Chained Polar Subcodes

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

***Abstract*—A generalization of polar codes is proposed, which makes use of a number of polarizing transformations of different size. This enables one to obtain arbitrary code length and prescribed minimum distance. Generalizations of the successive cancellation algorithm and its list extension to the case of the proposed codes are 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] were shown to provide substantially better performance under list, stack and sequential decoding [2], [4], [5]. Another problem with polar codes is that their length is limited to 2^m. Several puncturing and shortening techniques were suggested for polar codes [6], [7], [8], [9].

In this paper we introduce a generalization of polar codes, which enables one to obtain codes of length other than 2^m, and still use the efficient decoding algorithms developed for polar codes. The proposed approach is based on well-known X4 and XX constructions [10], [11], which are used to combine codes of different length. Essentially, we propose to obtain a codeword as the output of a few polarizing transformations, such that their input vectors have some linear dependencies. Furthermore, we present a straightforward generalization of the Tal-Vardy list decoding algorithm, which can be used for decoding of the proposed codes.

II. Background

A. Polar codes and polar subcodes

A $(n=2^m,k,d)$ polar code over $\mathbb{F}_2$ is a set of vectors $c_0^{n-1}=u_0^{n-1}A_m$, where $a_i^j=(a_i,\dots,a_j)$, $A_m=\begin{pmatrix}1&0\\1&1\end{pmatrix}^{\otimes m}B_m$ is a matrix of the polarizing transformation, B_m is the bit-reversal permutation matrix, $F^{\otimes m}$ denotes m-times Kronecker product of matrix F with itself, $u_i=0,i\in\mathcal{F}$, and $\mathcal{F}\subset\{0,\dots,n-1\}$ is a set of $n-k$ frozen symbol indices. It is possible to show that the polarizing transformation together with a memoryless channel $\mathbf{W}(y|c)$ gives rise to n synthetic bit subchannels with transition probability functions

$$\mathbf{W}_m^{(i)}(y_0^{n-1},u_0^{i-1}|u_i)=\frac{1}{2^{n-1}}\sum_{u_{i+1}^{n-1}}\prod_{j=0}^{n-1}\mathbf{W}(y_j|(u_0^{n-1}A_m)_j).$$

One can compute the capacity $C_{m,i}$ and bit error rate $P_{m,i}$ in each of the subchannels $\mathbf{W}_m^{(i)}$ using the techniques presented in [12], [13]. The set $\mathcal{F}$ is typically constructed as the set of $n-k$ integers with the highest $P_{m,i}$ or lowest $C_{m,i}$.

The successive cancellation (SC) decoding algorithm makes decisions

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

For classical polar codes, the frozen value of $u_i,i\in\mathcal{F}$, is always 0. It is possible to implement these calculations with complexity $O(n\log n)$. The block error probability of the SC algorithm is given by

$$P(\mathcal{F})=1-\prod_{i\notin\mathcal{F}}(1-P_{m,i}).$$

The decoding error probability can be improved by employing list or sequential decoding algorithms [2], [5].

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.

$$u_i=\sum_{j<i}V_{s_i,j}u_j,i\in\mathcal{F},\quad(2)$$

where V is a $(n-k)\times n$ constraint matrix, such that distinct rows end[1] in distinct columns, and s_i is the index of the row ending in column i. Symbols u_i with at least one term in the r.h.s. of (2) are referred to as dynamic frozen. Matrix V is constructed so that codewords c_0^{n-1} belong to some (n,k',d) parent code with check matrix H and sufficiently high minimum distance d, and the SC decoding error probability $P(\mathcal{F})$ is minimized. This can be obtained by computing $V'=HA_m^T$, performing elementary row operations in order to ensure that all rows end in distinct columns, and appending additional $k'-k$ weight-1 rows, having 1's in positions with the highest $P_{m,i}$, such that no other row ends in these positions. The obtained codes are referred to as polar subcodes. In other words, (n,k,d) polar subcode of (n,k',d) parent code C' is its k-dimensional subcode with the smallest error probability under SC decoding.

Extended primitive narrow-sense BCH (e-BCH) codes were shown to be good parent codes. Polar subcodes of e-BCH

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

codes were shown to provide substantially better performance compared to classical polar codes.

