Randomized Chained Polar Subcodes

Peter Trifonov
Saint Petersburg Polytechnic University, Russia
Email: petert@dcn.icc.spbstu.ru

***Abstract*—A generalization of polar subcodes is proposed, which enables a simple construction of codes of arbitrary length. The obtained codes are shown to outperform punctured and shortened polar codes under list/sequential decoding. Furthermore, a simplified Gaussian approximation for polar codes is presented.**

I. Introduction

Polar codes is a novel class of error-correcting codes, which asymptotically achieve the symmetric capacity of memoryless channels, have low complexity construction, encoding and decoding algorithms [1]. However, the performance of polar codes of practical length is quite poor. Generalizations of Arikan polar codes, such as polar codes with CRC [2] and polar subcodes [3], [4] were shown to provide substantially better performance under list, stack and sequential decoding [2], [5], [6]. Another problem with polar codes is that their length is limited to 2^m. Several puncturing, shortening and extension techniques were suggested for polar codes [7], [8], [9], [10]. However, the performance of the obtained codes is not quite good. Furthermore, some of these constructions require employing two-dimensional Gaussian approximation, which is highly challenging numerically, or computationally expensive density evolution [11].

In this paper we propose an alternative method for construction of polar codes of arbitrary length, which is based on the chaining construction [12]. Essentially, the idea is to combine several polarizing transformations of different size, and make their input values linearly dependent. This enables obtaining codes of arbitrary length without shortening or puncturing. Furthermore, randomized dynamic freezing constraints are used to improve the weight distribution of the obtained code, similarly to the construction introduced in [4], and obtain performance better than in the case of BCH-based polar subcodes. We propose also a simplified implementation of the Gaussian approximation, which avoids evaluation of transcendent functions.

II. Background

A. Channel polarization

A $(n=2^m,k)$ polar code over $\mathbb{F}_2$ is a set of vectors $c_0^{n-1}=u_0^{n-1}A_m$, where $a_i^j=(a_i,\ldots,a_j)$, $A_m=\begin{pmatrix}1&0\\1&1\end{pmatrix}^{\otimes m}$ is a matrix of the polarizing transformation, $F^{\otimes m}$ denotes m-times Kronecker product of matrix F with itself, $u_i=0, i\in\mathcal{F}$, and $\mathcal{F}\subset\{0,\ldots,n-1\}$ is a set of $n-k$ frozen symbol indices.

Let $W(y|x), y\in\mathcal{Y}, x\in\mathbb{F}_2$ be a transition probability function of a binary input memoryless symmetric channel. Then the Arikan channel transformations are defined by

$$(W'\boxplus W'')(y_0,y_1|u_0)=\frac{1}{2}\sum_{u_1\in\mathcal{X}}W'(y_0|u_0\oplus u_1)W''(y_1|u_0)$$

$$(W'\circledast W'')(y_0,y_1,u_0|u_1)=\frac{1}{2}W'(y_0|u_0\oplus u_1)W''(y_1|u_0)$$

By recursive application of these transformations, one obtains synthetic bit subchannes

$$\mathbf{W}_{2i}^{(m)}=\mathbf{W}_i^{(m-1)}\boxplus\mathbf{W}_i^{(m-1)} \quad (1)$$

$$\mathbf{W}_{2i+1}^{(m)}=\mathbf{W}_i^{(m-1)}\circledast\mathbf{W}_i^{(m-1)}, \quad (2)$$

where $\mathbf{W}_0^{(0)}=W$ is the original communication channel. It is possible to show that the capacities of the obtained subchannels converge to 0 or 1, and the fraction of subchannels with capacity close to 1 converges to the capacity of the original channel W.

In the case of W being the binary erasure channel (BEC) with erasure probability $Z_{0,0}$, the obtained subchannels are also BEC with Bhattacharyya parameters given by

$$Z_{m,2i}=2Z_{m-1,i}-Z_{m-1,i}^2 \quad (3)$$

$$Z_{m,2i}=Z_{m-1,i}^2, \quad (4)$$

so that their capacities can be computed as $I_{m,i}=1-Z_{m,i}$. For other types of channels W, computing the capacities of the synthetic bit subchannels requires employing computationally expensive density evolution [13], [11].

