On Construction of Polar Subcodes with Large Kernels

Peter Trifonov
ITMO University
Email: pvtrifonov@corp.ifmo.ru

***Abstract*—Polar subcodes with large kernels were shown to require lower complexity to achieve the same performance as Arikan polar codes under SCL decoding. In this paper we present design techniques for polar (sub)codes with large kernels. Namely, methods are presented to estimate the capacities of bit subchannels, as well as to eliminate low-weight non-zero codewords from the obtained codes.**

I. INTRODUCTION

The discovery of polarization phenomenon by E. Arikan lead to construction of polar codes, which achieve the symmetric capacity of a binary-input discrete memoryless channel W, have low-complexity construction, encoding and decoding algorithms [1]. However, the classical Arikan construction polarizes the channel quite slowly. Hence, for codes of practical length one has to transmit some data symbols over mediocre bit subchannels. This results in poor performance of such polar codes under successive cancellation (SC) decoding. Improved code constructions, such as polar codes with CRC [2] and polar subcodes [3], [4] require SC list (SCL) decoding with large list size to obtain good performance.

The situation improves considerably if one replaces the Arikan 2×2 matrix, called kernel, with a larger one. Polar codes with sufficiently large kernels were shown to provide higher polarization rate [5], [6] and achieve optimal scaling exponent [7], [8]. Furthermore, in some cases the SCL decoding complexity needed by polar subcodes with large kernel to achieve some target performance is lower compared to the case of codes with Arikan kernel [9].

Polar codes with Arikan kernel can be constructed by means of density evolution [10], [11], Gaussian approximation [12], binary erasure channel recursion [1], or Monte-Carlo simulations [13]. For larger kernels, only two latter methods are available, except for the case of kernels derived from the Arikan matrix. However, the codes constructed for BEC [14] may not be optimal for the AWGN channel, which is much more important in practice. Furthermore, as shown in [9], obtaining a practical performance-complexity tradeoff with large kernels requires employing SCL decoding with sufficiently large list size. This requires one to take care of both SC and maximum likelihood (ML) decoding error probability of the obtained code.

In this paper an approximate method for computing the reliability of bit subchanels induced by a binary polarizing transformation for AWGN channel is presented. Furthermore, a method is presented for elimination of low-weight non-zero (LWNZ) codewords from the obtained polar code, which provides substantially reduced ML decoding error probability.

II. BACKGROUND

A. Polar codes

Let F be an $l \times l$ non-singular matrix, such that no column permutation can transform it into an upper triangular matrix [5]. A polarizing transformation is given by matrix $A = B_{l,m}F^{\otimes m}$, where $B_{l,m}$ is a digit-reversal permutation matrix, corresponding to mapping $\sum_{i=0}^{m-1} t_i l^i \to \sum_{i=0}^{m-1} t_{m-1-i} l^i$,$t_i \in [l]$, and $[l] = \{0, 1, , \dots, l-1\}$. This construction can be also extended to the case of mixed-kernel polarizing transformations [15]. It is possible to show that a binary input memoryless symmetric channel $\mathbf{W}(y|c) = \mathbf{W}_0^{(0)}(y|c)$ together with matrix A gives rise to $n = l^m$ bit subchannels

$$\mathbf{W}_m^{(i)}(y_0^{n-1}, u_0^{i-1}|u_i) = \frac{1}{2^{n-1}} \sum_{u_{i+1}^{n-1} \in \mathbb{F}_q^{n-i-1}} \prod_{j=0}^{n-1} \mathbf{W}_0^{(0)}(y_j|(u_0^{n-1}A)_j) \quad (1)$$

with capacities approaching 0 or 1 symbols per channel use, and fraction of noiseless subchannels approaching $I(\mathbf{W})$, the capacity of channel $\mathbf{W}$. Rate of polarization depends on partial distances $D_i, 0 \leq i < l$, where D_i is the minimum distance between the i-th row of F and the space generated by rows $i+1, \dots, l-1$ [5].

An $(n = l^m, k)$ polar code over $GF(2)$ is a set of codewords $c_0^{n-1} = u_0^{n-1}A$, where $u_i = 0, i \in \mathcal{F}$, $\mathcal{F} \subset [n]$ is the set of frozen symbol indices, and $|\mathcal{F}| = n - k$. The classical construction of polar codes assumes that $\mathcal{F}$ is the set of indices of subchannels $\mathbf{W}_m^{(i)}$ with the lowest capacity.

