal On Selected Areas In Communications*, vol. 32, no. 5, May 2014.

[17] V. Miloslavskaya and P. Trifonov, "Sequential decoding of polar codes," *IEEE Communications Letters*, vol. 18, no. 7, pp. 1127–1130, 2014.