Decoding of polar codes can be implemented using the successive cancellation (SC) algorithm, which makes decisions

$$\hat{u}_i=\begin{cases}\operatorname{sgn}L_i^{(m)}(y_0^{n-1},\hat{u}_0^{i-1})>0, & i\notin\mathcal{F}\\ \text{the frozen value of } u_i, & i\in\mathcal{F},\end{cases} \quad (5)$$

where the log-likelihood ratios (LLRs)

$$L_i^{(m)}(y_0^{N-1},\widehat{u}_0^{i-1})=L_i^{(m)}=\ln\frac{\mathbf{W}_i^{(m)}(y_0^{n-1},\widehat{u}_0^{i-1}|u_i=0)}{\mathbf{W}_i^{(m)}(y_0^{n-1},\widehat{u}_0^{i-1}|u_i=1)}$$

are given by

$$L_{2i}^{(l+1)}=2\tanh^{-1}\left(\tanh(L_i^{(l)}/2)\cdot\tanh(L_{i+\eta}^{(l)}/2)\right), \quad (6)$$

$$L_{2i+1}^{(l+1)}=(-1)^{v_{l,i}}L_i^{(l)}+L_{i+\eta}^{(l)}, \quad (7)$$

$\eta=2^{m-l-1}$, $(v_{l,j},v_{l,j+\eta})=(v_{l+1,j},v_{l+1,j+\eta})A_1$, $v_{m,j}=\hat{u}_j$, and $L_i^{(0)}$ are the LLRs for the input symbols.

However, since the estimates $\hat{u}_i$ are constructed without taking into account freezing constraints on symbols $u_j, j > i$, this algorithm does not provide ML decoding.

B. Polar subcodes

It is possible to show that the minimum distance of classical Arikan polar codes is $O(\sqrt{n})$, which is too low for practical applications [14]. It was suggested in [3] to set frozen symbols $u_i, i \in \mathcal{F}$, not to zero, but to some linear combinations of $u_j, j < i$, i.e.

$$u_i = \sum_{j<i} V_{s_i,j} u_j, i \in \mathcal{F}, \tag{8}$$

where V is a $(n-k) \times n$ constraint matrix, such that the last non-zero elements of each row are located in distinct columns, and s_i is the index of the row ending in column i. Symbols u_i with at least one non-zero coefficient in the r.h.s. of (8) are referred to as dynamic frozen. The obtained codes are referred to as polar subcodes. Techniques for construction of matrix V, which result in codes with improved minimum distance or reduced error coefficient, are presented [3], [4]. Decoding of polar subcodes can be performed using the above described successive cancellation algorithm and its extensions [2], [6].

C. Gaussian approximation

Finding the set of frozen symbols both for the case of classical polar codes and polar subcodes requires one to evaluate the reliability of bit subchannels. Let us assume that a zero codeword is transmitted. It was suggested in [15] to approximate the distributions of $L_i^{(\lambda)}, 0 \le i < 2^\lambda$, by Gaussian ones, so that the *reliability* of each subchannel can be characterized by the expected values

$$\mathcal{L}_i^{(m)} = \mathbf{E}_{y_0^{N-1}} \left[L_i^{(m)}(y_0^{N-1}, \mathbf{0}) \right]$$

of the LLRs. These values can be computed as

$$\mathcal{L}_{2i}^{(\lambda+1)} = \phi^{-1} \left(1 - (1 - \phi(\mathcal{L}_i^{(\lambda)}))(1 - \phi(\mathcal{L}_{i+\eta}^{(\lambda)})) \right) \tag{9}$$

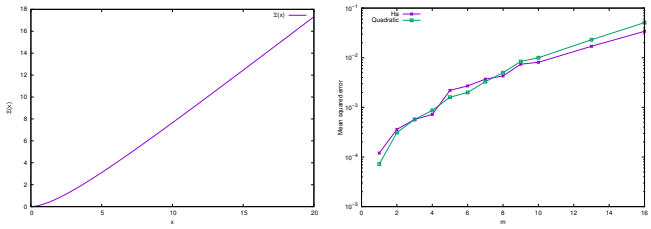
$$\mathcal{L}_{2i+1}^{(\lambda+1)} = \mathcal{L}_i^{(\lambda)} + \mathcal{L}_{i+\eta}^{(\lambda)}, \tag{10}$$

where

$$\phi(x) = \begin{cases} 1 - \frac{1}{\sqrt{4\pi x}} \int_{-\infty}^{\infty} \tanh \frac{z}{2} e^{-\frac{(z-x)^2}{4x}} dz, & x > 0, \\ 1, & x = 0. \end{cases} \tag{11}$$

The initial values for this recursion are given by $\mathcal{L}_i^{(0)} = 2/\sigma^2$ for normal codeword symbols c_i, $\mathcal{L}_i^{(0)} = 0$ for punctured[1] symbols, and $\mathcal{L}_i^{(0)} = \infty$ for shortened symbols, where σ^2 is the target noise variance. This approach is referred to as one-dimensional Gaussian approximation (GA) if $\forall \lambda, i : \mathcal{L}_i^{(\lambda)} = \mathcal{L}_{i+\eta}^{(\lambda)}$, and two-dimensional GA otherwise. The bit error rate in $\mathbf{W}_i^{(m)}$ can be estimated as $P_{m,i} \approx Q\left(\sqrt{\mathcal{L}_i^{(m)}/2} \right)$.