It is convenient to define probabilities

$$\mathbf{W}_m^{(i)}(u_0^i|y_0^{n-1}) = \sum_{u_{i+1}^{n-1}} \prod_{i=0}^{n-1} \mathbf{W}((u_0^{n-1}A)_i|y_i). \quad (2)$$

These probabilities can be recursively computed as

$$\mathbf{W}_\lambda^{(lj+i)} \left\{u_0^{lj+i}|y_0^{N-1}\right\} = \sum_{u_{lj+i+1}^{lj+l-1} \in \mathbb{F}_q^{l-i-1}} \prod_{s=0}^{l-1} \mathbf{W}_{\lambda-1}^{(j)} \left\{(u_{lt}^{lt+l-1}F)_s, 0 \leq t \leq j | y_{\frac{N}{l}s}^{\frac{N}{l}s+\frac{N}{l}-1}\right\}, \quad (3)$$

where $N = l^\lambda$. Computing this expression reduces to soft-output decoding of a (coset of) non-systematic linear code $\mathcal{C}_i$ generated by rows $i, \dots, l-1$ of F. This can be implemented by employing a BCJR-like algorithm over an extended trellis of this code [16].

Decoding of polar codes can be implemented by the successive cancellation algorithm, which makes decisions

$$\widehat{u}_i = \begin{cases} \arg\max_{u_i \in \mathbb{F}_2} \mathbf{W}_m^{(i)}(\widehat{u}_0^{i-1}, u_i|y_0^{n-1}), & i \notin \mathcal{F}, \\ \text{the frozen value of } u_i & i \in \mathcal{F}. \end{cases} \tag{4}$$

It was suggested in [17] to replace the sum in (2) and (3) with the maximal term, i.e. to consider probabilities

$$\begin{aligned} \widetilde{\mathbf{W}}_\lambda^{(lj+i)}(u_0^{lj+i}|y_0^{N-1}) &= \max_{u_{lj+i+1}^{N-1}} \prod_{i=0}^{N-1} W((u_0^{N-1} B_{l,\lambda} F^{\otimes\lambda})_i|y_i) \\ &= \max_{u_{lj+i+1}^{lj+l-1}} \prod_{s=0}^{l-1} \widetilde{\mathbf{W}}_{\lambda-1}^{(j)} \left\{(u_{lt}^{lt+l-1}F)_s, 0 \le t \le j | y_{\frac{N}{l}s}^{\frac{N}{l}s+\frac{N}{l}-1}\right\}. \end{aligned}$$

These probabilities can be used to approximate $\mathbf{W}_m^{(j)}(u_0^{lj+i}|y_0^{N-1})$ in the successive cancellation decoding algorithm. Alternatively, one can employ approximate log-likelihood ratios

$$\mathcal{L}_m^{(lj+i)}(u_0^{lj+i-1}|y_0^{n-1}) = \log \frac{\widetilde{\mathbf{W}}_m^{(lj+i)}(u_0^{lj+i-1}, 0|y_0^{n-1})}{\widetilde{\mathbf{W}}_m^{(lj+i)}(u_0^{lj+i-1}, 1|y_0^{n-1})}. \tag{5}$$

Fast exact and approximate *kernel processing* algorithms can be derived for computing these values [17], [9], [18]. The corresponding decoding algorithms were shown to provide a very good performance-complexity tradeoff.

B. Polar subcodes

Classical polar codes are known to have quite poor minimum distance. Their performance under SCL decoding can be substantially improved by replacing static freezing constraints $u_i = 0, i \in \mathcal{F}$, with dynamic freezing constraints

$$u_i = \sum_{j<i} V_{s_i,j} u_j, i \in \mathcal{F}, \tag{6}$$

where V is a $(n-k) \times n$ constraint matrix, such that distinct rows end[1] in distinct columns $i \in \mathcal{F}$, and s_i is the index of the row ending in column i. Such constraints eliminate many low-weight codewords from a classical polar code. Symbols u_i with at least one term in the r.h.s. of (6) are referred to as dynamic frozen (DFS), and those with $V_{s_i,j} = 0, j < s_i$, are denoted static frozen. The obtained codes are referred to as polar subcodes [3].