It can be seen that any codeword of a polar subcode can be obtained as

$$c = xWA_m,$$

where x is the information vector, and W is a $k \times n$ matrix, such that $WV^T = 0$. Assume without loss of generality that the rows of matrix W start in distinct positions l_j, and each column l_j of matrix W has weight 1. Observe that

$$\mathcal{N} = \{l_j | 0 \leq j < k\} = \{0, \dots, n-1\} \setminus \mathcal{F}.$$

Then the SC algorithm can be equivalently stated as follows:

1) Let $\hat{u}_i \leftarrow 0, i \in \mathcal{F}$
2) For i from 0 to $n-1$, such that $i \notin \mathcal{F}$:
   a) Let $\hat{u}_i \leftarrow \arg\max_{u_i \in \mathbb{F}_2} \mathbf{W}_m^{(i)}(y_0^{n-1}, \hat{u}_0^{i-1}|u_i)$
   b) Let $\hat{u}_j \leftarrow \hat{u}_j + \hat{u}_i, j > i, W_{t_i,j} = 1$, where t_i is the index of row of matrix W having the first non-zero element in column i.

B. Code combining techniques

1) Construction X4: Let $C_i, i \in \{0, , \dots, 3\}$, be (n_i, k_i, d_i) codes over $\mathbb{F}_2$ with generator matrices $G^{(i)}$ and check matrices $H^{(i)}$, such that $C_0 \subset C_1$, $C_2 \subset C_3$, $n_0 = n_1, n_2 = n_3, k_1 - k_0 = k_3 - k_2$, and

$$G^{(1)} = \begin{pmatrix} G^{(0)} \\ G' \end{pmatrix}, \ G^{(3)} = \begin{pmatrix} G^{(2)} \\ G'' \end{pmatrix}.$$

Then

$$G = \begin{pmatrix} G^{(0)} & 0 \\ 0 & G^{(2)} \\ G' & G'' \end{pmatrix}$$

is a generator matrix of $(n_1 + n_2, k_1 + k_2, \min(d_0, d_2, d_1 + d_3))$ code $\mathcal{C}_{X4}$ [10]. In the case of $k_2 = 0$ this is known as construction X.

2) Construction XX: Consider codes $C_i(n_i, k_i, d_i), 0 \leq i \leq 5$, such that $C_0 = C_1 + C_2, C_3 = C_1 \cap C_2$, $k_4 = k_1 - k_3$, and $k_5 = k_2 - k_3$, so that their generator matrices are given by $G^{(1)} = \begin{pmatrix} G' \\ G^{(3)} \end{pmatrix}, G^{(2)} = \begin{pmatrix} G'' \\ G^{(3)} \end{pmatrix}, G^{(0)} = \begin{pmatrix} G' \\ G'' \\ G^{(3)} \end{pmatrix}$. Then matrix

$$G = \begin{pmatrix} G' & G^{(4)} & 0 \\ G'' & 0 & G^{(5)} \\ G^{(3)} & 0 & 0 \end{pmatrix}$$

generates a $(n_0 + n_4 + n_5, k_0, \min(d_3, d_1 + d_4, d_2 + d_5, d_0 + d_4 + d_5)$ code C_{XX} [11]. This can be considered as the result of applying the construction X two times.

III. Chained polar subcodes

We propose to construct a code of length $n = \sum_{i=0}^{z-1} 2^{m_i}$ 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

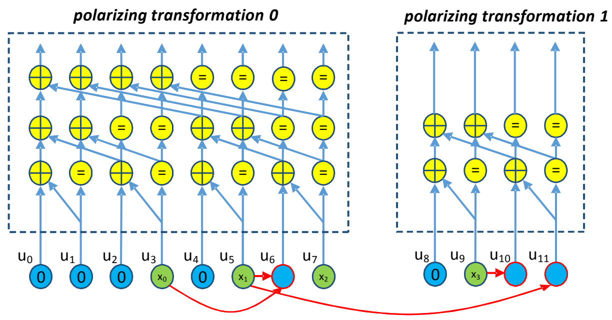

Fig. 1. Encoder for a chained polar subcode

some linear dependencies, and $m_i > m_{i+1}, 0 \leq i < z - 2$. Namely, the codeword is obtained as

$$c = xW \underbrace{\begin{pmatrix} A_{m_0} & 0 & \dots & 0 \\ 0 & A_{m_1} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & A_{m_{z-1}} \end{pmatrix}}_{A},$$

where $x \in \mathbb{F}_2^k$ is an information vector. Such code can be still treated in the framework of dynamic frozen symbols introduces in Section II-A. Figure 1 illustrates an encoder for the proposed code construction. Here frozen symbols are denoted by circles filled in blue. Dynamic frozen symbols are denoted by circles with red border, and red arrows illustrate the linear dependencies.

In order to ensure good performance under list/sequential SC algorithms, the minimum distance of the code must be sufficiently high. To obtain such a code, we propose a generalization of the above described constructions X4 and XX.

