Polar Subcodes

Peter Trifonov, *Member, IEEE,* and Vera Miloslavskaya, *Member, IEEE*

.

***Abstract*—An extension of polar codes is proposed, which allows some of the frozen symbols, called dynamic frozen symbols, to be data-dependent. A construction of polar codes with dynamic frozen symbols, being subcodes of extended BCH codes, is proposed. The proposed codes have higher minimum distance than classical polar codes, but still can be efficiently decoded using the successive cancellation algorithm and its extensions. The codes with Arikan, extended BCH and Reed-Solomon kernel are considered. The proposed codes are shown to outperform LDPC and turbo codes, as well as polar codes with CRC.**

***Keywords*—Polar codes, generalized concatenated codes, BCH codes, Reed-Solomon codes.**

I. Introduction

Polar codes were recently shown to be able to achieve the capacity of binary input memoryless output-symmetric channels [1]. Low-complexity construction, encoding and decoding algorithms are available for polar codes. However, the performance of polar codes of moderate length appears to be quite poor. This is both due to suboptimality of the successive cancellation (SC) decoding algorithm and low minimum distance of polar codes. The first problem can be solved by employing list/stack SC decoding techniques [2]–[7], which far outperform the SC algorithm. Alternatively, one can use the belief propagation algorithm [8]. Its performance, however, is still inferior to list/stack SC decoding.

The second problem can be solved by constructing a generalized concatenated code with inner polar codes [9]–[11], or employing a serial concatenation of an error detecting or error correcting code and a polar code [2], [8], [12]–[14]. However, in the second case it is not clear how the parameters of the outer codes affect the minimum distance and finite-length performance of the concatenated code.

It was shown recently that a sequence of linear codes achieves capacity on a memoryless erasure channel under MAP decoding if their blocklengths are strictly increasing, rates converge to some $r \in (0,1)$, and the permutation group of each code is doubly transitive [15], [16]. This class of codes includes Reed-Muller (RM) and extended primitive narrow-sense BCH (EBCH) codes. Observe that RM codes can be considered as a special case of polar codes. On the other hand, EBCH codes are known to have much higher minimum distance than comparable RM codes, and are therefore likely to provide better finite length performance. However, there are still no efficient MAP decoding algorithms for these codes.

It was suggested in [17] to construct subcodes of RM codes, which can be efficiently decoded by a recursive list decoding algorithm. In this paper we generalize this approach, and propose a code construction "in between" polar codes and EBCH codes. The proposed codes can be efficiently decoded using the techniques developed in the area of polar coding, but provide much higher minimum distance, which can be accurately controlled. The obtained codes outperform state-of-the art LDPC, turbo and polar codes. More specifically, in Section III we introduce an extension of generalized concatenated codes (GCC), called interlinked generalized concatenated codes (IGCC). Recursive application of this construction enables one to represent a linear block code in a form which, in principle, enables its decoding by the SC algorithm (Section IV). This form, called polar codes with dynamic frozen symbols, can be considered as a generalization of polar codes. We show that EBCH codes are particularly well suited for such representation, although their SC decoding is still not very efficient. Furthermore, we present a special case of IGCC, called polar subcodes, with good performance under the SC algorithm and its derivatives (Section V). The proposed codes are subcodes of EBCH codes. We consider polar subcodes with Arikan, EBCH and Reed-Solomon kernel. Simulation results presented in Section VI show that the proposed codes outperform state-of-the-art polar, LDPC and turbo codes.

II. Background

A. Generalized concatenated codes

A generalized concatenated code (GCC) [18] over $\mathbb{F}_q$ is defined using a family of nested inner (n, k_i, d_i) codes $\mathcal{C}_i : \mathcal{C}_0 \supset \mathcal{C}_1 \supset \cdots \supset \mathcal{C}_{\nu-1}$, and a family of outer (N, K_i, D_i) codes $\mathbb{C}_i$, where the i-th outer code is defined over $\mathbb{F}_{q^{k_i-k_{i+1}}}, 0 \leq i < \nu, k_\nu = 0$. It will be assumed in this paper that $k_i = k_{i+1}+1, \nu = n$. Let $\mathcal{G}$ be a $n \times n$ matrix, such that its rows $i, \ldots, n-1$ generate code $\mathcal{C}_i$. GCC encoding is performed as follows. First, partition a data vector into n blocks of size $K_i, 0 \leq i < n$. Second, encode these blocks with codes $\mathbb{C}_i$ to obtain codewords $(\tilde{c}_{i,0}, \ldots, \tilde{c}_{i,N-1})$. Finally, multiply vectors $(\tilde{c}_{0,j}, \ldots, \tilde{c}_{n-1,j}), 0 \leq j < N$, by $\mathcal{G}$ to obtain a GCC codeword $(c_{0,0}, \ldots, c_{n-1,0}, c_{0,1}, \ldots, c_{n-1,N-1})$. Figure 1 illustrates this construction. A GCC generator matrix can be obtained as

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

The authors are with the Distributed Computing and Networking Department, Saint-Petersburg Polytechnic University, Polytechnicheskaya str., 21, office 103, 194021, Saint-Petersburg, Russia, Email: petert@dcn.icc.spbstu.ru.

This work was partially presented at IEEE Information Theory Workshop 2013 and International Symposium on Information Theory and Its Applications 2014.

This work was partially supported by the Russian Foundation for Basic Research under the grant 12-01-00365-a and the Samsung Global Research Outreach grant.

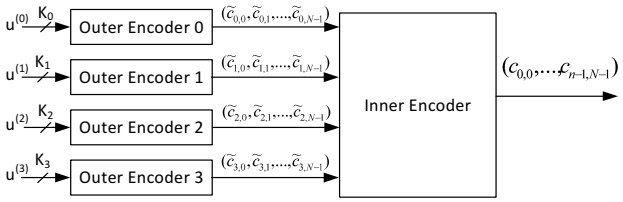


Fig. 1. Encoding with a generalized concatenated code

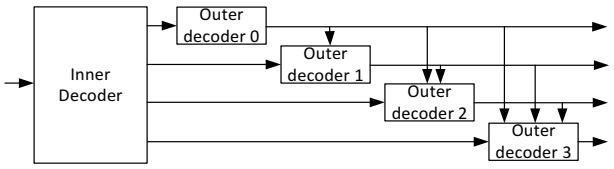


Fig. 2. Multistage decoding

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

GCC can be decoded with a multistage decoding (MSD) algorithm [19]–[21]. For $i = 0, 1, \ldots, N-1$, this algorithm takes as input noisy instances $y_{t,j}$ of codeword symbols $c_{t,j}, 0 \leq t < n, 0 \leq j < N$, successively computes estimates of $\tilde{c}_{i,j}$ using a SISO decoder of $\mathcal{C}_i$, and passes these estimates to a decoder of $\mathbb{C}_i$ to recover the corresponding codeword. Then it proceeds with decoding of $\mathcal{C}_{i+1}$ and $\mathbb{C}_{i+1}$, as shown in Figure 2. The performance of the MSD algorithm depends strongly on parameters of outer codes. An extensive survey of various methods for their selection can be found in [21].

B. Polar codes

$(n = l^m, k)$ polar code over $\mathbb{F}_q$ is a linear block code generated by k rows of matrix $A = B_{l,m} F_l^{\otimes m}$, where $B_{l,m}$ is the digit-reversal permutation matrix, F_l is a $l \times l$ matrix called kernel (e.g. $F_2 = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$ is the Arikan kernel), and $\otimes m$ denotes m-times Kronecker product of the matrix with itself [1]. The digit-reversal permutation maps integer $i = \sum_{j=0}^{m-1} i_j l^j, 0 \leq i_j < l$, onto $\sum_{j=0}^{m-1} i_j l^{m-1-j}$. The particular rows to be used in a generator matrix are selected so that the error probability under the below described successive cancellation (SC) decoding algorithm is minimized. Hence, a codeword of a classical polar code is obtained as $c = uA$, where $u_i = 0, i \in \mathcal{F}$, and $\mathcal{F} \subset \{0, \ldots, n-1\}$ is the set of $n-k$ frozen symbol indices. It is possible to show that matrix A transforms the original binary input memoryless output-symmetric channel $W_1^{(0)}(y|c)$ into bit subchannels $W_n^{(i)}(y_0^{n-1}, u_0^{i-1}|u_i)$, the capacities of these subchannels converge with m to 0 or 1 symbols per channel use, and the fraction of subchannels with capacity close to 1 converges to the capacity of $W_1^{(0)}(y|c)$. Here $a_s^t = (a_s, \ldots, a_t)$, and $y_0, \ldots, y_{n-1}$ are the noisy symbols obtained by transmitting codeword symbols $c_0, \ldots, c_{n-1}$ over a binary input memoryless output-symmetric channel $W_1^{(0)}(y|u)$.

The SC decoding algorithm at phase i computes $W_n^{(i)}(y_0^{n-1}, u_0^{i-1}|u_i), u_i \in \mathbb{F}_q$ (or $W_n^{(i)}(u_0^i|y_0^{n-1})$, which is more convenient for implementation), and makes decisions

$$\widehat{u}_i = \begin{cases} \arg\max_{u_i \in \mathbb{F}_q} W_n^{(i)}(u_0^i|y_0^{n-1}), & i \notin \mathcal{F} \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

This decision is used at subsequent steps instead of the true value of u_i to determine the values of $u_{i+1}, \ldots, u_{n-1}$. It was shown in [1] that these calculations can be implemented with complexity $O(n \log n)$. For example, in the case of $l = 2, q = 2$ these probabilities can be computed as

$$W_n^{(2i)}(u_0^{2i}|y_0^{n-1}) = \sum_{u_{2i+1}=0}^{1} W_{\frac{n}{2}}^{(i)}(u_{0,even}^{2i+1} \oplus u_{0,odd}^{2i+1}|y_0^{\frac{n}{2}-1}) W_{\frac{n}{2}}^{(i)}(u_{0,odd}^{2i+1}|y_{\frac{n}{2}}^{n-1}) \quad (2)$$

$$W_n^{(2i+1)}(u_0^{2i+1}|y_0^{n-1}) = W_{\frac{n}{2}}^{(i)}(u_{0,even}^{2i+1} \oplus u_{0,odd}^{2i+1}|y_0^{\frac{n}{2}-1}) W_{\frac{n}{2}}^{(i)}(u_{0,odd}^{2i+1}|y_{\frac{n}{2}}^{n-1}). \quad (3)$$

For $q = 2$ the Bhattacharyya parameters $Z_{n,i}$ of the bit subchannels $W_n^{(i)}(y_0^{n-1}, u_0^{i-1}|u_i)$ satisfy [22]

$$Z_{n/l,i}^{\Delta_j} \leq Z_{n,il+j} \leq 2^{l-j} Z_{n/l,i}^{\Delta_j}, \quad (4)$$

where $Z_{1,0}$ is the Bhattacharyya parameter of the original binary memoryless symmetric channel, and $\Delta_i, 0 \leq i < l$, are partial distances of matrix F_l. Similar bounds are provided in [23] for $q > 2$. For the case of $l = 2$ (Arikan kernel), one can obtain more precise estimates as [1], [24]

$$Z_{n/2,i}\sqrt{2 - Z_{n/2,i}^2} \leq Z_{n,2i} \leq 2Z_{n/2,i} - Z_{n/2,i}^2 \quad (5)$$

$$Z_{n,2i+1} = Z_{n/2,i}^2. \quad (6)$$

Furthermore, for the case of the binary erasure channel, one has $Z_{n,2i} = 2Z_{n/2,i} - Z_{n/2,i}^2$.