C. SCL Decoding

The SC algorithm does not provide ML decoding. Substantially better performance can be obtained by considering at each phase i a few partial input vectors u_0^{i-1}, constructing their possible continuations u_0^i, and selecting for further processing L ones with the highest score $\widetilde{\mathbf{W}}_m^{(i)}(u_0^i|y_0^{n-1})$ [2]. The decoding error probability for such method can be estimated as $P_e(L) \le P_{ML} + P'(L)$, where P_{ML} is the maximum likelihood decoding error probability, and $P'(L)$ is the probability of the score of the correct vector u_0^i becoming less than the scores of L incorrect vectors at some phase i of the SCL algorithm, provided that the ML decoder does not make an error. Tight bounds on P_{ML} are available (see [19] and references therein), which depend on the weight distribution of the code and, in particular, on the number of LWNZ codewords. No estimates are available for $P'(L)$, but simulations show that it decreases with the SC decoding error probability of the considered code. Hence, one should keep both of these terms low to obtain a code well-decodable by the SCL algorithm with small list size.

III. Assessing the reliability of bit subchannels

A. The capacity of coded channels

Let us assume for the sake of simplicity that $m = 1$.
Consider bit subchannel

$$\mathbf{W}_1^{(i)}(Y_0^{l-1}, U_0^{i-1}|U_i) = \frac{1}{2^{l-1}} \sum_{u_{i+1}^{l-1}} \mathbf{W}(Y_0^{l-1}|U_0^{l-1}F)$$

induced by some $l \times l$ kernel F and binary input channel $\mathbf{W}(Y|C)$, where $\mathbf{W}(Y_0^{l-1}|C_0^{l-1}) = \prod_{i=0}^{l-1} \mathbf{W}(Y_i|C_i)$, and U_i, C_i, Y_i are random variables corresponding to input values of the polarizing transformation, channel input and output symbols, respectively. The mutual information of $\mathbf{W}_1^{(i)}$ is given by

$$I_1^{(i)} = I_1(Y_0^{l-1}, U_0^{i-1}; U_i) = I_1(Y_0^{l-1}; U_i|U_0^{i-1}) + I(U_0^{i-1}; U_i).$$

Since the random variables $U_j, 0 \le j < l$, are assumed to be independent, the latter term is equal to 0. Furthermore, by the chain rule of the mutual information, one obtains

$$\begin{aligned} I_1^{(i)} = I_1(Y_0^{l-1}; U_i|U_0^{i-1}) = & I_1(Y_0^{l-1}; U_i^{l-1}|U_0^{i-1}) - \\ & I_1(Y_0^{l-1}; U_{i+1}^{l-1}|U_0^i). \end{aligned} \tag{7}$$

For a symmetric channel, $I_1(Y_0^{l-1}; U_i^{l-1}|U_0^{i-1})$ is independent of U_0^{i-1}, and the latter values can be assumed to be equal to 0. Then $I_1^{[i]} = I_1(Y_0^{l-1}; U_i^{l-1}|U_0^{i-1} = 0)$ becomes the capacity of a channel consisting of an encoder of a linear code $\mathcal{C}_i$ generated by rows $i, \dots, l-1$ of matrix F, and the underlying channel $\mathbf{W}_0^{(0)}$.

From the definition of mutual information, one obtains

$$\begin{aligned} I_1^{[i]} = 2^{i-l} \int_{y_0^{l-1} \in \mathbb{R}^l} & \sum_{u_i^{l-1} \in \mathbb{F}_2^{l-i}} \mathbf{W}(y_0^{l-1}|u_i^{l-1} F^{(i)}) \cdot \\ & \log_2 \frac{2^{l-i} \mathbf{W}(y_0^{l-1}|u_i^{l-1} F_l^{(i)})}{\sum_{v_i^{l-1} \in \mathbb{F}_2^{l-i}} \mathbf{W}(y_0^{l-1}|v_i^{l-1} F_l^{(i)})} dy_0^{l-1} \\ = l - i + \int_{y_0^{l-1} \in \mathbb{R}^l} & \mathbf{W}(y_0^{l-1}|0) \log_2 \frac{\mathbf{W}(y_0^{l-1}|0)}{\sum_{c' \in \mathcal{C}_i} \mathbf{W}(y_0^{l-1}|c')} dy_0^{l-1}, \end{aligned} \tag{8}$$

where $F^{(i)}$ is the matrix consisting of rows $i, \dots, l-1$ of matrix F.

[1] Given some binary vector a_0^{n-1}, we say that it ends in position j iff $a_j = 1$ and $a_t = 0, j < t < n$.