[1]Puncturing means that codeword symbol c_i, which may take arbitrary values, is omitted at the transmitter side. Shortening means that the encoder ensures that $c_i = 0$, and this symbol is omitted at the transmitter side.



(a) Function $\Xi(x)$ (b) Accuracy

Fig. 1: Gaussian approximation

Function $\phi(x)$ is rather challenging to implement. An approximation for it was suggested in [16], but it still involves transcendent functions, which may take arbitrary small and large values, and are therefore very difficult to implement in hardware using fixed-point arithmetic.

III. Simplified Gaussian approximation

Let us consider design of a polar code of length 2^m based on Gaussian approximation. The main computation burden of this approach consists in calculation of the non-linear function

$$\Xi(x) = \phi^{-1} \left(1 - (1 - \phi(x))^2 \right),$$

which employs numerically challenging function $\phi(x)$. However, it turns out that $\Xi(x)$ is very well-behaved, as shown in Figure 1a. We propose to approximate the latter function with a piecewise quadratic one

$$\Xi(x) \approx \begin{cases} 0.9861x - 2.3152, & x > 12 \\ x(9.005 \cdot 10^{-3} x + 0.7694) - 0.9507, & x \in (3.5, 12] \\ x(0.062883x + 0.3678) - 0.1627, & x \in (1, 3.5] \\ x(0.2202x + 0.06448), & \text{otherwise.} \end{cases} \tag{12}$$

This approximation was obtained by minimum squared error curve-fitting. It can be seen that the complexity of polar code design using this approximation is essentially the same as in the case of the BEC recursion (3)–(4), since both of them are based on evaluation of quadratic polynomials. This operation involves only summations and multiplications, and avoids any transcendent functions. Figure 1b shows the mean squared error of $\log P_{m,i}$ for $E_s/N_0 = 0$ dB for various values of m. It can be seen that the proposed approach provides almost the same accuracy as Ha et al approximation [16].

IV. Chained polar subcodes

A. Motivation

Most of the existing techniques for construction of polar codes of arbitrary length are based on puncturing and shortening [7], [8], [9], [10]. In general, puncturing and shortening does affect the reliability of bit subchannels, and it must be properly taken into account in (9)–(10) while constructing $\mathcal{F}$. In this case $\mathcal{L}_i^{(\lambda)}$ and $\mathcal{L}_{i+\eta}^{(\lambda)}$ may be different, so that one cannot use the simple approximation presented in Section III.

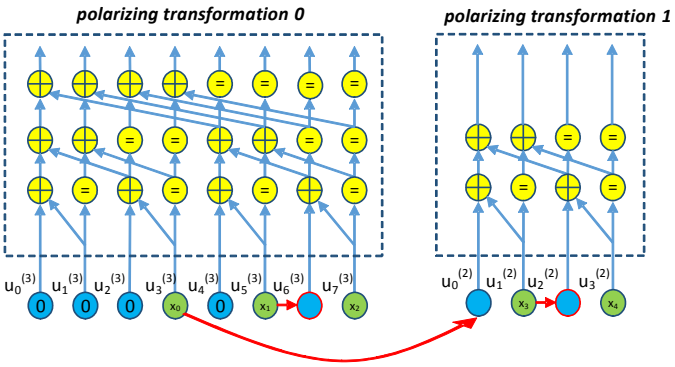


Fig. 2: Encoding diagram for $(12, 5)$ chained code

Therefore, we propose to design codes, which can be decoded by a simple generalization of the successive cancellation algorithm or its derivatives, but do not require shortening or puncturing, and can be constructed using the simple one-dimensional Gaussian approximation method presented in Section III.

B. Code construction

We propose to construct a code of length $n = \sum_{j=0}^{\zeta-1} 2^{m_j}$ and dimension k, such that its codewords represent a concatenation of the output vectors of polarizing transformations A_{m_i}, the input vectors of these polarizing transformations have some linear dependencies, and $m_j > m_{j+1}, 0 \le i < \zeta - 2$. Namely, the codeword is obtained as

$$c = \underbrace{(u^{(m_0)}, \dots, u^{(m_{\zeta-1})})}_{\mathbf{u}} \underbrace{\mathrm{diag}(A_{m_0}, \dots, A_{m_{\zeta-1}})}_{A},$$

where $\mathbf{u}\mathbb{V}^T = 0$, and $\mathbb{V}$ is a constraint matrix with rows ending in distinct columns. Such code can be still treated in the framework of dynamic frozen symbols introduces in Section II-A. Figure 2 illustrates an encoder for the proposed code construction.

It is possible to implement decoding of chained polar subcodes by applying the above described SC decoding algorithm to the parts of the received sequence corresponding to the output of different polarizing transformations A_{m_i}.