Let $P_i = 1 - P\{C_i|C_0, \ldots, C_{i-1}\}$ be the error probability of symbol u_i under SC decoding, where C_i is the event corresponding to correct estimation of symbol u_i. Then the SC decoding error probability is given by

$$P = 1 - \prod_{i \notin \mathcal{F}} (1 - P_i). \quad (7)$$

Efficient techniques are available for computing P_i in the case of Arikan kernel [9], [25]. The standard way to construct practical polar codes is to select $\mathcal{F}$ as the set of $n-k$ indices i with the highest error probability P_i.

For any $s, 0 < s \leq m$, polar codes can be considered as GCC with inner codes generated by the rows of matrix $F_l^{\otimes s}$, and outer codes generated by some submatrices of $F_l^{\otimes(m-s)}$. The SC decoding algorithm can be considered as an instance of the MSD method, where symbol-by-symbol decoding of outer codes is used. It was shown in [9] that significant performance improvement can be achieved by employing near-ML decoding algorithms for outer codes. Even better performance can be obtained by employing list or stack decoding algorithms [2]–[6]. These algorithms keep track of a number of vectors $\hat{u}_0^{i-1}$,

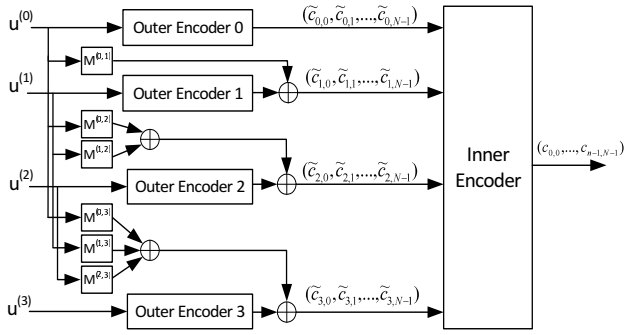


Fig. 3. Interlinked generalized concatenated code

and at each step increase the length of one or more of these vectors by 1, and compute probabilities $W_n^{(i)}(\hat{u}_0^i|y_0^{n-1})$ (or related values). Vectors with low probabilities are discarded, so that there are at most L vectors of length i for each i. The worst-case complexity of these algorithms is $O(Ln\log n)$.

C. BCH codes

An $(n = q^m, k, \geq d)$ extended primitive narrow-sense BCH (EBCH) code is a set of vectors $c_0^{n-1} \in \mathbb{F}_q^n$, such that $\sum_{i=0}^{n-1} c_i x_i^j = 0, 0 \leq j < d-1$, where $(x_0, \ldots, x_{n-1})$ is a vector of distinct values of $\mathbb{F}_{q^m}$, called code locators. Setting $x_0 = 0, x_i = \alpha^{i-1}, 1 \leq i < n$, results in an extended cyclic code with generator polynomial $g(x) = LCM(M_1(x), \ldots, M_{d-2}(x))$, where $M_i(x)$ is a minimal polynomial of α^i, and α is a primitive element of $\mathbb{F}_{q^m}$. However, in this paper, unless stated otherwise, it will be assumed that x_i are arranged in the standard digit order, where $x_i = \sum_{j=0}^{m-1} X_{i,j}\beta_j$, $i = \sum_{j=0}^{m-1} X_{i,j}q^j, X_{i,j} \in \{0, \ldots, q-1\}$, and $\beta_0, \ldots, \beta_{m-1}$ is some basis of $\mathbb{F}_{q^m}$. By abuse of notation, $X_{i,j}$ denotes here both an integer, and the corresponding element of $\mathbb{F}_q$.

III. Interlinked Generalized Concatenated Codes

A. The construction

In this section we present an extension of the generalized concatenated codes, called interlinked GCC (IGCC). This extension can be used to represent a broad class of linear block codes. It enables one to decode such codes using the techniques developed in the area of generalized concatenated and multilevel coding. These decoding algorithms, however, are not guaranteed to perform well for an arbitrary IGCC. But it will be shown below how to construct IGCC with good performance under MSD (actually, SC) and its list extensions.

Interlinked GCC encodes the subvector $u^{(i)} \in \mathbb{F}_q^{K_i}$ of the data vector not with the outer code $\mathbb{C}_i$, as in the classical GCC, but with its coset given by $\mathbb{C}_i + \left(\sum_{s=0}^{i-1} u^{(i)} M^{(s,i)}\right)$, where $M^{(s,i)} \in \mathbb{F}_q^{K_s \times N}$ are some matrices, as shown in Figure 3. This results in a linear block code of length Nn and dimension $\sum_{i=0}^{n-1} K_i$. It is, however, quite difficult to estimate the minimum distance of the obtained code. Obviously, for any pair of non-negative integers $\eta, k : \eta > k$, if there exists a (η, k, d) GCC, then there also exists a $(\eta, k, \geq d)$ IGCC.

The MSD algorithm can be used to decode IGCC. However, one needs to perform decoding not in outer codes, but in their cosets. This can be done with any decoder for $\mathbb{C}_i$, provided that its input LLRs are appropriately adjusted.

B. Generalized Plotkin decomposition of linear codes

As a special case of IGCC, which corresponds to the case of inner codes generated by rows of $\mathcal{G} = F_2 = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$, we consider an extension of the classical Plotkin construction. This extension will be used below to derive a generalization of Arikan polar codes.

Theorem 1. *Any linear $(2n, k, d)$ code $\mathcal{C}$ has a generator matrix given by*

$$G = \begin{pmatrix} I_{k_1} & 0 & \tilde{I} \\ 0 & I_{k_2} & 0 \end{pmatrix} \begin{pmatrix} G_1 & 0 \\ G_2 & G_2 \\ G_3 & G_3 \end{pmatrix}, \tag{8}$$

where I_l is a $l \times l$ identity matrix, $G_i, 1 \leq i \leq 3$, are $k_i \times n$ matrices, $k = k_1 + k_2$, and $\tilde{I}$ is obtained by stacking a $(k_1 - k_3) \times k_3$ zero matrix and I_{k_3} , where $k_3 \leq k_1$.

Proof: Let $\tilde{G} = (G' \quad G'')$, where G' and G'' are some $k \times n$ matrices, be a generator matrix of the code, and let $\tilde{H} = (H' \quad H'')$ be the corresponding parity check matrix. Let G_2 be a maximum rank solution of matrix equation $G_2(H' + H'')^T = 0$. Gaussian elimination can be used to construct matrix $G = Q\tilde{G} = \begin{pmatrix} G_5 & 0 \\ G_4 & G_3 \\ G_2 & G_2 \end{pmatrix}$, such that Q is an invertible matrix, rows of G_3 are linearly independent with rows of G_2, and $k = k_2 + k_3 + k_5$. It can be seen that

$$G = \begin{pmatrix} I_{k_5} & 0 & 0 & 0 \\ 0 & I_{k_3} & 0 & I_{k_3} \\ 0 & 0 & I_{k_2} & 0 \end{pmatrix} \begin{pmatrix} G_5 & 0 \\ G_4 - G_3 & 0 \\ G_2 & G_2 \\ G_3 & G_3 \end{pmatrix}. \tag{9}$$

Then the statement follows by setting $G_1 = \begin{pmatrix} G_5 \\ G_4 - G_3 \end{pmatrix}$. ■

Another way to construct G_1 is to compute $G' + G''$, and eliminate linearly dependent rows from the obtained matrix.

Classical Plotkin concatenation of two codes corresponds to the case of $k_3 = 0$, so the representation of a generator matrix in the form (8) will be referred to as a generalized Plotkin decomposition (GPD) of G or the corresponding code $\mathcal{C}$. Applying the GPD to equivalent codes may result in codes $\mathcal{C}_1, \mathcal{C}_2$ with different dimensions and performance.

$$u_0^{15}\begin{pmatrix}1\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\\ 1\,0\,0\,0\,0\,0\,0\,0\,1\,0\,0\,0\,0\,0\,0\,0\\ 1\,0\,0\,0\,1\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\\ 1\,0\,0\,0\,1\,0\,0\,0\,1\,0\,0\,0\,1\,0\,0\,0\\ 1\,0\,1\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\\ 1\,0\,1\,0\,0\,0\,0\,0\,1\,0\,1\,0\,0\,0\,0\,0\\ 1\,0\,1\,0\,1\,0\,1\,0\,0\,0\,0\,0\,0\,0\,0\,0\\ 1\,0\,1\,0\,1\,0\,1\,0\,1\,0\,1\,0\,1\,0\,1\,0\\ 1\,1\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\\ 1\,1\,0\,0\,0\,0\,0\,0\,1\,1\,0\,0\,0\,0\,0\,0\\ 1\,1\,0\,0\,1\,1\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\\ 1\,1\,0\,0\,1\,1\,0\,0\,1\,1\,0\,0\,1\,1\,0\,0\\ 1\,1\,1\,1\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\\ 1\,1\,1\,1\,0\,0\,0\,0\,1\,1\,1\,1\,0\,0\,0\,0\\ 1\,1\,1\,1\,1\,1\,1\,1\,0\,0\,0\,0\,0\,0\,0\,0\\ 1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1\end{pmatrix}\begin{pmatrix}1 & 0 & 0\\ 1 & 1 & 1\\ 1 & \alpha & \alpha^3\\ 1 & 1+\alpha & (1+\alpha)^3\\ 1 & \alpha^2 & 1+\alpha+\alpha^2+\alpha^3\\ 1 & 1+\alpha^2 & 1+\alpha\\ 1 & \alpha(1+\alpha) & 1+\alpha^2\\ 1 & 1+\alpha+\alpha^2 & 1+\alpha+\alpha^2+\alpha^3\\ 1 & \alpha^3 & 1+\alpha^2\\ 1 & 1+\alpha^3 & 1+\alpha\\ 1 & \alpha(1+\alpha^2) & 1\\ 1 & 1+\alpha+\alpha^3 & 1\\ 1 & \alpha^2(1+\alpha) & 1+\alpha\\ 1 & 1+\alpha^2+\alpha^3 & \alpha^3\\ 1 & \alpha(1+\alpha+\alpha^2) & 1+\alpha^2\\ 1 & 1+\alpha+\alpha^2+\alpha^3 & \alpha^3\end{pmatrix}=0.$$