Let d be the design distance of the code to be constructed. Let $G^{(i)}$ be $2^{m_i} \times 2^{m_i}$ matrix such that its first κ rows generate $(2^{m_i}, \kappa, d_{i,\kappa})$ code, $d_{i,\kappa} \geq d_{i,\kappa+1}$. In what follows, $G^{(i)}$ are referred to as kernel matrices. Let

$$\delta(i, d) = \max_{\kappa: d_{i,\kappa} < d} d_{i,\kappa}.$$

It can be also assumed without loss of generality that the rows of $W^{(i)} = G^{(i)} A_{m_i}$ start in distinct columns. Let $l_{i,j}$ be the position of the first non-zero element in $W_{j,-}^{(i)}$, where $X_{j,-}$ denotes the j-th row of matrix X.

We propose to construct a generator matrix of a (n, k, d) chained polar subcode as $G = \begin{pmatrix} \tilde{G} \\ \overline{G} \end{pmatrix}$. Both $\tilde{G}$ and $\overline{G}$ are block matrices, and the blocks consist of 2^{m_i} columns. The first part $\tilde{G}$ is a block-diagonal matrix with blocks $G(i, d)$, where $G(i, d)$ is a matrix consisting of some vectors $G_{s,-}^{(i)}$, such that

$$d_{i,s+1} \geq d. \tag{3}$$

The second part $\overline{G}$ has exactly two non-zero blocks in each row. If the first non-zero block in a row is the i-th one, then it is given by $G_{s,-}^{(i)}$, where

$$d_{i,s+1} \geq \delta(i, d). \tag{4}$$

If the second non-zero block in the same row is the j-th one, $j > i$, then it is given by some $G^{(j)}_{t,-}$, where

$$d_{i,t+1} \geq d - \delta(i, d). \tag{5}$$

The selection of rows from kernel matrices is performed in such way, so that each row is used at most once in G. This ensures that the obtained matrix is a full-rank one. Additional rules for construction of G are presented below.

In order to see that the minimum distance of the code generated by G is at least d, consider arbitrary non-zero codeword c of this code. If c has a single non-zero block (say, the i-th one), then this non-zero block must be a codeword of $(2^{m_i}, \kappa, d_{i,\kappa})$ code with $d_{i,\kappa} \geq d$. If c has at least two non-zero blocks (say, blocks i and $j > i$), then the first non-zero block must be a codeword of $(n_i = 2^{m_i}, \kappa', d_{i,\kappa'}), d_{\kappa'} \geq \delta(i, d)$, while the second one must be a codeword of $(2^{m_j}, \kappa'', d_{\kappa''})$ code, where $d_{\kappa''} \geq d - \delta(i, d)$.

In order to ensure that the obtained code provides good performance under (list) SC decoding, we impose the following additional rules on selection of rows from kernel matrices to be used for construction of matrix G:

1) The k rows $G^{(i)}_{s,-}$ used in $\tilde{G}$ or as the first non-zero blocks in $\overline{G}$ are those with the smallest values of $P_{m_i, l_{i,s}}$, satisfying the constraint (3) or (4), respectively.
2) The rows $G^{(j)}_{t,-}$ used as the second non-zero blocks in $\overline{G}$ are those with the smallest values of $P_{m_j, l_{j,t}}$.
3) The τ rows $G^{(i)}_{s,-}$ with $d_{i,s+1} \geq d$ and the highest $P_{m_i,s}$ are used as the first non-zero blocks in $\overline{G}$. Here τ is a parameter of the proposed code construction.
4) For any pair of rows in $\overline{G}$ having non-zero blocks $G^{(i)}_{s,-}$, $G^{(j)}_{t,-}$ and $G^{(i)}_{s',-}$, $G^{(j)}_{t',-}$, respectively, $l_{i,s} < l_{i,s'}$ implies $l_{j,t} < l_{j,t'}$.

Rule 1 ensures that the decoding error probability under the generalized successive cancellation (GSC) algorithm, presented in Section IV-A, is minimized. Rule 4, together with the algorithm described in Section IV-B, allows one to process frozen symbols at early phases of the GSC algorithm. We do not have a proof of optimality for the remaining rules, but simulations suggest that these rules indeed provide codes with good performance.

The precoding matrix $W = GA$ of the obtained (n, k, d) chained polar subcode has the same block structure as the generator matrix G. Hence, its rows start in distinct columns $l_j, 0 \leq j < k$, and the values l_j are equal to some $l_{i,j}$. In order to minimize the error probability under the below described GSC algorithm, one should use kernel matrices $G^{(i)}$, such that the corresponding bit subchannel error probabilities $P_{m_i, l_{i,j}}$ are as small as possible. Following [3], we propose to use kernel matrices, such that their submatrices generate extended primitive narrow sense BCH codes, and the columns of these matrices are arranged in the standard bit ordering.