1) Binary symmetric channel: Consider the case of $\mathbf{W}_0^{(0)}$ being a BSC with crossover probability p. It was shown in [20], [21] that in this case

$$I_1(Y_0^{l-1};U_i^{l-1}|U_0^{i-1}=0)=l-i+H(R_i)-nh(p),$$

where $h(p)$ is the binary entropy function, and $H(R_i)$ is the entropy of a random variable corresponding to the row in the standard array which contains the channel output vector. It can be computed as $H(R_i)=-\sum_{j=1}^{2^i}P_{ij}\log\left(P_{ij}\right),$ where $P_{ij}=\sum_{s=0}^{l}w_i(j,s)p^s(1-p)^{l-s}$, and $w_i(j,s)$ is the number of vectors of weight s in the j-th row of the standard array of the code $\mathcal{C}_i$. Hence, one obtains $I_1^{(i)}=1+H(R_i)-H(R_{i+1})$. This enables one to compute the capacities of the subchannels of a single-layer polarizing transformation. Unfortunately, the obtained subchannels are not binary symmetric, so this approach cannot be extended to multi-layer polarizing transformation with $m>1$.

2) Binary erasure channel: If $\mathbf{W}$ is the binary erasure channel with erasure probability $Z_{0,0}$, then all subchannels $\mathbf{W}_\lambda^{(i)}$ are also BEC. Hence, their capacities can be computed as $I_\lambda^{(i)}=1-Z_{\lambda,i}$, where $Z_{\lambda,i}$ is the corresponding erasure probability. It can be computed as $Z_{\lambda,li+j}=\sum_{s=1}^{l}B_s^{(j)}Z_{\lambda-1,i}^s(1-Z_{\lambda-1,i})^{l-s}$, where $B_s^{(j)}$ is the number of erasure patterns of weight s, which cause the 0-th information symbol of code $\mathcal{C}_j$ to be unrecoverable. An efficient algorithm for computing these values is given in [14].

3) EXIT Functions: For general channels, exact evaluation of (8) does not seem to be feasible. However, one can extend to the case of polar codes the semi-analytic EXIT function method, which has been used with great success in the design of LDPC and turbo codes [22].

From (7) one obtains

$$\begin{aligned}I_1^{(i)}=&I_1(Y_0^{l-1},U_0^{i-1};U_i)\\=&1+\int_{y_0^{l-1}\in\mathbb{R}^l}\mathbf{W}(y_0^{l-1}|0)\log_2\frac{\sum_{c\in\mathcal{C}_{i+1}}\mathbf{W}(y_0^{l-1}|c)}{\sum_{c'\in\mathcal{C}_i}\mathbf{W}(y_0^{l-1}|c')}dy_0^{l-1}\\=&1-\int_{y_0^{l-1}\in\mathbb{R}^l}\mathbf{W}(y_0^{l-1}|0)\log_2\left(1+1/\mathcal{R}_i(y_0^{l-1})\right)dy_0^{l-1},\end{aligned}\tag{9}$$

where

$$\begin{aligned}\mathcal{R}_i(y_0^{l-1})=&\frac{\sum_{c\in\mathcal{C}_i^{(0)}}\mathbf{W}(y_0^{l-1}|c)}{\sum_{c'\in\mathcal{C}_i^{(1)}}\mathbf{W}(y_0^{l-1}|c')}=\frac{\sum_{c\in\mathcal{C}_i^{(0)}}\mathbf{W}(c|y_0^{l-1})}{\sum_{c'\in\mathcal{C}_i^{(1)}}\mathbf{W}(c'|y_0^{l-1})}\\=&\frac{\mathbf{W}_1^{(i)}(\mathbf{0},0|y_0^{l-1})}{\mathbf{W}_1^{(i)}(\mathbf{0},1|y_0^{l-1})}\end{aligned}$$

is the likelihood ratio for the i-th symbol, and $C_i^{(0)}=C_{i+1}$, $C_i^{(1)}=C_i\setminus C_{i+1}$. Instead of computing the integral in (9) over $y_0^{l-1}\in\mathbb{R}^l$, one can compute

$$I_1^{(i)}=1-\int_{-\infty}^{\infty}p_i(\xi|0)\log_2(1+e^{-\xi})d\xi,\tag{10}$$

where $p_i(\xi|0)$ is the probability density function of the log-likelihood ratio $\xi=\log\mathcal{R}_i(Y_0^{l-1})$, where Y_j follow the distribution $\mathbf{W}(y|0)$. Such integral can be computed by the Monte-Carlo method, i.e. by supplying the random vectors Y_0^{l-1} to a decoder capable of computing $\log\mathcal{R}_i(Y_0^{l-1})$, and averaging the corresponding values.