Example 1. *Consider a* $(16,7,6)$ *EBCH code generated by*

$$G=\left(\begin{array}{c|c}1\,0\,0\,1\,0\,1\,1\,0 & 1\,0\,0\,1\,0\,1\,1\,0\\ 0\,1\,0\,1\,0\,1\,0\,1 & 0\,1\,0\,1\,0\,1\,0\,1\\ 0\,0\,1\,1\,0\,0\,1\,1 & 0\,0\,1\,1\,0\,0\,1\,1\\ 0\,0\,0\,0\,1\,1\,1\,1 & 0\,0\,0\,0\,1\,1\,1\,1\\ \hline 0\,0\,1\,0\,1\,0\,1\,1 & 0\,0\,0\,1\,1\,0\,0\,0\\ 1\,0\,0\,0\,0\,0\,1\,0 & 1\,1\,0\,1\,1\,0\,0\,0\\ 1\,1\,1\,1\,1\,1\,1\,1 & 0\,0\,0\,0\,0\,0\,0\,0\end{array}\right).$$

Its GPD is given by $G_1=\begin{pmatrix}0\,0\,1\,1\,0\,0\,1\,1\\ 0\,1\,0\,1\,1\,0\,1\,0\\ 1\,1\,1\,1\,1\,1\,1\,1\end{pmatrix}, G_2=\begin{pmatrix}1\,0\,0\,1\,0\,1\,1\,0\\ 0\,1\,0\,1\,0\,1\,0\,1\\ 0\,0\,1\,1\,0\,0\,1\,1\\ 0\,0\,0\,0\,1\,1\,1\,1\end{pmatrix}, G_3=\begin{pmatrix}0\,0\,0\,1\,1\,0\,0\,0\\ 1\,1\,0\,1\,1\,0\,0\,0\end{pmatrix}.$

GPD enables one to perform hard-decision decoding of code $\mathcal{C}$ as follows. Let $\mathcal{C}_i$ be the code generated by G_i. Consider a noisy codeword $(y'|y'')=(c'|c'')+(e'|e'')$, where $e=(e'|e'')$ is an error vector. Compute $z=y'+y''=(c'+c'')+(e'+e'')$. One can decode z in $\mathcal{C}_1$ to identify information vector u' and codeword $c'+c''=u'G_1$. If this step is completed successfully, one can compute $\tilde{y}'=y'-u'(G_1+\tilde{I}G_3)$ and $\tilde{y}''=y''-u'\tilde{I}G_3$, and try to decode these vectors in $\mathcal{C}_2$. This algorithm can be easily tailored to implement soft-decision decoding.

One can see from (8) that $\mathcal{C}_2$ has minimum distance $d_2\geq d/2$. However, d_1 can be very low. Hence, the above described algorithm may fail to correct even $t\leq\lfloor(d-1)/2\rfloor$ errors. A workaround for this problem is to employ list decoding for $\mathcal{C}_1$ to identify a number of possible vectors u', for each of them decode the corresponding vectors $\tilde{y}',\tilde{y}''$ in $\mathcal{C}_2$, and select the codeword $(c'|c'')$ closest to the received sequence.

GPD may be also applied recursively. This results in codes of length 1 and dimension at most 1, as discussed below.

IV. Dynamic Frozen Symbols

A. Representation of a linear code for SC decoding

Consider an $(n=l^m,k,d)$ code $\mathcal{C}$ over $\mathbb{F}_q$ with check matrix H. Let $A=B_{l,m}F_l^{\otimes m}$ be a matrix of an $n\times n$ polarizing transformation. Since A is invertible, any vector of length n can be obtained as an output $c_0^{n-1}=u_0^{n-1}A$ of the polarizing transformation. Let us investigate the constraints which need to be imposed on u_0^{n-1}, so that the output of the polarizing transformation is a codeword of $\mathcal{C}$.

These constraints are given by the equation $u_0^{n-1}AH^T=0$. By applying Gaussian elimination, one can construct the *constraint matrix* $V=QHA^T$, where Q is an invertible matrix, such that all rows of V end in distinct columns, i.e. the values $j_i=\max\{t|V_{i,t}\neq 0\}, 0\leq i<n-k$ are distinct. It can be assumed without loss of generality that $V_{i,j_i}=-1$. Let $\mathcal{F}=\{j_i|0\leq i<n-k\}$. Then one obtains

$$u_{j_i}=\sum_{s=0}^{j_i-1}u_sV_{i,s}, 0\leq i<n-k. \tag{10}$$

These equations can be considered as a generalization of the concept of frozen symbols, i.e. constraints of the form $u_{j_i}=0, j_i\in\mathcal{F}$, used in the construction of polar codes. Observe that symbols $u_{j_i}, j_i\in\mathcal{F}$ can take arbitrary values, which, however, depend on the values of some other symbols with smaller indices. Therefore, symbols u_{j_i} given by (10) will be referred to as dynamic frozen symbols.

Example 2. *Consider* $(16,7,6)$ *EBCH code* $\mathcal{C}$ *over* $\mathbb{F}_2$. *The generator polynomial of the corresponding non-extended code has roots* α,α^3 *and their conjugates, where* α *is a primitive root of* x^4+x^3+1. *The constraints on vector* u_0^{15}, *such that* $u_0^{15}A\in\mathcal{C}$, *are given by the equation at the top of this page. Multiplying matrices, expanding their elements in the standard basis and applying elementary linear operations, one obtains*

$$u_0^{15}\begin{pmatrix}0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,0\,1\,0\,0\,0\\ 0\,0\,0\,1\,0\,1\,0\,0\,0\,0\,1\,0\,0\,0\,0\,0\\ 0\,0\,0\,0\,0\,1\,0\,0\,0\,1\,0\,0\,0\,0\,0\,0\\ 0\,0\,0\,0\,0\,0\,0\,0\,1\,0\,0\,0\,0\,0\,0\,0\\ 0\,0\,0\,1\,0\,0\,1\,0\,0\,0\,0\,0\,0\,0\,0\,0\\ 0\,0\,0\,0\,1\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\\ 0\,0\,1\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\\ 0\,1\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\\ 1\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\end{pmatrix}^T=0$$

This means that $u_0 = u_1 = u_2 = u_4 = u_8 = 0$*, and* $u_6 = u_3$*,* $u_9 = u_5$*,* $u_{10} = u_3 + u_5$*,* $u_{12} = u_{10}$*. Symbols* $u_3, u_5, u_7, u_{11}, u_{13}, u_{14}, u_{15}$ *are non-frozen.*

Observe that any linear code of length l^m can be represented by a system of equations (10). This enables one to employ the SC decoding algorithm and its variations for decoding of arbitrary linear codes of length l^m. That is, one can successively make decisions

$$\widehat{u}_i = \begin{cases} \arg\max_{u_i \in \mathbb{F}_q} W_n^{(i)}(u_0^i | y_0^{n-1}), & i \notin \mathcal{F} \\ \sum_{s=0}^{i-1} u_s V_{t_i,s}, & \text{otherwise,} \end{cases} \tag{11}$$

where t_i is an integer, such that $j_{t_i} = i$. Observe that if u_0^{i-1} are the correct values of the input symbols of the polarizing transformation, the probability of symbol error P_i in this case remains the same as in the case of classical polar codes. Hence, the error probability of the considered code under SC decoding can be still computed via (7).

The set $\mathcal{F}$ of dynamic frozen symbol indices for a generic linear code is not guaranteed to contain all symbols with high error probability. Hence, for most linear codes the SC decoding error probability, given by (7), far exceeds the error probability of other decoding algorithms. Substantially better performance can be obtained by employing list or stack SC decoding techniques. However, the list size (i.e. the decoding complexity) needed to obtain near-ML performance, in general, increases exponentially with code dimension.

The complexity of computing $W_n^{(i)}(u_0^i | y_0^{n-1})$ is exactly the same as in the case of classical polar codes, i.e. $O(n \log n)$. However, evaluation of the expression (10) may increase the decoding complexity to $O(n^2)$.

B. Extended BCH codes and Arikan kernel

Let us investigate in more details the structure of the set of dynamic frozen symbol indices of binary extended primitive narrow-sense BCH (EBCH) codes for the case of Arikan polarizing transformation. Observe that in this case construction of the system of equations (10) can be viewed as a recursive application of the GPD to the considered code.

It was shown in [26]–[28] that a punctured RM code of order r and length 2^m is equivalent to a cyclic code with generator polynomial $g(x)$ having roots $\alpha^i : 1 \leq \mathrm{wt}(i) < m - r, 1 \leq i \leq 2^m - 2$, where α is a primitive element of $\mathbb{F}_{2^m}$, and $\mathrm{wt}(i)$ is the number of non-zero digits in the binary expansion of integer i. Furthermore, it was shown in [29] that an EBCH code $\mathcal{C}'$ of length 2^m with design distance $d \geq \delta(r, m) + 3$ is a subcode of the RM code of order $m - r - 1$, where

$$\delta(r, m) = \max_{i:\mathrm{wt}(i)=r} \min\left\{ i 2^j \bmod (2^m - 1) | 0 \leq j < m \right\}.$$

A recursive expression for $\delta(r, m)$ is derived in [29]. One can consider a RM code of order $m - r - 1$ as a polar code with the set of frozen symbol indices $\mathcal{F}'' = \{i | \mathrm{wt}(i) \leq r\}$. Hence, the set of dynamic frozen symbol indices $\mathcal{F}'$ for the EBCH code includes $\mathcal{F}''$. It can be seen from (5)–(6) that $Z_{n,i} = O(Z_{1,0}^{2^{\mathrm{wt}(i)}})$. Hence, the set of frozen symbols for EBCH codes includes all those ones, such that their Bhattacharyya parameters (and error probability $P_i \leq \frac{1}{2} Z_{n,i}$) decrease slowly while decreasing the Bhattacharyya parameter $Z_{1,0}$ of the original channel. Most of these symbols have high error probability P_i.