Example 1. *Consider construction of a* $(26, 11, 6)$ *code for BEC with erasure probability* 0.5*. One obtains* $m_0 = 4, m_1 = 3, m_2 = 1$*. The kernel matrices are*

$$G^{(0)} = \left(\begin{array}{cccccccccccccccc}
1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1\\ \hline
1&1&0&0&0&0&1&1&0&0&1&1&1&1&0&0\\
0&0&0&0&1&1&1&1&1&1&1&1&0&0&0&0\\
1&1&1&1&1&1&1&1&0&0&0&0&0&0&0&0\\
1&0&1&0&1&0&1&0&1&0&1&0&1&0&1&0\\ \hline
0&1&0&0&0&0&1&0&1&1&1&0&1&0&0&0\\
0&1&0&1&1&1&0&0&0&1&1&0&0&0&0&0\\ \hline
0&0&0&0&1&1&0&0&1&1&0&0&0&0&0&0\\
1&1&1&1&0&0&0&0&0&0&0&0&0&0&0&0\\
1&0&1&0&1&0&1&0&0&0&0&0&0&0&0&0\\
1&1&0&0&1&1&0&0&0&0&0&0&0&0&0&0\\ \hline
1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
1&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0\\
1&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0\\
1&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0\\ \hline
1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0
\end{array}\right),$$

$$G^{(1)} = \left(\begin{array}{cccccccc}
1&1&1&1&1&1&1&1\\ \hline
0&0&1&1&1&1&0&0\\
1&1&1&1&0&0&0&0\\
1&0&1&0&1&0&1&0\\ \hline
1&1&0&0&0&0&0&0\\
1&0&1&0&0&0&0&0\\
1&0&0&0&1&0&0&0\\ \hline
1&0&0&0&0&0&0&0
\end{array}\right), G^{(2)} = \begin{pmatrix}1&1\\1&0\end{pmatrix}.$$

These matrices correspond to

$$W^{(0)} = \begin{pmatrix}
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1\\
0&0&0&0&0&0&0&0&0&0&0&1&0&1&1&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0\\
0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0\\
0&0&0&1&0&1&1&0&0&1&0&0&0&0&0&0\\
0&0&0&0&0&1&0&0&0&1&1&0&1&0&0&0\\
0&0&0&0&0&0&0&0&0&1&1&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0\\
0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0\\
0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0\\
0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0\\
0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0
\end{pmatrix}
\begin{matrix}
1.5\cdot 10^{-5}\\ 0.037\\ 0.014\\ 0.0078\\ 0.10\\ 0.77\\ 0.65\\ 0.47\\ 0.23\\ 0.53\\ 0.346\\ 0.90\\ 0.96\\ 0.985\\ 0.992\\ 0.999
\end{matrix}$$

$$W^{(1)} = \begin{pmatrix}
0&0&0&0&0&0&0&1\\
0&0&0&0&0&1&1&0\\
0&0&0&0&0&0&1&0\\
0&0&0&1&0&0&0&0\\
0&0&0&0&1&0&0&0\\
0&0&1&0&0&0&0&0\\
0&1&0&0&0&0&0&0\\
1&0&0&0&0&0&0&0
\end{pmatrix}
\begin{matrix}
0.0039\\ 0.19\\ 0.12\\ 0.32\\ 0.68\\ 0.81\\ 0.88\\ 0.996
\end{matrix}, W^{(2)} = \begin{pmatrix}0&1\\1&0\end{pmatrix}$$

The numbers near the above matrices are the Bhattacharyya parameters of the subchannels corresponding to the first non-zero elements of the corresponding rows. In order to obtain the required code, one should use rows $G^{(0)}_{0,-}, G^{(1)}_{0,-}, G^{(0)}_{3,-}, G^{(0)}_{2,-}, G^{(0)}_{1,-}, G^{(0)}_{4,-}, G^{(1)}_{2,-}, G^{(0)}_{8,-}, G^{(0)}_{10,-}, G^{(0)}_{7,-}, G^{(0)}_{9,-}$ *as first non-zero blocks in matrices* $\tilde{G}$ *and* $\overline{G}$*. Row* $G^{(1)}_{2,-}$ *should be placed in* $\overline{G}$ *and augmented with* $G^{(2)}_{0,-}$*. Rows* $G^{(0)}_{9,-}, G^{(0)}_{7,-}, G^{(0)}_{10,-}, G^{(0)}_{8,-}$ *should be augmented with* $G^{(1)}_{5,-}, G^{(1)}_{3,-}, G^{(1)}_{4,-}, G^{(1)}_{1,-}$*, respectively. Hence, one obtains the following precoding matrix:*

$$\hat{W} = \left(\begin{array}{cccccccccccccccc|cccccccc|cc}
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&1&0&1&1&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\ \hline
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0\\ \hline
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&1\\
0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&1&1&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&1&1&0&0&0
\end{array}\right)$$

IV. Decoding

A. Generalized successive cancellation algorithm

Let W be a precoding matrix of a chained polar subcode, and let l_j be the positions of the first non-zero element in its j-th row. The set of frozen symbols for a chained subcode can be defined as

$$\mathcal{F} = \{0, \dots, n-1\} \setminus \{l_j | 0 \leq j < k\}.$$

It can be assumed without loss of generality that any column l_j of matrix W has weight 1.