Obviously, in this case by setting $m_{\zeta-1} \to \infty$ and defining $\mathbb{V}$ as a matrix having weight-1 rows, having 1's in distinct columns corresponding to the least reliable bit subchannels $\mathbf{W}_j^{(m_i)}$, one can obtain capacity-achieving codes. However, we are interested in obtaining finite-length codes, which provide good performance under the (list) SC algorithm.

Chained polar subcodes were originally introduced in [12], where the classical X4 construction based on BCH codes was used to obtain codes with a prescribed minimum distance. In this paper we present an alternative approach, which results in codes with improved performance under list/sequential SC decoding with small list size.

C. Merging bit subchannels

1) Extending a polar code by one symbol: Consider construction of a chained code of length $n = 2^{m_0} + 1$, i.e. let $m_1 = 0, \zeta = 2$. Assume that all symbols corresponding to bit subchannels $\mathbf{W}_i^{(m_j)}$, with reliability $\mathcal{L}_i^{(m_j)}$ less than some threshold T are frozen, $0 \le j < 2, 0 \le i < 2^{m_j}$. Let k be the dimension of the obtained code. However, if $u_0^{(m_1)}$ is frozen, it may be possible to obtain a code of dimension $k+1$. Indeed, in this case there exists at least one bit subchannel $\mathbf{W}_i^{(m_0)}$ with reliability less than the reliability of $\mathbf{W}_0^{(m_1)}$, so that $u_i^{(m_0)}$ is also frozen. Let

$$i_0 = \arg \max_{i:\mathcal{L}_i^{(m_0)} < T} \mathcal{L}_i^{(m_0)}$$

be the index of the most reliable bit subchannel corresponding to a frozen $u_{i_0}^{(m_0)}$. We propose to define merged subchannels

$$\mathbf{W}_{i_0,0}^{(m_0,m_1)} = \mathbf{W}_{i_0}^{(m_0)} \boxplus \mathbf{W}_0^{(m_1)} \tag{13}$$

$$\mathbf{W}_{i_0,1}^{(m_0,m_1)} = \mathbf{W}_{i_0}^{(m_0)} \circledast \mathbf{W}_0^{(m_1)} \tag{14}$$

If the reliability $\mathcal{L}_{i_0}^{(m_0)} + \mathcal{L}_0^{(m_1)}$ of $\mathbf{W}_{i_0,1}^{(m_0,m_1)}$ is better than T, then one can transmit one more symbol over this bit subchannel. This corresponds to $\mathbb{V}$ being a $(2^{m_0} - k) \times n$ matrix having $2^{m_0} - 1 - k$ weight-1 rows, and one weight-2 row, containing 1's in positions i_0 and 2^{m_0}.

D. General case

The above approach can be recursively extended to the case of arbitrary length n as follows.

- Let the codes of length 2^μ be obtained as classical polar codes.
- Let the codes of length $2^\mu n$, where $n > 1$ is odd, be obtained as generalized concatenated codes with inner Arikan polar codes of length 2^μ, and outer codes of length n, recursively constructed with this method.
- Let the codes of length $2^\mu n + 1$, where n is odd, be obtained by applying the construction described in Section IV-C1 to a code of length $2^\mu n$, recursively constructed with this method.

Let us describe the code construction method in more details.

We propose to construct constraint matrix $\mathbb{V}$ so that the data symbols, which are mapped onto insufficiently reliable bit subchannels corresponding to some polarizing transformation A_{m_j}, are repeated over a subchannel in another polarizing transformation A_{m_p}. Such operation will be referred to as *symbol boosting*. Boosting enables one to improve the reliability of the corresponding symbols, reducing thus the probability of error under successive cancellation decoding. Let $u = (u^{(m_0)}, \dots, u^{(m_{s-1})})$ be the input vector of the combined polarizing transformation given by matrix A. We call symbols $u_i^{(m_j)}$ and $u_q^{(m_p)}$, $m_j > m_p$, adjacent iff $\lfloor i 2^{m_p - m_j} \rfloor = q$. In order to simplify the design of the code and the decoder, we propose to apply boosting only to adjacent symbols. Essentially, this results in a generalized concatenated code (see [17], [15] for definition and details) with inner Arikan polar