The above statement is true only if one employs standard digit ordering. That is, each coordinate c_i of a codeword $(c_0, \ldots, c_{n-1})$ can be associated with some $x_i \in \mathbb{F}_{2^m}$, so that all x_i are distinct, and all codewords satisfy check equations

$$\sum_{i=0}^{n-1} c_i x_i^j = 0, 0 \leq j < d - 1. \tag{12}$$

The standard digit ordering is given by $x_i = \sum_{j=0}^{m-1} X_{i,j} \beta_j$, where $i = \sum_{j=0}^{m-1} X_{i,j} 2^j, X_{i,j} \in \{0, 1\}$, and $\beta_0, \ldots, \beta_{m-1}$ is some basis of $\mathbb{F}_{2^m}$. In what follows, more detailed characterization of the set of dynamic frozen symbols for EBCH codes will be derived.

Let

$$C_t = \left\{ t 2^i \bmod 2^{m-1} | 0 \leq i < m_t, t 2^{m_t} \equiv t \bmod 2^m - 1 \right\}$$

be a cyclotomic coset generated by t. Let $\mathcal{Q}$ be the set of minimal cyclotomic coset representatives. It can be seen that all elements of a cyclotomic coset have the same weight. Therefore

$$\sum_{\substack{s \in \mathcal{Q} \\ \mathrm{wt}(s) = r}} m_s = \binom{m}{r}.$$

Theorem 2. *Consider a* $(2^m, k, d)$ *extended primitive narrow-sense BCH code over* $\mathbb{F}_2$*. Let* $S = \{i \in \mathcal{Q} | 0 \leq i < d - 1\}$*. Let* N_t *be the number of dynamic frozen symbols* u_i *for this code, such that* $\mathrm{wt}(i) = t$*. Then* $N_t = \sum_{s \in S_t} m_s$*, where* $S_t = \{s \in S | \mathrm{wt}(s) = t\}$*, and* m_s *is the size of the cyclotomic coset generated by* s*.*

Proof: Consider parity check equation (12). Let $x_i = \sum_{s=0}^{m-1} X_{i,s} \beta_s, X_{i,s} \in \{0, 1\}$. Then

$$x_i^j = \left(\sum_{s=0}^{m-1} X_{i,s} \beta_s \right)^j = \left(\sum_{s=0}^{m-1} X_{i,s} \beta_s \right)^{\sum_{t=0}^{m-1} j_t 2^t}$$

$$= \prod_{t=0}^{m-1} \left(\sum_{s=0}^{m-1} X_{i,s} \beta_s^{2^t} \right)^{j_t} = \sum_{\substack{\mathrm{wt}(e_0^{m-1}) \leq \mathrm{wt}(j) \\ e_s \in \{0,1\}}} v_{j;e_0^{m-1}} \prod_{s=0}^{m-1} X_{is}^{e_s},$$

where $v_{j;e_0^{m-1}} \in \mathbb{F}_{2^{m_j}}$ are some coefficients. Hence, any codeword c_0^{n-1} satisfies

$$0 = \sum_{\substack{\mathrm{wt}(e_0^{m-1}) \leq \mathrm{wt}(j) \\ e_s \in \{0,1\}}} v_{j;e_0^{m-1}} \sum_{i=0}^{n-1} c_i \prod_{s=0}^{m-1} X_{is}^{e_s}, j \in \tilde{S},$$

where $\tilde{S} = \left\{ j 2^l | j \in S, 0 \leq l < m_j, j 2^{m_j} \equiv j \bmod 2^m - 1 \right\}$.

It can be seen that the i-th row of $A = B_{2,m} F_2^{\otimes m}$ is a sequence of values of various monomials $X^{(a)} = \prod_{s=0}^{m-1} X_s^{a_{m-1-s}}, a_s \in \{0, 1\}$ at point $(X_{i,0}, \ldots, X_{i,m-1}) \in$

$\mathbb{F}_2^m$. Hence, $u_{e'} = \sum_{i=0}^{n-1} c_i \prod_{s=0}^{m-1} X_{i,s}^{e_s}$ is the value of the e'-th element of the input vector of the polarizing transformation, where $e' = \sum_{s=0}^{m-1} e_s 2^{m-1-s}$, $e_s \in \{0,1\}$, and $u_0^{n-1} = c_0^{n-1} A$, so that

$$0 = \sum_{\mathrm{wt}(e_0^{m-1}) \leq \mathrm{wt}(j)} v_{j;e_0^{m-1}} u_{e'}, j \in \tilde{S}$$

Any such equation gives rise to m_j equations with coefficients in $\mathbb{F}_2$. Observe that there are $\widetilde{M}_t = \sum_{i=t}^{\rho} \sum_{s \in S_i} m_s$ equations, which involve symbols $u_{e'} : t \leq \mathrm{wt}(e') \leq \rho$, where $\rho = \max_{0 \leq j < d-1} \mathrm{wt}(j)$. Hence, the number $\widehat{M}_t = \sum_{i=t}^{\rho} N_t$ of dynamic frozen symbols $u_i : t \leq \mathrm{wt}(i) \leq \rho$, is upper bounded by $\widetilde{M}_t$. It can be also seen that $\widehat{M}_0 = \widetilde{M}_0$.

The equality $N_0 = 1 = m_0$ holds for any EBCH code with $d \geq 2$. Assume that $N_t = \sum_{s \in S_t} m_s$ for all $t < t_0$, so that $\widehat{M}_{t_0} = \widetilde{M}_{t_0}$. Since $\widehat{M}_{t_0+1} = \widehat{M}_{t_0} - N_{t_0} \leq \widetilde{M}_{t_0+1} = \widetilde{M}_{t_0} - \sum_{s \in S_{t_0}} m_s$, one obtains $N_{t_0} \geq \sum_{s \in S_{t_0}} m_s$. Assume that this inequality is strict.

Any codeword of the considered EBCH code can be represented as a vector of values of polynomial

$$f(x) = \sum_{t \in \mathcal{Q} \setminus S} \mathrm{Tr}_{m_t}(\gamma_t x^{n-1-t}) \tag{13}$$

in distinct points $x_i \in \mathbb{F}_{2^m}$, where $\gamma_i \in \mathbb{F}_{2^{m_i}}$, $\mathrm{Tr}_m(x) = \sum_{i=0}^{m-1} x^{2^i}$. This polynomial can be represented in multivariate form as

$$f(X_0, \ldots, X_{m-1}) = \sum_{t \in \mathcal{Q} \setminus S} \mathrm{Tr}_{m_t} \left(\gamma_t \prod_{j=0}^{m-1} \left(\sum_{l=0}^{m-1} \beta_l^{2^j} X_l \right)^{1-t_j} \right)$$
$$= \sum_{e_0, \ldots, e_{m-1} \in \{0,1\}} u_{e'} \prod_{l=0}^{m-1} X_l^{1-e_l} \tag{14}$$

where $t = \sum_{j=0}^{m-1} t_j 2^j, t_j \in \{0,1\}$, so that $c_i = f(x_i) = f(X_{i,0}, \ldots, X_{i,m-1})$. Observe that the e'-th row of matrix A can be considered as a table of values of $\prod_{l=0}^{m-1} X_l^{1-e_l}$ in various points of $\mathbb{F}_2^m$. Hence, $u_{e'}$ can be considered as input symbols of the polarizing transformation.

Hence, the set of polynomials $f(X_0, \ldots, X_{m-1})$ corresponding to the considered code contains $K_{t_0} = \binom{m}{t_0} - \sum_{s \in S_{t_0}} m_s$ linearly independent polynomials given by (14) of degree $m - t_0$. Observe also that the forms of degree $m - t_0$ of these polynomials are also linearly independent. However, this is not possible since, by assumption, there are $N_t > \sum_{s \in S_{t_0}} m_s$ constraints on the coefficients of these forms. The obtained contradiction proves the theorem. ■

The particular set of dynamic frozen symbol indices $\mathcal{F}'$ of the EBCH code depends on the basis being used. One may enumerate different bases of $\mathbb{F}_{2^m}$ and select the one which minimizes the SC decoding error probability (7). Similar approach was used in [30], [31] to obtain trellis diagrams of EBCH codes.

Theorem 2 and the existence of a RM supercode suggest that the SC algorithm and its variations may work for EBCH codes.

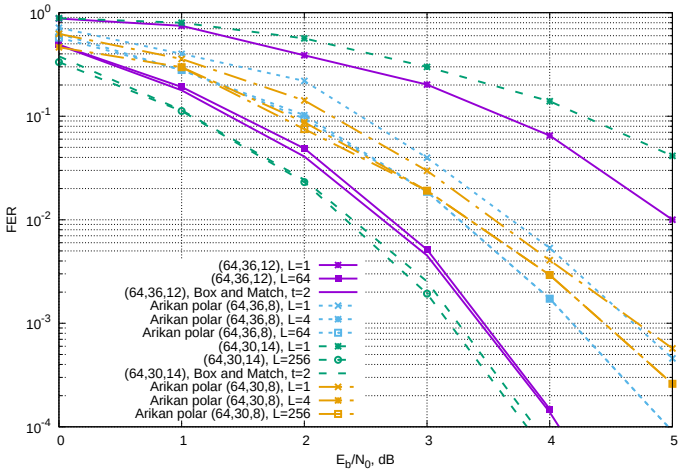


Fig. 4. Performance of list/stack SC decoding of extended BCH codes

Unfortunately, experiments show that this is true only for small n. Figure 4 illustrates the performance of EBCH codes under list/stack SC algorithm with list size L and box-and-match [32] algorithm with reprocessing order t, as well as Arikan polar codes. It can be seen that Arikan polar codes far outperform extended BCH codes in the case of list size L equal to 1 (i.e. classical SC decoding). However, higher minimum distance results in significant performance gain of EBCH codes under box-and-match near-ML decoding algorithm. Huge list size L is needed in order to obtain comparable performance under list/stack SC decoding, while Arikan polar codes achieve the near-ML performance already for $L = 4$.

V. Polar Subcodes

It is possible to show that the minimum distance of polar codes with Arikan kernel is given by $O(\sqrt{n})$ [33]. This results in quite poor ML decoding performance.

Exact performance analysis of the list/stack SC decoding algorithm, which is commonly used to implement near-ML decoding of polar codes, still remains an open problem. It was empirically observed that in the low-SNR region codes with lower SC decoding error probability provide lower error probability under list SC decoding. However, in the high-SNR region the performance of list/stack SC decoding algorithm depends mainly on code minimum distance. Therefore, we propose to explicitly construct codes with a given minimum distance, which would minimize the SC decoding error probability.