Definition 1. *Decoding schedule Φ for $k\times n$ precoding matrix W is a sequence of distinct pairs $\Phi_s=(i_s,\phi_s), 0\leq s<n, 0\leq i_s<z, 0\leq \phi_s<n_{i_s}$, such that:*

1) *if $s'<s''$ and $i_{s'}=i_{s''}$, then $\phi_{s'}<\phi_{s''}$,*
2) *for any s',s'', such that $\Psi(\Phi_{s'})=l_j$ for some $j:0\leq j<k$, and $t=\Psi(\Phi_{s''})>\Psi(\Phi_{s'})$, such that $W_{j,t}=1$, one has $s'<s''$,*

where

$$\Psi(\Phi_s)=\Psi(i_s,\phi_s)=\phi_s+\sum_{j=0}^{i_s-1}2^{m_j}$$

is a global phase function.

The natural decoding schedule is given by the sequence $[(0,0),\ldots,(0,n_0-1),\ldots,(z-1,0),\ldots,(z-1,n_{z-1}-1)]$, so that $\Psi(\Phi_s)=s$.

Let us further define $\tilde{n}_s=\sum_{j=0}^{i_s-1}2^{m_j}$ and $\overline{n}_s=\sum_{j=0}^{i_s}2^{m_j}$.

Given some schedule Φ, one can implement decoding of the chained polar subcode using the following generalized successive cancellation (GSC) algorithm:

1) Let $\hat{u}_i\leftarrow 0, i\in\mathcal{F}$
2) For s from 0 to $n-1$, such that $\Psi(\Phi_s)\notin\mathcal{F}$:
   a) Let
   $$\hat{u}_i\leftarrow\arg\max_{u_i\in\mathbb{F}_2}\mathbf{W}_{m_{i_s}}^{(\phi_s)}(y_{\tilde{n}_s}^{\overline{n}_s-1},\hat{u}_{\tilde{n}_s}^{i-1}|u_i),$$
   where $i=\Psi(\Phi_s)$.
   b) Let $\hat{u}_j\leftarrow\hat{u}_j+\hat{u}_i$ for all $j>i$, such that $W_{t_i,j}=1$, where t_i is the index of row of matrix W which starts in column i.

This algorithm can be equivalently represented in a form similar to (1) using (2) for computing of the values of the frozen symbols.

It can be seen that the error probability of the above algorithm is given by

$$P=1-\prod_{j=0}^{k-1}(1-\Pi_{l_j}),$$

where

$$\Pi=(P_{m_0,0},\ldots,P_{m_0,n_0-1},\ldots,P_{m_{z-1},0},\ldots,P_{m_{z-1},n_{z-1}-1}).$$

Observe that P does not depend on the decoding schedule. It is minimized by setting W to be a block-diagonal matrix with distinct weight-1 rows, having 1's in columns j with the smallest Π_j. This corresponds to independent encoding of the data with z classical Arikan polar codes of length $2^{m_i}, 0\leq i<z$. The proposed chained polar subcodes cannot provide lower error probability than z independent classical polar codes of lengths 2^{m_j} *under the GSC algorithm.* However, significant performance gain is possible if list decoding is used.

B. Interleaved decoding schedule

The estimates $\hat{u}_i$ obtained by the above described generalized SC algorithm are not optimal, since they do not take into account the constraints on symbols $u_j, j\in\mathcal{F}, j>i$. The standard workaround for this problem is list decoding, which essentially defers the decision on u_i until more frozen symbols are processed [2]. The same approach is possible in the case of chained polar subcodes. However, its efficiency can be improved by constructing an interleaved decoding schedule, which causes the list decoder to process the frozen symbols from different polarizing transformations as early as possible, subject to the constraints given in Definition 1.

In order to improve the accuracy of the estimates $\hat{u}_i$, we propose to construct the decoding schedule in such way, so that the frozen symbols are processed as soon as possible. This can be implemented by the following algorithm.