However, exact computation of $\log\mathcal{R}_i(Y_0^{l-1})$ may be too complex for a practical implementation. It turns out that substituting these values with approximate LLRs given by (5) leads to invalid results (e.g. $I_1^{(i)}<0$). Therefore, we propose to rewrite (9) as

$$\begin{aligned}I_1^{(i)}=&1-\int_{\mathbb{R}^l}\mathbf{W}(y_0^{l-1}|0)\log_2\left(1+\frac{\mathbf{W}_1^{(i)}(\mathbf{0},1|y_0^{l-1})}{\mathbf{W}_1^{(i)}(\mathbf{0},0|y_0^{l-1})}\right)dy_0^{l-1}\\\approx&1-\int_{-\infty}^{\infty}f_i(\psi|0)\log_2\left(1+\frac{P\left\{u_i=1|\psi\right\}}{P\left\{u_i=0|\psi\right\}}\right)d\psi\\=&1-\int_{-\infty}^{\infty}f_i(\psi|0)\log_2\left(1+\frac{f_i(-\psi|0)}{f_i(\psi|0)}\right)d\psi,\end{aligned}\tag{11}$$

where the last equality is valid for a symmetric channel, $f_i(\psi|0)$ is the probability density function of $\mathcal{L}_1^{(i)}(\mathbf{0}|y_0^{l-1})$, and $P\left\{u_i=c|\psi\right\}$ is the probability of $u_i=c$ under the condition of $\mathcal{L}_1^{(i)}(\mathbf{0}|y_0^{l-1})=\psi$. Hence, one can estimate $I_1^{(i)}$ by construction of a histogram for $f_i(\psi|0)$ from the output of an algorithm capable of computing (5).

B. Gaussian approximation

To construct a polar code with some kernel F_l, we propose to assume that all subchannels $\mathbf{W}_\lambda^{(i)},0\leq\lambda\leq m,0\leq i<l^\lambda$, are Gaussian ones, so that they can be completely characterized by the corresponding mutual information $I_\lambda^{(i)}$. We propose to construct tables $I_1^{(i)}(C)$ of values of $I_1^{(i)}$ for some finite set of parameters of the underlying AWGN channel $\mathbf{W}$, where C is the capacity of $\mathbf{W}$. These tables can be used to interpolate the value of $I_1^{(i)}(C)$ for any $C\in[0,1]$.

To construct an (l^m,k) polar code for AWGN channel with capacity C, we propose to recursively compute the capacities of bit subchannels

$$I_m^{(li+j)}(C)\approx I_1^{(j)}(I_{m-1}^{(i)}(C)),0\leq j<l,m>1,\tag{12}$$

where $I_0^{(0)}(C)=C$, and declare frozen those symbols u_i, where i are the indices of subchannels with the smallest $I_m^{(i)}(C)$. Observe that this approach can be also used with approximate kernel processing algorithms [17].

IV. Polar subcodes

A. Low-weight codewords of a polar code

The following theorem provides a simple characterization of the set of unfrozen symbol indices, which are responsible for introducing LWNZ codewords into a polar code.

Theorem 1 ([23]). *Consider an (n,k) polar code $\mathbf{C}$ given by a polarizing transformation $A=BF_0\otimes\cdots\otimes F_{m-1}$ and the set of frozen symbol indices $\mathcal{F}$, where F_i is an l_i-dimensional kernel, B is the digit-reversal permutation matrix, $n=\prod_{i=0}^{m-1}l_i$, and the partial distances D_{ij} of each kernel F_i satisfy*

$$D_{i,j}=\mathrm{wt}(F_{i,j}),0\leq j<l_i,0\leq i<m,\tag{13}$$

where $F_{i,j}$ is the j-th row of F_i. Then:

1) *The minimum distance of the polar code is $d = \min_{i \notin \mathcal{F}} \mathrm{wt}(A_i)$, where A_i is the i-th row of matrix A.*
2) *Any codeword $c_0^{n-1} = u_0^{n-1}A$ of weight d, where d is the minimum distance, has $u_i = 1$ for at least one $i : \mathrm{wt}(A_i) = d$.*

Proof. It is sufficient to consider the case of $D_{i,j} \leq D_{i,j+1}, 0 \leq j < l_i - 1$, since otherwise one can swap the corresponding rows of F_i [5, Proposition 15], and modify appropriately the set $\mathcal{F}$, so that the code remains the same. In this case $D_{i,j}$ is the minimum distance of the code generated by rows $j, \ldots, l_i - 1$ of F_i.