Definition 1. *Consider a q-ary input memoryless output symmetric channel $W(y|c)$ and an $(n = l^m, k', d)$ code $\mathcal{C}'$ over $\mathbb{F}_q$, called parent code. Let $\mathcal{F}'$ be the set of dynamic frozen symbol indices of $\mathcal{C}'$ for the case of kernel F_l. An $(n, k \leq k', \geq d)$ polar subcode $\mathcal{C}$ of code $\mathcal{C}'$ is defined as the set of vectors $c_0^{n-1} = u_0^{n-1} B_{l,m} F_l^{\otimes m}$, where u_0^{n-1} simultaneously satisfies the dynamic freezing equations* (10) *for code $\mathcal{C}'$, and additional constraints $u_s = 0$ for $k' - k$ indices $s \notin \mathcal{F}'$ with the highest error probabilities P_s for a given channel $W(y|c)$.*

Encoding of polar subcodes can be performed as

$$c = xWA, \tag{15}$$

where x is an information vector, W is a matrix, such that $WV^T = 0$, and V is the dynamic freezing constraint matrix. This can be considered as pre-coding the data with some outer code with generator matrix W, and encoding its codeword with a polar code. However, we do not explicitly specify an outer code for this construction. Instead, we require that the obtained codeword c should belong to the parent code with sufficiently high minimum distance.

Polar codes with CRC [2] and LDPC outer codes [8] can be considered as a special case of the proposed construction. However, these codes employ ad-hoc constraints (10). Therefore, it is difficult to control their minimum distance.

It must be recognized that the SC decoding error probability P given by (7) of a polar subcode cannot be less than the SC decoding error probability of a classical polar code of the same length and dimension, constructed for the same channel using the same kernel F_l. Therefore, polar subcodes provide no advantage with respect to classical polar codes if SC decoding is used. However, significant performance gain under list/stack SC decoding can be obtained. Experiments show that for given values of n, k, d polar subcodes with lower P provide lower list/stack SC decoding error probability. Hence, one should select $\mathcal{C}'$ so that its set $\mathcal{F}'$ includes as many as possible indices j_i corresponding to symbols with high error probability P_{j_i}.

A. Arikan kernel

1) The construction: We propose to employ EBCH codes as parent ones in the proposed construction of polar subcodes. Theorem 2 implies that the indices of the most of the frozen symbols of EBCH codes have low weight. Bounds (5)–(6) imply that the Bhattacharyya parameter of the i-th bit subchannel is given by $Z_{n,i} = O(Z_{1,0}^{2^{\mathrm{wt}(i)}})$. Hence, employing EBCH codes as parent ones in the proposed construction enables one to avoid freezing of bit subchannels with low $Z_{n,i}$. This improves the performance of the obtained code under SC decoding and its variations.

Observe that increasing minimum distance of the parent code causes more bit subchannels with low $Z_{n,i}$ to be frozen. In order to keep code dimension k fixed, one needs to unfreeze some bits subchannels with high $Z_{n,i}$. This results in higher SC decoding error probability. This can be compensated to a certain extent by employing list SC decoding and its variations with larger list size. Unfortunately, there are still no analytical techniques for finding a trade-off between the performance and decoding complexity. We have to use simulations in order to find optimal values of the code minimum distance.

Example 3. *Let us construct a* $(16, 6, 6)$ *polar subcode of* $(16, 7, 6)$ *EBCH code considered in Example 2, by optimizing it for the case of the binary erasure channel with erasure probability* $Z_{1,0} = 0.5$*. The vector of bit subchannel Bhattacharyya parameters (i.e. symbol erasure probabilities) equals* $Z_{16} = (\underline{0.999}, \underline{0.992}, \underline{0.985}, 0.77, \underline{0.96}, \underline{0.65}, 0.53, 0.1, \underline{0.9}, \underline{0.47}, \underline{0.35}, 3.7{\cdot}10^{-2}, \underline{0.23}, 1.5{\cdot}10^{-2}, 7.8{\cdot}10^{-3}, 1.5{\cdot}10^{-5})$*. Here the values corresponding to dynamic frozen symbols of the EBCH code are underlined. It can be seen that* u_3 *has the highest erasure probability* 0.77 *among not yet frozen symbols. Therefore, we propose to introduce an additional constraint* $u_3 = 0$*. This is equivalent to removing the first row from matrices* G_1 *and* G_3 *presented in Example 1.*

Example 4. *Consider construction of a* $(1024, 512)$ *code. There exists a* $(1024, 513, 116)$ *EBCH code, which cannot be decoded efficiently with (list) SC decoder. On the other hand, the classical polar code optimized for AWGN channel with* $E_b/N_0 = 2dB$ *has minimum distance 16. One can take a* $(1024, 893, 28)$ *EBCH parent code* $\mathcal{C}' : RM(5, 10) \subset \mathcal{C}' \subset RM(8, 10)$ *and freeze* 381 *additional bit subchannels to obtain a* $(1024, 512, \geq 28)$ *polar subcode with dynamic frozen symbols. The specification of the obtained code includes only* $f = 20$ *non-trivial equations* (10) *with* $T = 111$ *terms, so the cost of evaluation of dynamic frozen symbols is negligible compared to the cost of multiplication by matrix* A*.*

Observe that the SC decoding error probability P of a (n, k, d) polar subcode of any code cannot be less than the SC decoding error probability $\overline{P}$ for a classical (n, k) polar code constructed for the same channel using the same kernel. However, the performance of a polar subcode under list/stack SC decoding with sufficiently large list size L may be substantially better. It was empirically observed that the size of the list L needed to obtain such gain increases with P. Hence, one needs to quantify the value of $\overline{P}/P$ needed to obtain a given minimum distance d. However, it would be easier to characterize the rate of a polar subcode with a given minimum distance, such that it has the same SC decoding error probability as a given classical polar code of the same length.

Let $\overline{\mathcal{C}}$ be a polar code with kernel F_2 of rate $\overline{\rho}(z)$, such that all symbols with $Z_{n,i} < z$ are not frozen. Consider now an (n, k, d) polar subcode $\mathcal{C}$ of rate $\rho(z, d) = k/n$, obtained from a $(n = 2^m, k' = \beta(m, d)n, d)$ EBCH code $\mathcal{C}'$ by freezing all symbols u_i with $Z_{n,i} \geq z$. The set of non-frozen symbols of code $\mathcal{C}$ can be represented as $\Delta(d, z) = \cup_{r=0}^{m-1} \left(\Delta(r, m, z) \setminus \mathcal{F}'_{d,r}\right)$, where $\Delta(r, m, z) = \{i | 0 \leq i < 2^m, \mathrm{wt}(i) = r, Z_{2^m,i} < z\}$, and $\mathcal{F}'_{d,r}$ is the set of dynamic frozen symbol indices u_e of $\mathcal{C}'$, such that $\mathrm{wt}(e) = r$.

It is quite difficult to find $|\Delta(r, m, z)|$ analytically, although it can be computed in polynomial time for any specific binary input output symmetric memoryless channel and values r, m [25]. Therefore, we propose to approximate it by employing an asymptotic expression for the fraction of common non-frozen symbols of a RM code of rate $R(m-r, m) = 2^{-m} \sum_{j=0}^{m-r} \binom{m}{j}$ and a polar code of length $n = 2^m$ and rate ρ. This value was shown in [34] to converge with $m \to \infty$ to $\phi(\rho, r, m) = C \min\left(\frac{\rho}{C}, R(m-r, m)\right)$, where C is the capacity of the considered channel. Hence, the $2^{-m}|\Delta(r, m, z)| \approx \phi(\overline{\rho}(z), r, m) - \phi(\overline{\rho}(z), r+1, m)$.

Therefore, one obtains

$$|\Delta(r, m, z) \setminus \mathcal{F}'_{d,r}| \geq \max(0, |\Delta(r, m, z)| - |\mathcal{F}'_{d,r}|), \tag{16}$$

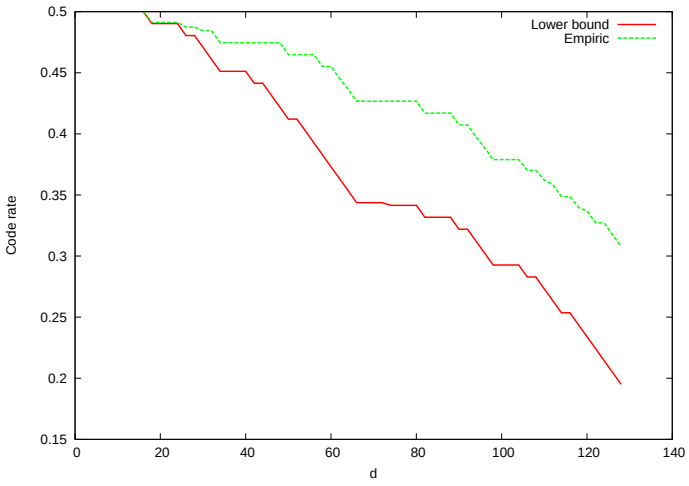


Fig. 5. Rate of polar subcodes of length 1024

so that

$$\rho(z,d) \geq 2^{-m} \sum_{r=0}^{m} \max(0, |\Delta(r,m,z)| - |\mathcal{F}'_{d,r}|)$$

$$\approx \sum_{r=0}^{m} \max\left(0, \phi(\overline{\rho}(z), r, m) - \phi(\overline{\rho}(z), r+1, m) - N_r 2^{-m}\right),$$

where N_r is given by Theorem 2.

Figure 5 illustrates this bound together with the actual rate of (n,k,d) polar subcodes of EBCH codes of length $n = 1024$. The dimension k of these subcodes was selected so that they achieve approximately the same successive cancellation decoding error probability at $E_s/N_0 = -1$ dB as the classical Arikan polar code $(1024, 512, 16)$ constructed for the same value of E_s/N_0. It can be seen that the bound is quite loose. This is both due to (16), which assumes that all dynamic frozen symbols induced by the EBCH code correspond to subchannels with the lowest possible Bhattacharyya parameters, and application of an asymptotic expression for approximation of $|\Delta(r,m,z)|$.

It can be seen that the degradation of polar subcode rate with respect to a classical polar code is negligible for d up to twice the minimum distance of the original polar code.

2) Encoding and Decoding Complexity: Encoding of the proposed polar subcodes of binary EBCH codes can be performed via (15) with complexity $C_W + \frac{1}{2} n \log n$, where $C_W = T - f$ is the cost of multiplication by matrix W, T is the number of terms in the right-hand side of non-trivial equations (10), and $f \leq \sum_t N_t$ is the number of such equations. Systematic encoding can be implemented using the approach introduced in [35] with complexity $C_W + n \log n$.

Theorem 2 implies that the set of dynamic frozen symbol indices for a parent EBCH code includes only the ones with sufficiently small weight. Furthermore, one can see that a dynamic freezing equation for symbol u_i cannot involve symbols $u_j : \mathrm{wt}(j) > \mathrm{wt}(i)$. On the other hand, most of the constraints $u_s = 0$, imposed on bit subchannels with high error probability P_s, correspond to low-weight integers s. This causes f to be much less than the value $\sum_t N_t$ predicted by Theorem 2 for the parent code, so that matrix W appears to be sparse. This was illustrated in Example 4. Hence, the encoding complexity of the proposed polar subcodes does not exceed that of polar codes with f-bit CRC.