1) Let $w_i\leftarrow \mathrm{wt}(W_{-,i}), 0\leq i<n$.
2) Let $w_i\leftarrow w_i-1, i\notin\mathcal{F}$
3) Let r_i be the cardinality of the largest set $T_i=\{i+1,i+2,\ldots,j\}$, such that $T_i\cap\mathcal{F}=\emptyset$.
4) Let $\phi_j=0, 0\leq j<z$.
5) Let $s=0$.
6) Let $\Phi_s\leftarrow(j,\phi_j)$, where j is the integer, such that $w_i=0$, r_i is the least possible, and $i=\Psi(j,\phi_j)$.
7) Let $\phi_j\leftarrow\phi_j+1$.
8) If $i\notin\mathcal{F}$, let $w_p\leftarrow w_p-1$ for all $p:W_{t_i,p}=1$, where t_i is the index of row of W, which starts in column i.
9) Let $s\leftarrow s+1$. If $s<n$, go to step 6.

Here w_i is the number of non-frozen symbols symbol u_i depends on, and r_i is the number of non-frozen symbols immediately following symbol u_i, i.e. it is the distance to the nearest frozen symbol $u_{j+1}, j\geq i$.

We have neither a proof of optimality of the above algorithm, nor any theoretical performance estimates for list decoding with the schedule produced by it. However, simulations show that it provides rather good performance.

C. List decoding

In this section we present a generalization of the Tal-Vardy list decoding algorithm for the proposed codes. We consider the min-sum approximation introduced in [15]. For the case of classical polar codes of length m it makes use of path metric given by

$$R(u_0^\phi)=R(u_0^{\phi-1})+\tau(L_m^{(\phi)}(u_0^{\phi-1}|y_0^{2^m-1}),u_\phi),\qquad(6)$$

where $R(\epsilon)=0$, ϵ is an empty vector,

$$\tau(S,u)=\begin{cases}0, & \mathrm{sgn}(S)=(-1)^u\\ -|S|, & \text{otherwise},\end{cases}$$

the log-likelihood ratios are given by

$$L_\lambda^{(2i)}(u_0^{2i-1},y_0^{N-1})=\mathrm{sgn}(a)\,\mathrm{sgn}(b)\min(|a|,|b|),\qquad(7)$$
$$L_\lambda^{(2i+1)}(u_0^{2i},y_0^{N-1})=(-1)^{u_{2i}}a+b,\qquad(8)$$

DECODE(y_0^{n-1}, l)

1 $\nu \leftarrow 0$
2 **for** $i \leftarrow 0$ **to** $z-1$
3 **do** $\Lambda \leftarrow E_i$.ASSIGNINITIALPATH()
4 $L = E_i$.GETARRAYPOINTERL$(0, \Lambda)$
5 $L[j] = y_{\nu+j}, 0 \leq j < 2^{m_i}$
6 E_i.RECURSIVELYCALCL$(m_i, 0)$
7 $\nu \leftarrow \nu + 2^{m_i}$
8 $ActivePath[\Lambda] = true; R_\Lambda = 0$
9 **for** $s \leftarrow 0$ **to** $n-1$
10 **do if** $\Psi(i_s, \phi_s) \in \mathcal{F}$
11 **then** E_{i_s}.CONTINUEPATHS_FROZENBIT(ϕ_s)
12 **else** E_{i_s}.CONTINUEPATHS_UNFROZENBIT(ϕ_s)
13 **if** $\phi_s \equiv 1 \bmod 2$
14 **then** E_{i_s}.RECURSIVELYUPDATEC(m_{i_s}, ϕ_s)
15 **if** $\phi_s < 2^{m_s} - 1$
16 **then** E_{i_s}.RECURSIVELYCALCL$(m_{i_s}, \phi_s + 1)$
17 $\Lambda_0 = \arg\max_s R_s$
18 **for** $i \leftarrow 0$ **to** $z-1$
19 **do** $C^{(i)} = E_i$.GETARRAYPOINTERC$(0, \Lambda_0)$
20 **return** $[C^{(0)}, \dots, C^{(z-1)}]$

Fig. 2. Generalized Tal-Vardy min-sum list decoding algorithm

for $0 < \lambda \leq m$, and $a = L_{\lambda-1}^{(i)}(u_{0,e}^{2i-1} \oplus u_{0,o}^{2i-1}, y_0^{\frac{N}{2}-1})$, $b = L_{\lambda-1}^{(i)}(u_{0,o}^{2i-1}, y_{\frac{N}{2}}^{N-1})$. This definition of path metric can be immediately extended to the case of list decoding of chained polar subcodes with any schedule.

The decoder essentially consists of z list decoders for the polarizing transformations $A_{m_i}, 0 \leq i < z$, which operate synchronously. That is, all path clone and kill operations are performed simultaneously for all list decoders. Let E_i denote the collection of Tal-Vardy data structures for A_{m_i}. By $E_i.FunctionName$ we denote a function which operates with the data corresponding to A_{m_i}.