ALLOCATE$(n, k, \mathcal{L}_0^{(0)})$

1 Let $n = \sum_{i=0}^{\zeta-1} 2^{m_i}, m_i > m_{i+1}$
2 $\mathcal{L}_{2j}^{(m_i)} \leftarrow \Xi(\mathcal{L}_j^{(m_i-1)}), \mathcal{L}_{2j+1}^{(m_i)} \leftarrow 2\mathcal{L}_j^{(m_i-1)}, 0 \le j < 2^{m_i-1}$
3 $T_0 = 0; k_0 = n; T_1 = (n+1)\mathcal{L}_0^{(0)}; k_1 = 0$
4 **while** $k_0 > k_1$
5 **do for** $j = 0, \ldots, \zeta-1, i = 0, \ldots, 2^{m_j}-1$
6 **do** $T \leftarrow (T_0 + T_1)/2$
7 **if** $\mathcal{L}_i^{(m_j)} \le T$
8 **then** $u_i^{(m_j)} \leftarrow \mathbf{F}$
9 **if** $u_i^{(m_j)} = \mathbf{F}$
10 **then for** $p \leftarrow 0$ **to** j
11 **do** $i_p = \arg \max_{\substack{v: u_v^{(m_p)} = \mathbf{F} \\ \lfloor v 2^{m_j - m_p} \rfloor = i}} \mathcal{L}_v^{(m_p)}$
12 $t_0 \leftarrow \max_{\substack{0 \le t < j \\ \sum_{q=t}^{j} \mathcal{L}_{i_q}^{(m_q)} > T}} t$
13 **if** t_0 exists
14 **then** $u_{i_{t_0}}^{(m_{t_0})} \leftarrow \mathbf{U}$
15 $u_{i_p}^{(m_p)} \leftarrow \mathbf{A}, t_0 < p \le j$
16 $K \leftarrow \left| \left\{ (j,i) | u_i^{(m_j)} = \mathbf{U}, 0 \le i < 2^{m_j}, 0 \le j < \zeta \right\} \right|$
17 **if** $K < k$
18 **then** $T_1 \leftarrow T; k_1 \leftarrow K$
19 **else** $T_0 \leftarrow T; k_0 \leftarrow K$
20 $u_i^{(m_j)} \leftarrow \mathbf{F}$ for all $i, j : u_i^{(m_j)} = \mathbf{A}$

Fig. 3: Initial symbol allocation

BOOST$(u_j^{(m_i)})$

1 $R \leftarrow 0; \mathbb{V} \leftarrow 0_{(n-k)\times n}$
2 **for** $j \leftarrow 0$ **to** $\zeta - 1$
3 **do for** $i \leftarrow 0$ **to** $2^{m_j} - 1$
4 **do** $w_{z_{m_j,i}} \leftarrow 2^{\mathrm{wt}(i)}$
5 **if** $u_i^{(m_j)} = \mathbf{F}$
6 **then** $\mathbb{V}_{R, z_{m_j,i}} \leftarrow 1; Pz_{m_j,i} \leftarrow R$
7 **for** $p \leftarrow 0$ **to** $j-1$
8 **do** $i_p = \arg \min_{\substack{\lfloor i' 2^{m_j - m_p} \rfloor = i \\ u_{i'}^{(m_p)} = \mathbf{U}}} \mathcal{L}_{i'}^{(m_p)}$
9 **if** i_p exists
10 **then** $\mathbb{V}_{R, z_{m_p,i_p}} \leftarrow 1$
11 $\mathcal{L}_{i_p}^{(m_p)} \leftarrow \mathcal{L}_{i_p}^{(m_p)} + \mathcal{L}_i^{(m_j)}$
12 $R \leftarrow R + 1$
13 **else for** $p \leftarrow 0$ **to** $j-1$
14 **do** $i_p = \arg \min_{\substack{\lfloor i' 2^{m_j - m_p} \rfloor = i \\ u_{i'}^{(m_p)} = \mathbf{F}}} \mathcal{L}_{i'}^{(m_p)}$
15 **if** i_p exists
16 **then** $\mathbb{V}_{P_{z_{m_p,i_p}}, z_{m_j,i}} \leftarrow 1$
17 $P_{z_{m_j,i}} \leftarrow Pz_{m_p,i_p}$
18 $u_i^{(m_j)} \leftarrow \mathbf{F}; u_{i_p}^{(m_p)} \leftarrow \mathbf{U}$
19 $\mathcal{L}_{i_p}^{(m_p)} \leftarrow \mathcal{L}_{i_p}^{(m_p)} + \mathcal{L}_i^{(m_j)}$
20 **return** $\mathbb{V}, w_0^{n-1}, R$

Fig. 4: Symbol boosting

codes of length $2^{m_{\zeta-1}}$, and outer chained polar code obtained recursively using the above described approach.

The proposed construction assumes that all symbols with reliability less than some threshold T are frozen. Setting $T = T_0 = 0$ and $T = T_1 = (n+1)2/\sigma^2$ results in rate-1 and rate-0 codes, respectively. The value of $T \in (T_0, T_1)$ needed to achieve the target code dimension (i.e. the number of unfrozen symbols) k can be determined iteratively using the bisection method or stored in a pre-computed table. Observe that this approach avoids sorting of n values $\mathcal{L}_i^{(m)}$, which may be challenging to implement in hardware for large m.