Both statements for $m = 1$ follow from (13). Assume that the theorem holds for some $A = B \bigotimes_{i=0}^{m-1} F_i$ with $m \geq 1$, and consider the polarizing transformation $A' = B' \bigotimes_{i=0}^{m} F_i$. Then $\mathrm{wt}(A'_{il_m+j}) = \mathrm{wt}(A_i)\,\mathrm{wt}((F_{l_m})_j)$, where $j \in [l_m], i \in [n]$. Consider a polar code with polarizing transformation A' and the set of frozen symbol indices $\mathcal{F}$. It can be considered as a generalized concatenated code with outer (l_m, k_i, d_i) codes $\mathcal{C}^{(i)}$ and inner (n, K_i, D_i) codes $\mathbb{C}^{(i)}$, where $\mathcal{C}^{(i)}$ is generated by $F_{l_m,j} : il_m + j \notin \mathcal{F}$, and $\mathbb{C}^{(i)}$ is a polar code with polarizing transformation A and the set of frozen symbol indices $\mathcal{F}^{(i)} = \{j | l_m j + s \in \mathcal{F}, i \leq j < n, 0 \leq s < l_m\}, 0 \leq i < n$. Then the minimum distance of the GCC is $d \geq d_i D_i$, where $D_i = \min_{j \notin \mathcal{F}^{(i)}} \mathrm{wt}(A_j)$, and $d_i \geq \min_{j : l_m i + j \notin \mathcal{F}} D_{m,j}$. This bound is achieved with equality, since there is a weight-d codeword in the considered code given by a row $il_m + j \notin \mathcal{F}$ of A'.

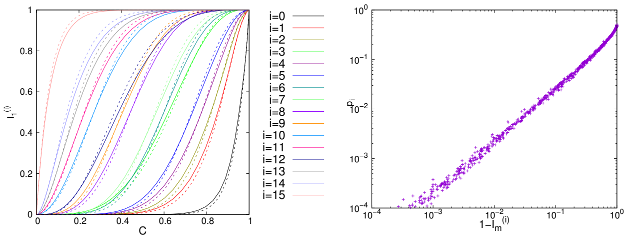
The second statement also holds for any m, since if a codeword of $\mathbf{C}$ has $u_i = 0$ for all $i : \mathrm{wt}(A_i) = d$, then it belongs to a code with the set of frozen symbol indices $\mathcal{F}' = \mathcal{F} \setminus \{i | \mathrm{wt}(A_i) = d\}$, which has minimum distance $d' > d$, according to the first statement. □

B. Polar subcodes

To obtain an (n, k) code with good performance under SCL decoding, we need to eliminate LWNZ codewords from a polar code. This can be done by introducing dynamic freezing constraints (6). To reduce the probability of the correct path being killed by the SCL decoder at an early phase, these constraints need to be imposed on symbols u_i with the smallest possible indices i in such way, so that low-weight codewords are still eliminated. Theorem 1 suggests that one can do this by extending the construction of [4], i.e. by selecting i as t maximal indices, such that $\mathrm{wt}(A_i) = d$, where d is the minimum distance of $(n, k+t, d)$ parent polar code, and obtain $V_{s_i,j}$ as independent random binary values.

However, more careful design is possible. Since the minimum distance of polar codes, even with large kernels, is very low, one can explicitly enumerate (almost) all non-zero codewords $c^{(p)}, 0 \leq p < P$, of the parent code with weight up to $\delta \geq d$ as described in [25]. Let $u^{(p)} = c^{(p)} A^{-1}$ be the corresponding information vectors, and let $\mathcal{P} = [P]$. We say that codeword $c^{(p)}$ is pruned at phase i if

$$u_i^{(p)} \neq \sum_{j<i} V_{s_i,j} u_j. \tag{14}$$



(a) Subchannel capacity functions (b) Accuracy for $m = 3$

Fig. 1: Gaussian approximation for kernel K_2

Let $\mathcal{P}_i$ be the set of codewords pruned at phase $i \in [n]$. Then for some i one can select the coefficients $V_{s_i,j}$ so that $|\mathcal{P}_i|$ is maximized[2]. Having obtained $\mathcal{P}_i$, we set $\mathcal{P} := \mathcal{P} \setminus \mathcal{P}_i$, and proceed with construction of the dynamic freezing constraint for the next suitable index $i' < i$.