Figure IV-C illustrates the decoding algorithm. The input parameters are the received noisy vector y_0^{n-1}, and list size l. Observe that the $AssignInitialPath$ function on all iterations returns the same index Λ. It is possible to simplify the implementation by removing the identical operations performed on all E_i, but this would require many minor changes in various auxiliary algorithms. Therefore, for the sake of brevity, we try to keep the description as close as possible to the original Tal-Vardy algorithm.

The function $RecursivelyCalcL$ computes the log-likelihood ratios $L_{m_{i_s}}$ according to (7)–(8). Observe that the call sequence (see lines 6 and 16) for this function is different from that of the original Tal-Vardy algorithm. This is needed in order to ensure the consistency of shared memory data structures.

The functions $ContinuePaths\_FrozenBit$ and $ContinuePaths\_UnfrozenBit$ extend all active paths $u_{\tilde{n}_s}^{\tilde{n}_s+\phi_s-1}$ by 1 bit, and compute the updated path metrics $R_\Lambda, 0 \leq \Lambda < l$, according to (6).

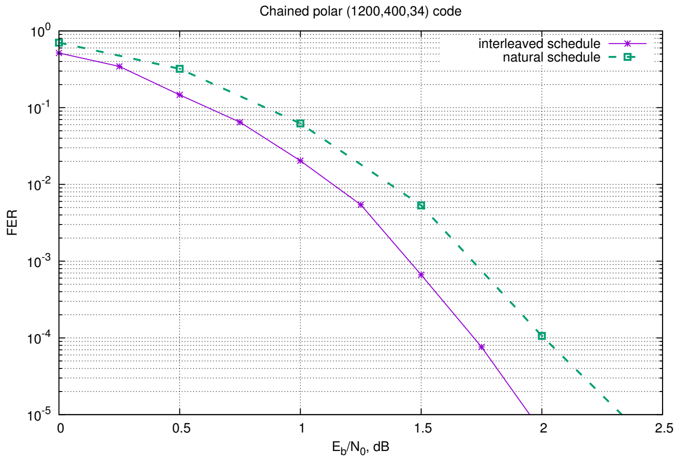

Fig. 3. Performance of list decoding with different schedules

$ContinuePaths\_FrozenBit$ computes the values of dynamic frozen symbols according to (2), while $ContinuePaths\_UnfrozenBit$ considers two possible values of $u_{\Psi(\Phi_s)}$ for each active path, and selects l paths with the highest metric R_Λ. Finally, the most probable codeword is selected in lines 17–20.

The complexity of the generalized Tal-Vardy algorithm is given by $O(L\sum_{i=0}^{z-1} n_i \log n_i)$. It can be reduced by constructing a sequential algorithm, similarly to [5].

V. NUMERIC RESULTS

In this section we consider the performance of chained polar subcodes in the case of AWGN channel and BPSK modulation. The generalized Tal-Vardy algorithm (see Section IV-C) with list size $l = 32$ was used for decoding of the proposed codes.

Figure 3 illustrates the performance of the list decoding algorithm under the natural (i.e. $\Psi(\Phi_s) = s$) and interleaved schedules. It can be seen that the proposed interleaved decoding schedule results in substantially better list decoder performance.

Figure 4 illustrates the performance of various chained polar subcodes. For comparison, we report the performance of shortened/punctured[2] polar codes, and LTE turbo codes (decoding with 8 iterations). It can be seen that chained polar subcodes outperform shortened/punctured codes. The gain becomes most significant in the high SNR region due to higher minimum distance of chained polar subcodes and/or the ability to process frozen symbols at earlier phases of the list decoding algorithm. Observe also, that chained polar subcodes outperform LTE turbo codes.

VI. CONCLUSIONS

In this paper a generalization of polar subcodes was presented, which makes use of a few polarizing transformations

[2] An algorithm for finding optimal shortening pattern for polar codes was proposed in [8]. It can be easily modified to find optimal puncturing patterns [16].