We propose a two-step method for symbol merging. The first step, called *initial symbol allocation*, is to identify the unfrozen symbols. By abuse of notation, we assign to variables $u_i^{m_j}$ states $\mathbf{F}, \mathbf{U}, \mathbf{A}$, which denote that the corresponding symbol is frozen, unfrozen, or auxiliary, respectively. The corresponding algorithm is shown in Figure 3. The algorithm presented in Figure 4 ensures that the value of each symbol $u_{i_t}^{(m_t)}$, which is declared unfrozen in step 14 above, is essentially transmitted over several subchannels $\mathbf{W}_{i_p}^{(m_p)}, t \le p \le j$, so that its reliability is improved. In order to ensure that these subchannels are not used for transmission of any other symbols, we have to declare them auxiliary.

The second step, called *symbol boosting*, is to construct the specific constraint matrix $\mathbb{V}$. Let $z_{m_j,i} = i + \sum_{p=0}^{j-1} 2^{m_p}$ denote the position of symbol $u_i^{(m_j)}, 0 \le i < 2^{m_j}$, in vector $u = (u^{(m_0)}, \ldots, u^{(m_{s-1})})$. We also use a shorthand notation $u[z_{m_j,i}] = u_i^{(m_j)}$. Figure 4 illustrates the algorithm. It identifies the frozen symbols, which can be used to boost unfrozen ones, and constructs constraints, which ensure that the values of such unfrozen symbols are transmitted over the subchannels corresponding to frozen symbols, i.e. step 10 of the algorithm boosts $u_{i_p}^{(m_p)}$ by $u_i^{(m_j)}, p < j$. Furthermore, unfrozen symbols are identified, which can be boosted by some frozen symbols (step 16).

Observe that step 10 of *Boost* algorithm may cause multiple unfrozen symbols to be boosted by the same frozen symbol, as shown in the below example. Exact evaluation of the reliability $\mathcal{L}_{i_p}^{(m_p)}$ of the obtained combined channels may require employing the two-dimensional Gaussian approximation. In order to keep the complexity of code construction as low as possible, and given rareness of such cases, we use a slightly inaccurate expression in step 10. The values w_j computed by the above algorithm are equal to the weight of the codewords corresponding to $u_j = 1, u_i = 0, i \ne j, j \notin \mathcal{F}$. These values are used in Section IV-E1 for construction of dynamic freezing constraints.

Example 1. *Let us apply the above described method for construction of a* $(n = 7, k = 4)$ *code for AWGN channel*

with $\sigma = 0.93$. *One has* $m_0 = 2, m_1 = 1, m_2 = 0$. *Gaussian approximation can be used to compute* $\mathcal{L}^{(2)} = (0.27, 2.0, 2.75, 9.14), \mathcal{L}^{(1)} = (1.00, 4.57), \mathcal{L}^{(0)} = (2.28)$. *Let* $T = 2.7$.

- *Initial Symbol Allocation:*
  - *Frozen symbols:* $u_0^{(2)}, u_0^{(0)}$
  - *Unfrozen symbols:* $u_1^{(2)}, u_2^{(2)}, u_3^{(2)}, u_1^{(1)}$
  - *Auxiliary symbols:*$u_0^{(1)}$
- *Symbol boosting:*
  - *boost* $u_1^{(2)}$ *by* $u_0^{(1)}$: $\mathcal{L}_1^{(2)} = 3.0$
  - *boost* $u_1^{(1)}$, $u_2^{(2)}$ *by* $u_0^{(0)}$: $\mathcal{L}_1^{(1)} = 6.85, \mathcal{L}_2^{(2)} = 5.03$

This results in constraint matrix

$$\mathbb{V} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \end{pmatrix}.$$

The corresponding check matrix of the code is given by $H = \mathbb{V}A^T = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 1 \end{pmatrix}$. *The obtained code is equivalent to (7,4,3) Hamming code.*

It can be seen that the complexity of the initial symbol allocation algorithm is $O(n)$. The complexity of the symbol boosting algorithm is $\sum_{j=0}^{\zeta-1} 2^{m_j}(1 + \sum_{p=0}^{j-1} 2^{m_p - m_j}) = O(\zeta n)$.

Note that a method for jointly increasing the length and minimum distance a linear block code was suggested in [18]. However, the codes obtained with that method do not possess an efficient decoding algorithm.

E. Dynamic freezing constraints

1) Eliminating low-weight codewords from the code: The performance of error correcting codes under a near-ML decoding algorithm at high SNR depends on their minimum distance d and error coefficient $\mathcal{A}_d$, i.e. the number of codewords of weight d. In order to reduce $\mathcal{A}_d$ and, possibly, increase d we propose to impose dynamic freezing constraints onto symbols $u_i^{(m)}$, which remain unfrozen after application of the above described code construction algorithm, and have low weight $w_{z_{m,i}}$ of the corresponding rows of the code generator matrix. Any non-zero minimum-weight codeword has $u_i^{(m)} = 1$ for at least one such pair (m, i).