Decoding of polar subcodes can be performed using the same algorithms as classical polar codes, which should be augmented with a subroutine for evaluation of dynamic frozen symbols. Hence, the number of operations with probabilities or log-likelihood ratios remains the same as in the case of classical polar codes. However, the cost of bit manipulations increases at least by $O(C_W)$. For example, in the case of Tal-Vardy list decoding algorithm and its derivatives, the values u_j, which are needed for evaluation of the dynamic frozen symbols, are not stored explicitly. One should either introduce for each path an additional array of size f, where the values of dynamic frozen symbols are accumulated, or recover u_j from intermediate values. In the first case the decoding complexity increases by $fLn + C_W L$ bit operations, where L is the list size, since the additional arrays need to be copied while cloning the paths. In the second case the complexity depends on the specific structure of dynamic freezing constraints.

The sequential decoding algorithm [5] and its block generalization [36] were shown to be able to decode polar codes with very low average complexity and good performance. These algorithms can be naturally used in the case of polar subcodes.

B. Improved polar subcodes with Arikan kernel

Let us consider a $(n = 2^m, k, d)$ polar subcode constructed as described in Section V-A. It can be represented as an IGCC with outer codes of length $2^s, s < m$. It appears that most outer codes obtained in this way are classical Arikan polar codes with quite low minimum distance and high decoding error probability. Therefore we propose to employ the approach suggested in [37]. Namely, we impose the requirement on outer codes to be $(2^s, k_i, d_i)$ EBCH codes (or their subcodes). The parameters k_i, d_i are selected in order to minimize the MSD error probability, which is given by

$$P = 1 - \prod_{i=0}^{2^{m-s}-1} (1 - \pi_i)$$

under the constraint $\sum_{i=0}^{2^{m-s}-1} k_i = k$. Here π_i denotes the decoding error probability of the code utilized at the i-th level of the IGCC. These probabilities can be estimated, for example, using the tangential sphere bound [38] together with density evolution [25] or Gaussian approximation [9].

The obtained IGCC can be also represented via a system of equations (10). The corresponding matrix V is given by

$$V = \begin{pmatrix} & & V' & \\ V_0 & 0 & \dots & 0 \\ 0 & V_1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & V_{2^{m-s}-1} \end{pmatrix},$$

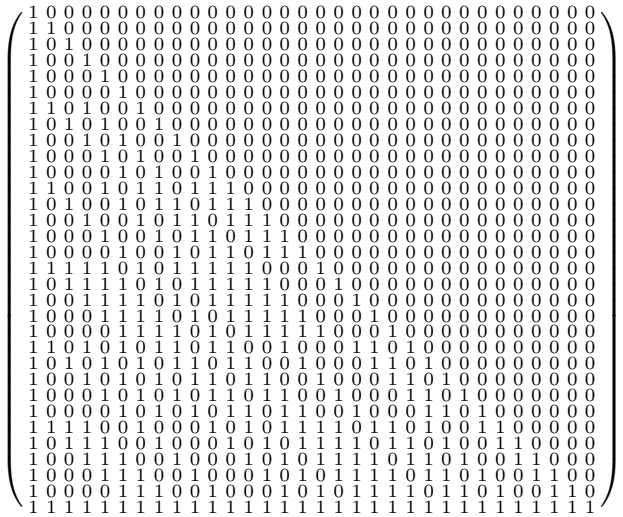


Fig. 6. Extended BCH kernel F_{32}

where V' is a constraint matrix for a parent $(2^m, K \geq k, d)$ EBCH code, and V_i are constraint matrices for outer $(2^s, k_i, d_i)$ codes.

The codes obtained in this way are supposed to be decoded by the block sequential decoding algorithm [36] with block size at least 2^s. This algorithm employs the fast tree-trellis list Viterbi algorithm [39] for decoding of outer codes of the IGCC. However, more efficient decoding techniques can be designed for specific outer codes.

The proposed approach can be considered as a generalization of the construction suggested in [40]. The Mondelli-Hassani-Urbanke codes can be considered as GCC with inner Arikan codes and outer RM or polar codes. Since EBCH codes provide higher minimum distance, one may expect the improved polar subcodes to provide better performance.

C. Polar subcodes with EBCH kernel

For $l = 2^\mu$ an extended BCH kernel can be obtained as matrix F_l, where $((F_l)_{i+1,1}, \ldots, (F_l)_{i+1,l-1})$ is a vector of coefficients of $x^j g_{i'}(x)$, where $g_{i'}(x)$ is a generator polynomial of a $(l-1, l-1-i')$ BCH code, and j is the smallest non-negative integer, such that $i = j + i'$. Furthermore, one has $(F_l)_{0,0} = 1$ and $(F_l)_{i+1,0} = \sum_{j=1}^{l-1} (F_l)_{i+1,j}$. Figure 6 presents an example of the EBCH kernel.

Let $C(q, s, m, J)$ be a code over $\mathbb{F}_q$ of length q^{sm}, which consists of evaluation vectors of m-variate polynomials $\frac{1}{(q^s-1)^m} A(x_0, \ldots, x_{m-1})$, such that all their coefficients $A_{j_0,\ldots,j_{m-1}}$ are equal to zero for $(j_0, \ldots, j_{m-1}) \in J$, in various points $(x_0, \ldots, x_{m-1}) \in \mathbb{F}_{q^s}^m$. Let $L(h) = \{(j_0, \ldots, j_{m-1}) | 0 \leq j_i < q^s, \sum_i j_i > h\}$, and $L^*(h) = \left\{ j = \sum_{i=0}^{m-1} j_i q^{si} | (j_0, \ldots, j_{m-1}) \in L(h) \right\}$. It was shown in [41] that codes $C(q, s, m, L(h))$ and $C(q, sm, 1, L^*(h))$ are equivalent. That is, these codes can be considered as GCC with inner and outer extended cyclic codes. Furthermore, outer codes in this construction can be recursively decomposed in the same way. This implies that an EBCH code has a supercode, which is equivalent to a GCC with inner and outer extended cyclic codes. This is similar to the fact of existence of a RM supercode for any EBCH code, which was used above to show that EBCH codes have "not so bad" set of dynamic frozen symbol indices in the case of Arikan polarizing transformation.

Therefore, we propose the following code construction. Let $\mathcal{C}'$ be an $(2^{\mu m}, k', d)$ EBCH code, such that its t-th locator is $x_t = \sum_{j=0}^{m-1} \beta_{t_j} \gamma_j$, where $(\gamma_0, \ldots, \gamma_{m-1})$ is a basis of $\mathbb{F}_{2^{\mu m}}$ considered as a vector space over $\mathbb{F}_{2^\mu}$, $t = \sum_{j=0}^{m-1} t_j 2^{\mu j}, 0 \leq t_j < 2^\mu$, and β_i is the i-th element of $\mathbb{F}_{2^\mu}$. The above described construction of EBCH kernel corresponds to the case $\beta_0 = 0, \beta_i = \alpha^{i-1}, 1 \leq i < 2^\mu$, where α is a primitive element of $\mathbb{F}_{2^\mu}$. Then one can construct polar subcode of $\mathcal{C}'$ using the polarizing transformation $A = B_{l,m} F_l^{\otimes m}$.

The proposed construction requires one to be able to compute symbol error probabilities P_i. To the best of our knowledge, there are still no analytical techniques for solving this problem, except for the case of the binary erasure channel [42]. Therefore, we use simulations to obtain these values.

The obtained polar subcode can be considered as an instance of the IGCC introduced in Section III. Indeed, let us consider a subset $\mathcal{F}_s = \{j_i \in \mathcal{F} | \lfloor j_i / l \rfloor = s\}$ of the set of dynamic frozen symbol indices corresponding to the s-th block, $0 \leq s < l^{m-1}$, and let $V^{(s)}$ be the corresponding $|\mathcal{F}_s| \times l^m$ submatrix of V. It can be assumed without loss of generality that $V^{(s)}$ has an identity submatrix in columns with indices in $\mathcal{F}_s$, so that $V^{(s)} = (\Delta_s | \underbrace{(\Sigma_s | I) \Pi_s}_{|\mathcal{F}_s| \times l} | 0)$, where Π_s is a $l \times l$ permutation matrix, and Δ_s, Σ_s are some matrices. Therefore, one obtains a system of equations

$$u_{sl^{m-1}}^{sl^{m-1}+l-1} ((\Sigma_s | I) \Pi_s)^T = u_0^{sl^{m-1}-1} \Delta_s^T.$$

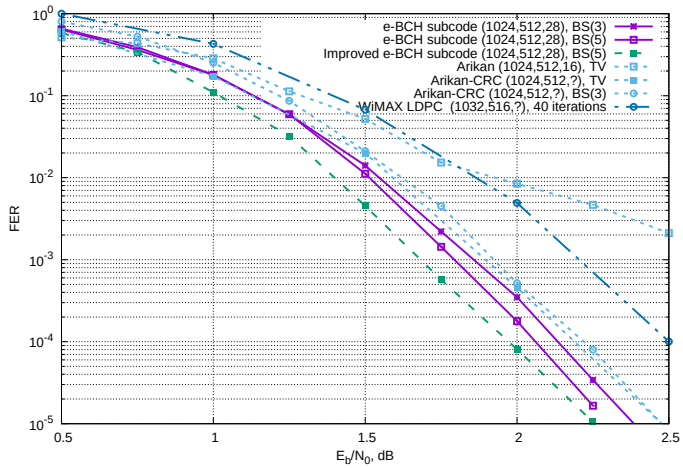
Its solution is given by

$$u_{sl^{m-1}}^{sl^{m-1}+l-1} = v(I | \Sigma_s) \Pi_s + u_0^{sl^{m-1}-1} \Delta_s^T (0 | I) \Pi_s,$$

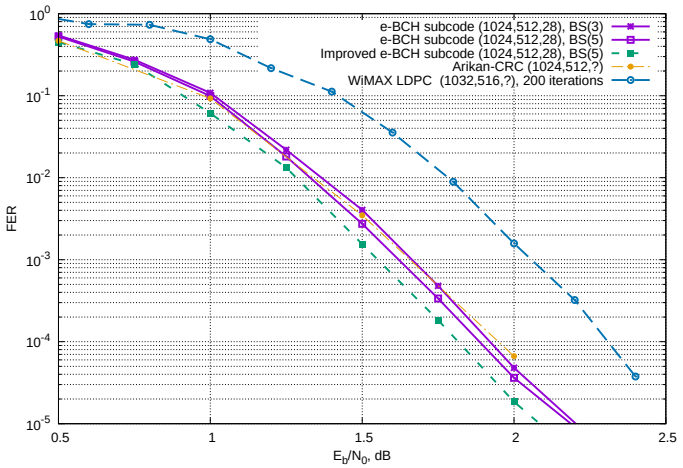
where v is an arbitrary vector in $\mathbb{F}_2^{l-|\mathcal{F}_s|}$. Hence, instead of successive decoding of symbols $u_{sl^{m-1}}, \ldots, u_{sl^{m-1}+l-1}$ according to (11), one can recover them jointly by decoding in a coset $x_s + \mathbb{C}_s$, where $\mathbb{C}_s$ is a code generated by matrix $(I | \Sigma_s) \Pi_s F_l$, and $x_s = u_0^{sl^{m-1}-1} \Delta_s^T (0 | I) \Pi_s F_l$, as shown in Figure 2. This enables one to improve the performance and/or reduce the decoding complexity.