The particular phases i to be considered in the above described elimination process are selected as follows. Let $w_0 < w_1 < \ldots$ be the weights of rows of A, which correspond to non-frozen symbols in the parent code. Let $i^{(j)}$, where $\mathrm{wt}(A_{i^{(j)}}) = w_j$, be the maximal index of a symbol, which is not yet frozen. If no such symbol exists, we assume $i^{(j)} = -\infty$. Let us set initially $\rho = 0$. We propose to set i as $i^{(\rho)}$, provided that $i^{(\rho)} \in [n]$, and the number of codewords, which can be pruned for this i as described above, is at least $(\frac{1}{2} - \delta)|\mathcal{P}|$, where δ is a small value. Otherwise, we set $\rho := \rho + 1$ and repeat this selection process until t dynamic freezing constraints are constructed. The parameter t should be selected as the smallest integer, which results in the number of LWNZ codewords in the obtained code to achieve some target value (e.g. 0).

Further performance gain can be obtained by employing protocodes with type-B dynamic frozen symbols, i.e. by imposing constraints (6) on q most reliable frozen symbols. The coefficients in these constraints can be selected as independent equiprobable binary values. Here t and q are the parameters of the proposed construction. Their optimal values depend on the target SCL decoder list size.

V. Numeric results

In this section we present simulation results for the case of 16×16 kernel K_2 with scaling exponent 3.45 introduced in [9]. Figure 1a presents the bit subchannel capacity functions for AWGN cnannel (solid lines) and BEC (dashed lines). It can be seen that there are some crossover points between the curves for different subchannels. Depending on the type of the underlying channel, these points are located at different values of its capacity C. Hence, the capacity functions for BEC cannot be reliably used to obtain codes for AWGN channel.

Figure 1b illustrates the simulated bit error rate P_i for each subchannel $W_3^{(i)}$ vs estimated capacity $I_3^{(i)}$ for a genie-aided SC decoder and AWGN channel at $E_s/N_0 = -1$ dB. It can be

[2] This can be implemented by generating randomly a few sets of coefficients $V_{s_i,j}$, checking (14) for each codeword in $\mathcal{P}$, and selecting the set which prunes the maximal number of codewords.

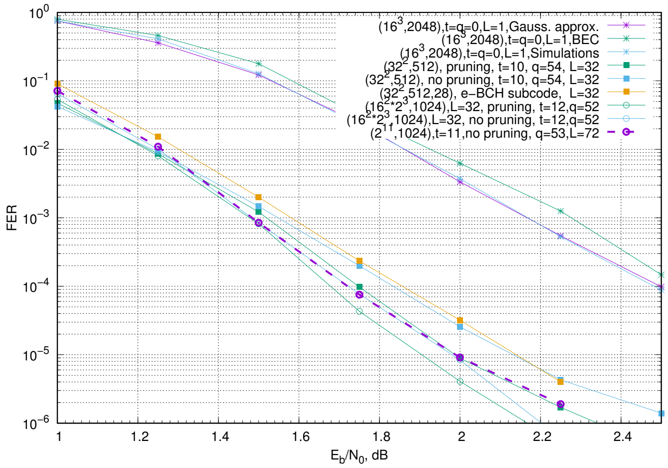


Fig. 2: Performance of Polar Subcodes

seen that P_i almost monotonically increases with $1-I_3^{(i)}$. This means that the proposed approach indeed enables accurate subchannel reliability estimation.

Figure 2 illustrates the performance of various polar (sub)codes for the case of SCL decoding. Here $(l_0^{m_0}l_1^{m_1}, k)$ denotes a k-dimensional code based on the polarizing transformation $A = BF_{l_0}^{\otimes m_0} \otimes F_{l_1}^{\otimes m_1}$, and F_{l_i} is a $l_i \times l_i$ kernel. For $L = 1$ (i.e. SC decoding) polar codes constructed using the proposed method for $E_b/N_0 = 2$ dB provide almost the same performance as those obtained by Monte-Carlo simulations. However, the code obtained for BEC with the same capacity value (0.64251) exhibits a noticeable performance loss. For SCL decoding, the proposed pruning method enables one to obtain up to 0.2 dB gain compared to a randomized construction without pruning. Observe that the obtained $(1024, 512)$ code outperforms the polar subcode of an extended BCH code [3], where low-weight codewords are eliminated algebraically. It can be also seen that the pruned mixed-kernel $(2048, 1024)$ polar subcode outperforms the one with Arikan kernel only, while the SCL decoding complexity of these codes is 10^6 and $2 \cdot 10^6$ operations, respectively. The gain of the pruning operation in the case of codes with Arikan kernel only was found to be negligible.