Therefore, we propose to employ the method described in Section IV-D to obtain a code of dimension $k + s$, and then take its pseudo-random k-dimensional subcode. This can be implemented as shown in Figure 5. This algorithm selects the last unfrozen symbol of the smallest possible index weight, and generates a linear constraint with random coefficients, which involves this symbol and other symbols, which correspond to outer codes, having smaller or equal indices in the interlinked generalized concatenated code (see [3] for the definition) representation of the obtained code.

Good performance/complexity tradeoff can be obtained by setting $s = \log_2 n$. The freezing constraints obtained with this method are referred to as type-A ones.

```
SUBCODE(u_0^{n-1}, w_0^{n-1}, s, 𝕍, R)
 1  for ω = 1, 2, 4, ...
 2  do for i = n − 1, ..., 1, 0
 3      do if s = 0
 4            then return
 5            else if u_i = U ∧ w_i < ω
 6                    then i_0 ← i; P ← 0
 7                         while i_0 ≥ 2^{m_P}
 8                         do i_0 ← i_0 − 2^{m_P}; P ← P + 1
 9                         for p ← 0 to P
10                         do 𝕍_{R,z_{p,i_0 2^{m_p}}} ← 1
11                            for j ← 0 to i_0 2^{m_p} − 1
12                            do 𝕍_{R,z_{p,j}} ← RAND()
13                            R ← R + 1; s ← s − 1
14  return 𝕍
```

Fig. 5: Construction of type-A dynamic freezing constraints

2) Code design for improved list decodability: The performance of polar codes under the Tal-Vardy algorithm depends not only on their distance properties, but also on the probability of the correct path being killed at an early phase of the decoding algorithm. In order to improve the performance of the obtained codes, we propose to select t frozen symbols with the highest values of $\mathcal{L}_i^{(m_j)}, 0 \le j < s$, and modify the corresponding rows $Pz_{m_j,i}$ of the constraint matrix. Namely, we propose to set $V_{P(z_{m_j,i}),z_{s,h}}, 0 \le h < i2^{m_s - m_j}, 0 \le s \le j$, to independent equiprobable random binary values. The obtained dynamic freezing constraints are referred to as type-B ones. Simulations suggest that selecting $t = \min((n-k)/2, 30)$ results in good codes with a wide range of parameters, although better performance can be obtained by more careful selection of this value.

The proposed construction enables the list decoder to quickly penalize incorrect paths, reducing thus the probability of the correct path being killed.

V. DECODING

The proposed codes can be decoded using the generalized successive cancellation algorithm introduced in [12], as well as its list or sequential extensions [19], [6]. Observe that careful selection of the decoding schedule, i.e. the order of processing of the symbols from different polarizing transformations, is needed to obtain good performance. A simple greedy algorithm for construction of a decoding schedule for chained polar subcodes was presented in [12].

VI. NUMERIC RESULTS

Figure 6 illustrates the performance of the proposed randomized chained polar subcodes (PS) for the case of sequential decoding with list size 32. For comparison, we report also the results for polar codes with quasi-uniform puncturing [10], the bit-reversal shortened codes introduced in [20], BCH-based chained polar subcodes [12], polar subcodes of extended BCH