D. Reed-Solomon kernel

The results of [41] allow us to extend the proposed construction of polar subcodes of EBCH codes to the case of Reed-Solomon (RS) kernel over $\mathbb{F}_q$. The RS kernel is given by matrix F_l, where $(F_l)_{i,j} = \beta_j^{l-1-i}$, and β_j are some distinct elements of $\mathbb{F}_q, l \leq q$. It was shown in [23] that for $l \leq q$ the Reed-Solomon kernel provides the highest possible polarization rate. However, polar codes with RS kernel still suffer from low minimum distance.



(a) $L = 32$



(b) $L = 256$

Fig. 7. Performance of polar subcodes with Arikan kernel of length 1024

In order to obtain a code with better performance, one can set $l = q$ and represent an $(n = q^m, k', d)$ EBCH code of length q^m over $\mathbb{F}_q$, such that its t-th locator is $x_t = \sum_{j=0}^{m-1} \beta_{t_j} \gamma_j$, where $(\gamma_0, \ldots, \gamma_{m-1})$ is a basis of $\mathbb{F}_{q^m}$ considered as a vector space over $\mathbb{F}_q$, $t = \sum_{j=0}^{m-1} t_j q^j, 0 \leq t_j < q^m$, via a system of equations (10), and introduce additionally $k' - k$ static freezing constraints $u_i = 0$ for non-frozen subchannels $W_n^{(i)}$ with the highest error probability. Again, simulations have to be used for performance evaluation of bit subchannels. The obtained codes can be decoded using the techniques presented in [6], [43].

Example 5. *Consider construction of a* $(16, 8, 6)$ *polar subcode over* $\mathbb{F}_4$. *The* 4×4 *Reed-Solomon kernel is given by*

$$F_4 = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 0 & 1 & \beta+1 & \beta \\ 0 & 1 & \beta & \beta+1 \\ 1 & 1 & 1 & 1 \end{pmatrix},$$

where β *is a primitive element of* $\mathbb{F}_4$. *The check matrix of the* $(16, 9, 6)$ *parent EBCH code is*

$$H = \begin{pmatrix} 1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1 \\ 0&1&\beta&\beta^2&0&1&\beta&\beta^2&0&1&\beta&\beta^2&0&1&\beta&\beta^2 \\ 0&0&0&0&1&1&1&1&\beta&\beta&\beta&\beta&\beta^2&\beta^2&\beta^2&\beta^2 \\ 0&1&\beta^2&\beta&\beta&\beta^2&1&0&1&0&\beta&\beta^2&\beta^2&\beta&0&1 \\ 0&0&0&0&1&1&1&1&\beta^2&\beta^2&\beta^2&\beta^2&\beta&\beta&\beta&\beta \\ 0&1&1&1&\beta&1&0&\beta&\beta&\beta&1&0&\beta&0&\beta&1 \\ 0&0&0&0&\beta^2&\beta^2&\beta&\beta&\beta^2&\beta&\beta^2&\beta&\beta^2&\beta&\beta&\beta^2 \end{pmatrix}$$

This corresponds to the following constraint matrix for the polarizing transformation $A = B_{4,2} F_4^{\otimes 2}$*:*

$$V = \begin{pmatrix} 0&0&0&0&0&0&1&0&0&\beta^2&0&0&1&0&0&0 \\ 0&0&0&\beta^2&0&0&1&0&0&1&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0 \\ 0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \end{pmatrix}.$$

In the case of transmission of a binary image of the output of the polarizing transformation A *over the AWGN channel with* $E_s/N_0 = -1$ *dB, the symbol error probabilities were found to be* $(\underline{0.74}, \underline{0.7}, \underline{0.55}, 0.27, \underline{0.58}, 0.33, 0.12, 0.02, \underline{0.23}, 0.04, 4 \cdot 10^{-3}, 2 \cdot 10^{-4}, \underline{0.03}, 4 \cdot 10^{-4}, 3 \cdot 10^{-6}, < 10^{-6})$. *Hence, we propose to set additionally* $u_5 = 0$.

VI. Numeric results

In this section we present simulation results illustrating the performance of proposed polar subcodes of EBCH codes in the case of AWGN channel and BPSK modulation. For comparison, we present also the results for the case of classical polar codes with the corresponding kernels, polar codes with Arikan kernel and CRC-16 (Arikan-CRC) [2], LTE turbo code, as well as LDPC codes specified in WiMAX and CCSDS standards. For polar subcodes with Arikan kernel we have used the block sequential (BS(s)) [36] decoding algorithm[1], where 2^s is the length of outer codes in the IGCC representation of the corresponding polar subcode. For polar codes with the BCH kernel, the sequential decoding algorithm [7] was used, which is based on the order-statistics soft-input hard-output decoding of the component codes. Both probability-domain

[1]For $s \leq 3$ the block sequential decoding algorithm provides slightly inferior performance compared to the probability-domain implementation of the Tal-Vardy list decoding algorithm with the same list size L, but requires much smaller number of arithmetic operations.

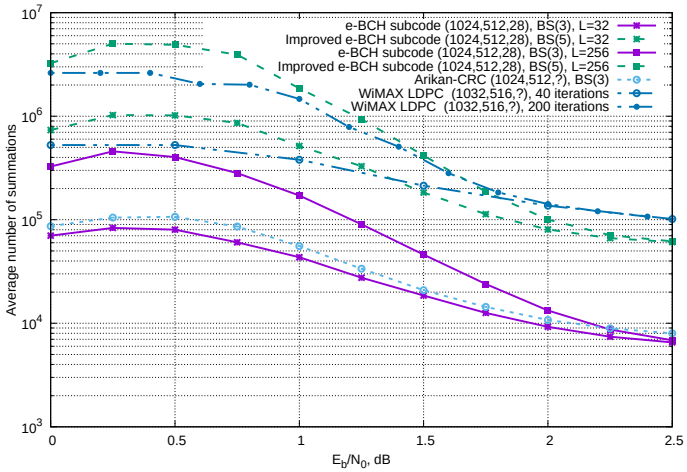


Fig. 8. Decoding complexity for polar subcodes with Arikan kernel

implementation of the Tal-Vardy list decoding algorithm (TV) and the block sequential decoding algorithm were used for decoding of polar codes with Arikan kernel an CRC. Observe that in the case of polar codes with CRC the block sequential decoding algorithm provides slightly worse performance compared to the original Tal-Vardy algorithm, but has much lower complexity. Belief propagation algorithm with flooding schedule was used for decoding of LDPC codes.

Figure 7 illustrates the performance of codes[2] of length ≈ 1024. It can be seen that polar subcodes of EBCH codes provide significant performance gain with respect to the classical polar codes of the same code length and dimension. Furthermore, they outperform polar codes with Arikan kernel and CRC. Observe that increasing s in the case of the block sequential decoding algorithm, i.e. employing in the decoder a representation of a polar subcode as an IGCC with longer outer codes, results in better performance. The best performance is achieved by improved polar subcodes, where outer EBCH codes of length 32 were selected so that the MSD error probability of the corresponding IGCC is minimized.

Figure 8 illustrates the average number of summation operations performed by the block sequential and belief propagation decoding algorithms for the case of polar subcodes and LDPC codes, respectively. Observe that decoding polar subcodes requires slightly lower average number of operations compared to polar codes with CRC, since the dynamic freezing constraints prevent the sequential decoder from constructing wrong paths up to the final phase of decoding.

It can be also seen that for $s = 3$ decoding of polar codes requires 10 times less operations compared to LDPC codes. For $s = 5$ the complexity becomes comparable. Furthermore, the average number of operations for the case of $L = 256, s = 3$ is less than in the case of $L = 32, s = 5$. From these results one may conclude that it is more advantageous to increase L instead of s. However, increasing s enables one to use the improved construction of polar subcodes. We also believe that the block sequential decoding algorithm can be further simplified by employing more efficient decoding algorithms for outer EBCH codes.

Figure 9 presents the performance of codes of length 2048. It can be seen that the proposed improved polar subcodes with Arikan kernel provide substantially better performance compared to the case of LDPC and turbo codes. Observe that increasing minimum distance of the polar subcode results in better performance in the high SNR region. For comparison, we provide also the results for the MHU construction, reproduced from [40]. As it may be expected, the improved polar subcode, which employs EBCH outer codes, provides better performance than the MHU code, which employs outer RM and Arikan polar codes. For comparison, we report also results for the case of a GCC with outer EBCH codes of length 32, which was obtained as described in [37], and decoded with the block sequential algorithm. It appears that some of the outer codes of the IGCC corresponding to the improved polar subcode, which correspond to good bit subchannels, have higher rate, while those corresponding to bad bit subchannels have lower rate than in the case of the classical GCC optimized for the same SNR. This causes the performance of the improved polar subcode to be better than that of the classical GCC.

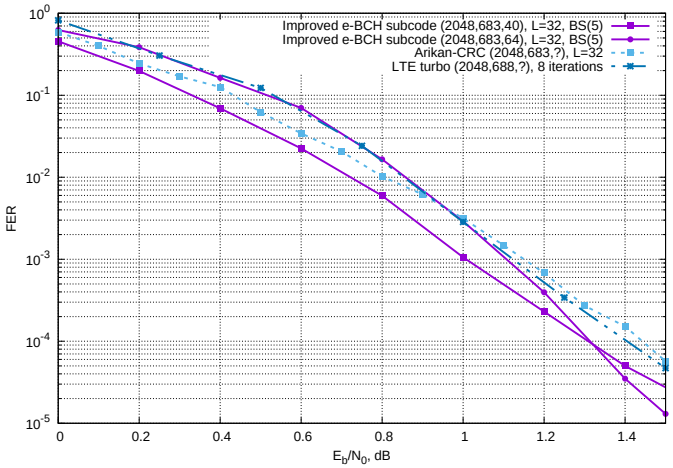
Figure 10(a) presents the performance of codes with the EBCH kernel F_{32}. It can be seen that these codes outperform those with Arikan kernel. For the case of polar subcodes, even better performance can be obtained by increasing the list size L at the cost of higher decoding complexity. Figure 10(b) presents the performance of the binary image of polar codes and polar subcodes with 4×4 Reed-Solomon kernel over $\mathbb{F}_{2^2}$. It can be seen that classical polar codes with Reed-Solomon kernel have quite low minimum distance, similarly to the case of Arikan kernel, but still provide better performance compared to a polar code with Arikan kernel with comparable parameters. In both cases employing the proposed construction of polar subcodes of EBCH codes results in improved minimum distance and even better performance compared to the codes presented in Figures 7(a) and 9(b).