VI. Conclusions

In this paper construction techniques for polar subcodes with large kernels were introduced. The proposed approach combines Gaussian approximation and explicit elimination of low-weight codewords. It enables one to explicitly control the minimum distance and the number of low-weight codewords in the obtained codes.

The obtained codes outperform polar subcodes of extended BCH codes and the randomized construction without pruning. With appropriate parameter selection, the obtained codes can provide better performance with lower SCL decoding complexity compared to polar subcodes with Arikan kernel only.

References

[1] E. Arikan, “Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels,” *IEEE Transactions on Information Theory*, vol. 55, no. 7, July 2009.

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

[3] P. Trifonov and V. Miloslavskaya, “Polar subcodes,” *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 2, February 2016.

[4] P. Trifonov and G. Trofimiuk, “A randomized construction of polar subcodes,” in *Proceedings of IEEE International Symposium on Information Theory*. Aachen, Germany: IEEE, 2017, pp. 1863–1867.

[5] 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.

[6] R. Mori and T. Tanaka, “Channel polarization on q-ary discrete memoryless channels by arbitrary kernels,” in *Proceedings of IEEE International Symposium on Information Theory*, 2010.

[7] H. Pfister and R. Urbanke, “Near-optimal finite-length scaling for polar codes over large alphabets,” in *Proceedings of IEEE International Symposium on Information Theory*, 2016.

[8] 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.

[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] R. Mori and T. Tanaka, “Performance of polar codes with the construction using density evolution,” *IEEE Communications Letters*, vol. 13, no. 7, July 2009.

[11] I. Tal and A. Vardy, “How to construct polar codes,” *IEEE Transactions on Information Theory*, vol. 59, no. 10, pp. 6562–6582, October 2013.

[12] P. Trifonov, “Efficient design and decoding of polar codes,” *IEEE Transactions on Communications*, vol. 60, no. 11, November 2012.

[13] H. Vangala, E. Viterbo, and Y. Hong, “A comparative study of polar code constructions for the AWGN channel,” *CoRR*, vol. abs/1501.02473, Jan. 2015. [Online]. Available: https://arxiv.org/abs/1501.02473

[14] V. Miloslavskaya and P. Trifonov, “Design of polar codes with arbitrary kernels,” in *Proceedings of IEEE Information Theory Workshop*, 2012, pp. 119–123.

[15] N. Presman, O. Shapira, and S. Litsyn, “Mixed-kernels constructions of polar codes,” *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 2, pp. 239–253, Feb 2016.

[16] H. Griesser and V. R. Sidorenko, “A posteriory probability decoding of nonsystematically encoded block codes,” *Problems of Information Transmission*, vol. 38, no. 3, 2002.

[17] 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.

[18] P. Trifonov, “Algebraic matching techniques for fast decoding of polar codes with Reed-Solomon kernel,” in *Proceedings of IEEE International Symposium on Information Theory*, Vail, USA, 2018.

[19] I. Sason and S. Shamai, “Performance analysis of linear codes under maximum-likelihood decoding: A tutorial,” *Foundations and Trends in Communications and Information Theory*, vol. 3, no. 1–2, 2006.

[20] S. J. MacMullan and O. M. Collins, “The capacity of binary channels that use linear codes and decoders,” *IEEE Transactions On Information Theory*, vol. 44, no. 1, January 1998.

[21] J. T. Coffey and A. B. Kiely, “The capacity of coded systems,” *IEEE Transactions On Information Theory*, vol. 43, no. 1, January 1997.

[22] S. ten Brink, “Convergence behavior of iteratively decoded parallel concatenated codes,” *IEEE Transactions On Communications*, vol. 49, no. 10, October 2001.

[23] P. Trifonov, “Design of randomized polar subcodes with non-Arikan kernels,” in *Proceedings of 16-th International Workshop on Algebraic and Combinatorial Coding Theory*, 2018.

[24] M. Bardet, V. Dragoi, A. Otmani, and J.-P. Tillich, “Algebraic properties of polar codes from a new polynomial formalism,” in *Proceedings of IEEE International Symposium on Information Theory*, 2016.

[25] A. Canteaut and F. Chabaud, “A new algorithm for finding minimum-weight words in a linear code: Application to McEliece’s cryptosystem and to narrow-sense BCH codes of length 511,” *IEEE Transactions on Information Theory*, vol. 44, no. 1, pp. 367–378, January 1998.