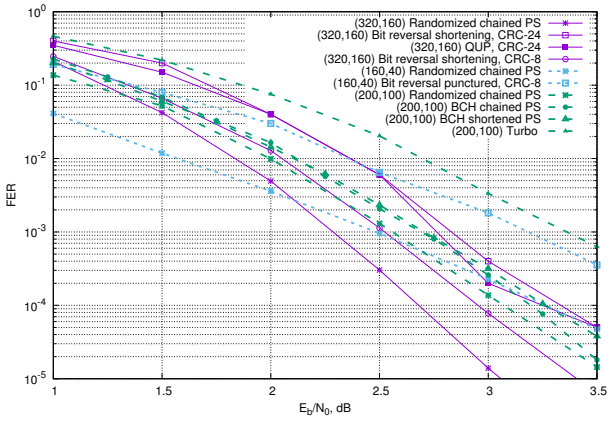


Fig. 6: Performance of randomized chained polar subcodes

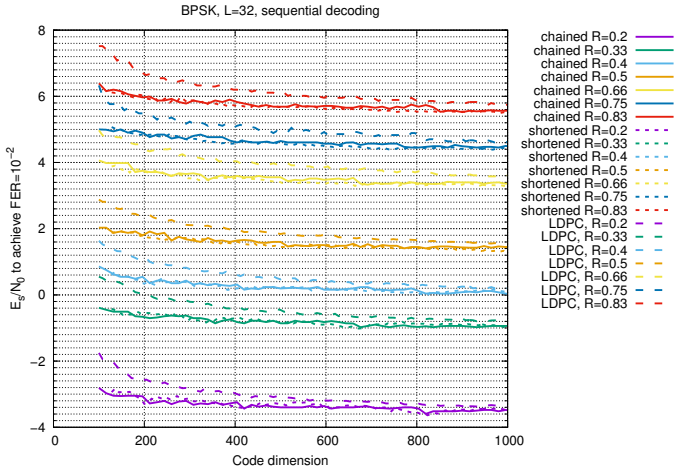


Fig. 7: E_s/N_0 required for achieving $FER = 10^{-2}$

codes with optimized shortening patterns [9], as well as for the LTE turbo code with 8 decoding iterations. It can be seen that the proposed randomized chained polar subcodes significantly outperform other code constructions.

Figure 7 illustrates the SNR required for achieving $FER = 10^{-2}$ by the proposed randomized chained polar subcodes of different rate and dimension. For comparison, we report also the results for LDPC codes suggested for 5G [21] under the belief propagation decoding with 15 iterations, as well as shortened randomized polar subcodes [4]. It can be seen that the proposed codes provide substantially better performance compared to LDPC codes, and approximately the same performance as shortened randomized polar subcodes. However, an important advantage of the proposed construction is that it does not require employing complicated two-dimensional Gaussian approximation given by (9)–(10). Instead, the reliability of bit subchannels, which is required for finding the set of frozen symbol indices, is computed using (12) and (10), i.e. using just simple quadratic expressions, while in [4] the two-dimensional Gaussian approximation has to be used in order to find the reliability of bit subchannels of a shortened polarizing transformation.

VII. Conclusions

In this paper a novel method for construction of polar subcodes of arbitrary length is proposed. The proposed code construction makes use of a number of polarizing transformations of different size, which are combined to obtain the required code length without shortening or puncturing. An algorithm with complexity growing linearly with code length is provided for finding the set of frozen symbol indices and constraint matrix for the proposed codes.

Since no shortening or puncturing is used, evaluation of the reliability of bit subchannels for the proposed codes can be performed using one-dimensional Gaussian approximation. A piecewise-quadratic approximation is proposed for the corresponding function, so that one can avoid evaluation of the transcendent functions, and evaluate bit subchannel reliability with essentially the same complexity as in the case of the binary erasure channel.

References

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

[2] I. Tal and A. Vardy, "List decoding of polar codes," in *Proceedings of IEEE Int. Symp. on Inf. Theory*, 2011, pp. 1–5.

[3] P. Trifonov and V. Miloslavskaya, "Polar subcodes," *IEEE Journal on Sel. Areas in Comm.*, vol. 34, no. 2, pp. 254–266, February 2016.

[4] P. Trifonov and G. Trofimiuk, "A randomized construction of polar subcodes," in *Proc. of IEEE Int. Symp. on Inf, Theory*. 2017.

[5] K. Niu and K. Chen, "Stack decoding of polar codes," *Electronics Letters*, vol. 48, no. 12, pp. 695–697, June 2012.

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

[7] R. Wang and R. Liu, "A novel puncturing scheme for polar codes," *IEEE Comm. Letters*, vol. 18, no. 10, October 2014.

[8] H. Saber and I. Marsland, "An incremental redundancy hybrid ARQ scheme via puncturing and extending of polar codes," *IEEE Transactions on Communications*, vol. 63, no. 11, November 2015.

[9] V. Miloslavskaya, "Shortened polar codes," *IEEE Transactions on Information Theory*, vol. 61, no. 9, pp. 4852–4865, 2015.

[10] K. Niu, K. Chen, and J. Lin, "Beyond turbo codes: Rate-compatible punctured polar codes," in *Proc. of IEEE Int. Comm. Conf*, 2013.

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

[12] P. Trifonov, "Chained polar subcodes," in *Proc. of 11th Int. ITG Conf. on Systems, Comm. and Coding*, 2017.

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

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

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

[16] J. Ha, J. Kim, and S. W. McLaughlin, "Rate-compatible puncturing of low-density parity-check codes," *IEEE Trans. on Inf. Theory*, vol. 50, no. 11, November 2004.

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

[18] M. Grassl, "Computing extensions of linear codes," in *Proceedings of IEEE International Symposium on Information Theory*, 2007.

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

[20] V. Bioglio, F. Gabry, and I. Land, "Low-complexity puncturing and shortening of polar codes," in *Proc. of IEEE WCNC Workshops*, 2017.

[21] Qualcomm, "R1-1709181: LDPC rate compatible design," 3GPP, Tech. Rep., 2017.