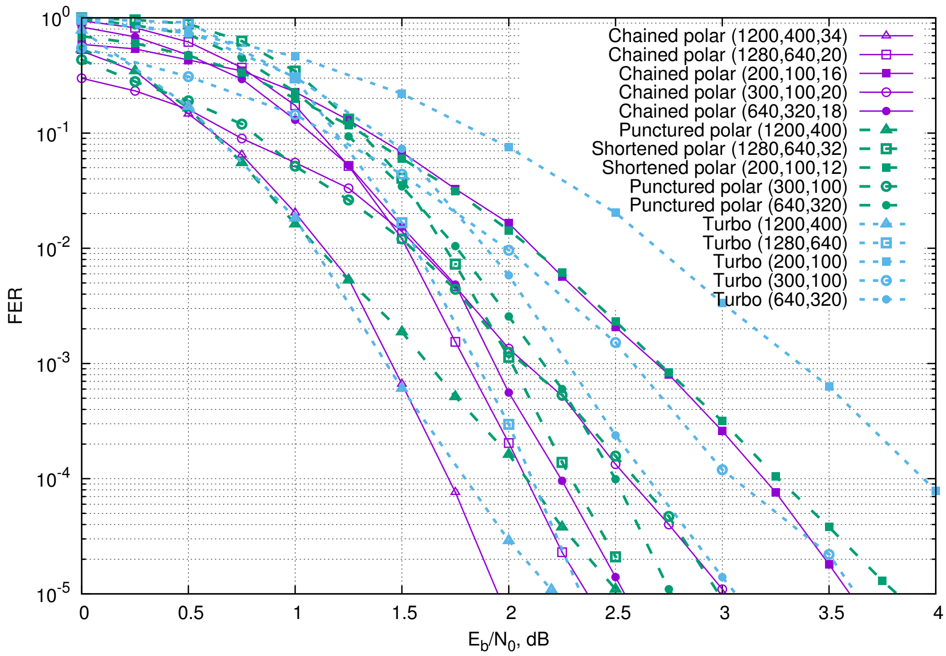

Fig. 4. Performance of chained polar subcodes

of different dimensions. The input symbols of these transformations are jointly pre-coded, in order to ensure that the obtained code has sufficiently high minimum distance. This approach enables one to obtain codes of arbitrary length, and represents an alternative to shortening and puncturing. The proposed codes can be decoded using simple generalizations of the successive cancellation algorithm, and its list/sequential extensions.

It was also shown that further performance gain can be obtained by varying the decoding schedule, i.e. the order of detection of the input symbols of the polarizing transformations. A simple method for construction of the decoding schedule was proposed, which attempts to process frozen symbols in the constituent polarizing transformations as soon as possible.

Both the proposed code construction and the method for construction of the decoding schedule are heuristical. The performance of the proposed codes under the generalized Tal-Vardy list decoding algorithm is dominated by the events of the correct path being killed by the decoder at early phases. Any progress in the performance analysis of list decoding of classical polar codes would enable design of a better decoding schedule, as well as construction of better chained polar subcodes.

Acknowledgement

The author thanks Dr. V. Miloslavskaya for providing the software for construction of shortened and punctured polar codes.

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, pp. 3051–3073, July 2009.
[2] I. Tal and A. Vardy, "List decoding of polar codes," in *Proceedings of IEEE International Symposium on Information Theory*, 2011, pp. 1–5.
[3] P. Trifonov and V. Miloslavskaya, "Polar subcodes," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 2, pp. 254–266, February 2016.
[4] K. Niu and K. Chen, "Stack decoding of polar codes," *Electronics Letters*, vol. 48, no. 12, pp. 695–697, June 2012.
[5] V. Miloslavskaya and P. Trifonov, "Sequential decoding of polar codes," *IEEE Communications Letters*, vol. 18, no. 7, pp. 1127–1130, 2014.
[6] R. Wang and R. Liu, "A novel puncturing scheme for polar codes," *IEEE Communications Letters*, vol. 18, no. 10, October 2014.
[7] 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.
[8] V. Miloslavskaya, "Shortened polar codes," *IEEE Transactions on Information Theory*, vol. 61, no. 9, pp. 4852–4865, 2015.
[9] K. Niu, K. Chen, and J. Lin, "Beyond turbo codes: Rate-compatible punctured polar codes," in *Proceedings of IEEE International Communications Conference*, 2013.
[10] N. J. A. Sloane, S. M. Reddy, and C.-L. Chen, "New binary codes," *IEEE Transactions on Information Theory*, vol. 18, no. 4, July 1972.
[11] W. Alltop, "A method for extending binary linear codes," *IEEE Transactions on Information Theory*, vol. 30, no. 6, November 1984.
[12] I. Tal and A. Vardy, "How to construct polar codes," *IEEE Transactions on Information Theory*, vol. 59, no. 10, pp. 6562–6582, October 2013.
[13] P. Trifonov, "Efficient design and decoding of polar codes," *IEEE Transactions on Communications*, vol. 60, no. 11, pp. 3221 – 3227, November 2012.
[14] N. Hussami, S. B. Korada, and R. Urbanke, "Performance of polar codes for channel and source coding," in *Proceedings of IEEE International Symposium on Information Theory*, 2009, pp. 1488–1492.
[15] A. Balatsoukas-Stimming, M. B. Parizi, and A. Burg, "LLR-based successive cancellation list decoding of polar codes," *IEEE Transactions On Signal Processing*, vol. 63, no. 19, October 2015.
[16] V. Miloslavskaya, Private communication, 2015.