VII. Conclusions

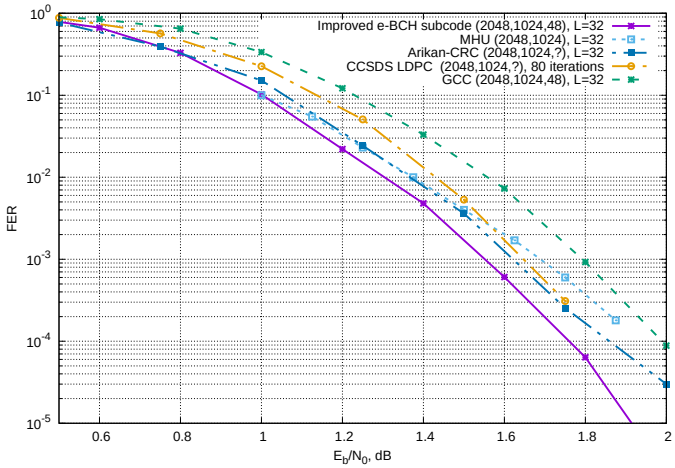
In this paper the construction of polar subcodes of linear block codes was introduced, which is based on the concept of dynamic frozen symbols. It enables one to obtain codes with higher minimum distance than classical polar codes, which can still be efficiently decoded using the derivatives of the list successive cancellation algorithm. Although we do not have a proof that the proposed codes achieve the channel capacity, they were shown to outperform some of the existing LDPC and turbo codes of moderate lengths. Many existing constructions based on polar codes, such as polar codes with CRC, can be considered as a special case of the proposed polar subcodes.

Unfortunately, due to lack of analytical techniques for predicting the performance of list/stack SC decoding algorithms, heuristical methods were used in this paper to construct the codes. Any progress in the performance analysis of these

[2]In order to ensure reproducibility of the results, we have set up a web site http://dcn.icc.spbstu.ru/index.php?id=polar containing the specifications of the considered polar subcodes with Arikan kernel.

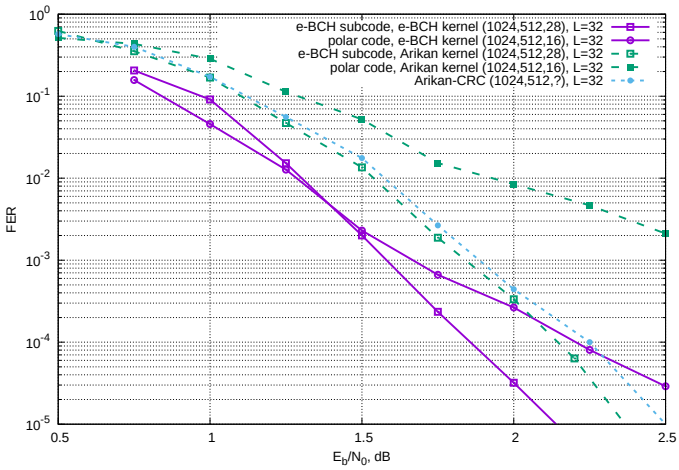


(a) rate $1/3$

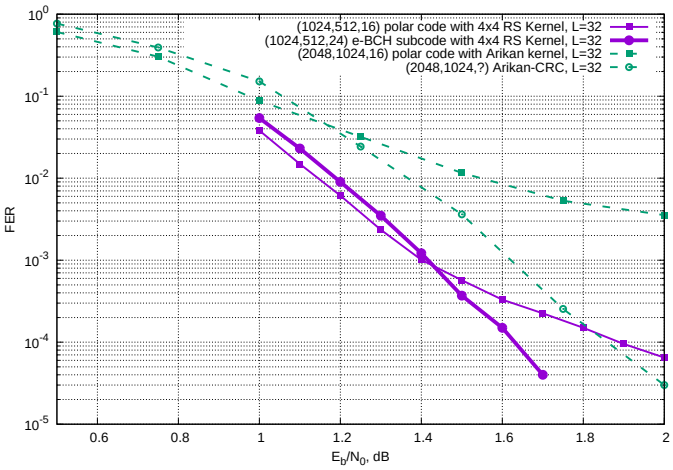


(b) rate $1/2$

Fig. 9. Performance of codes of length 2048



(a) EBCH kernel



(b) Reed-Solomon kernel

Fig. 10. Performance of codes with non-Arikan kernels

algorithms may lead to design of better codes. Another way to improve the performance of the proposed codes is to use longer outer EBCH codes. This, however, requires development of efficient list soft decision decoding algorithms for them.

Furthermore, an extension of the concept of generalized concatenated codes was provided, as well as a new method for representing linear block codes in a form, which enables application of the SC algorithm and its variations for their decoding. This approach enables one to construct polar subcodes with improved performance, as well as a more efficient decoding algorithm for them. It allows also near-ML decoding of short Reed-Solomon codes [44]–[46].

ACKNOWLEDGEMENTS

The authors thank R. Morozov for help in running simulations for polar codes with Reed-Solomon kernel.

The authors thank the anonymous reviewers and the Guest Editor for their helpful comments, which have greatly improved the quality of the paper.

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 *Proc. of IEEE Int. Symp. on Inf. Theory*, 2011, pp. 1–5.

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

[4] K. Chen, K. Niu, and J. Lin, "Improved successive cancellation decoding of polar codes," *IEEE Trans. on Communications*, vol. 61, no. 8, pp. 3100–3107, August 2013.

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

[6] P. Trifonov and V. Miloslavskaya, "Polar codes with dynamic frozen symbols and their decoding by directed search," in *Proc. of IEEE Inf. Theory Workshop*, September 2013, pp. 1 – 5.

[7] V. Miloslavskaya and P. Trifonov, "Sequential decoding of polar codes with arbitrary binary kernel," in *Proc. of IEEE Inf. Theory Workshop*, 2014, pp. 377–381.

[8] J. Guo, M. Qin, A. G. Fabregas, and P. Siegel, "Enhanced belief propagation decoding of polar codes through concatenation," in *Proc. of IEEE Int. Symp. on Inf. Theory*, 2014.

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

[10] H. Mahdavifar, M. El-Khamy, J. Lee, and I. Kang, "On the construction and decoding of concatenated polar codes," in *Proc. of IEEE Int. Symp. on Inf. Theory*, 2013.

[11] Y. Wang and K. R. Narayanan, "Concatenations of polar codes with outer BCH codes and convolutional codes," in *Proc. of 52nd Annual Allerton Conference on Communication, Control, and Computing*, 2014.

[12] M. Bakshi, S. Jaggi, and M. Effros, "Concatenated polar codes," in *Proc. of IEEE Int. Symp. on Inf. Theory*, 2010.

[13] B. Li, H. Shen, and D. Tse, "An adaptive successive cancellation list decoder for polar codes with cyclic redundancy check," *IEEE Communications Letters*, vol. 16, no. 12, December 2012.

[14] K. Niu and K. Chen, "CRC-aided decoding of polar codes," *IEEE Communications Letters*, vol. 16, no. 10, pp. 1668–1671, October 2012.

[15] S. Kumar and H. D. Pfister, "Reed-Muller codes achieve capacity on erasure channels," 2015. [Online]. Available: http://arxiv.org/pdf/1505.05123v2.pdf

[16] S. Kudekar, M. Mondelli, E. Sasoglu, and R. Urbanke, "Reed-Muller codes achieve capacity on the binary erasure channel under MAP decoding," 2015. [Online]. Available: http://arxiv.org/pdf/1505.05831.pdf

[17] I. Dumer and K. Shabunov, "Soft-decision decoding of Reed-Muller codes: Recursive lists," *IEEE Trans. on Inf. Theory*, vol. 52, no. 3, March 2006.

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

[19] H. Imai and S. Hirakawa, "A new multilevel coding method using error correcting codes," *IEEE Trans. on Inf. Theory*, vol. 23, no. 3, pp. 371–377, May 1977.

[20] T. Takata, N. Yamashita, T. Fujiwara, T. Kasami, and S. Lin, "Suboptimum decoding of decomposable block codes," *IEEE Trans. On Inf. Theory*, vol. 40, no. 5, September 1994.

[21] U. Wachsmann, R. F. H. Fischer, and J. B. Huber, "Multilevel codes: Theoretical concepts and practical design rules," *IEEE Trans. on Inf. Theory*, vol. 45, no. 5, pp. 1361–1391, July 1999.

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

[23] R. Mori and T. Tanaka, "Source and channel polarization over finite fields and reed-solomon matrix," *IEEE Trans. on Inf. Theory*, vol. 60, no. 5, May 2014.

[24] S. H. Hassani and R. Urbanke, "On the scaling of polar codes: I. the behavior of polarized channels," in *Proc. of IEEE Int. Symp. on Inf. Theory*, 2010.

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

[26] V. Kolesnik and E. Mironchikov, "Cyclic Reed-Muller codes and their decoding," *Problems of Inf. Transmission*, vol. 4, no. 4, pp. 15–19, 1968.

[27] T. Kasami, S. Lin, and W. Peterson, "New generalizations of the Reed-Muller codes part I: Primitive codes," *IEEE Trans. on Inf. Theory*, vol. 14, no. 2, pp. 189–199, March 1968.

[28] P. Delsarte, J. Goethals, and F. MacWilliams, "On generalized Reed-Muller codes and their relatives," *Inf. and control*, vol. 16, pp. 403–442, 1970.

[29] V. Zinoviev and S. Litsyn, "On the dual distance of BCH codes," *Problems of Inf. Transmission*, vol. 22, no. 4, 1986.

[30] A. Vardy and Y. Beery, "Maximum-likelihood soft decision decoding of BCH codes," *IEEE Trans. On Inf. Theory*, vol. 40, no. 2, March 1994.

[31] T. Kasami, T. Takata, E. Fujiwara, and S. Lin, "On the optimum bit orders with respect to the state complexity of trellis diagrams for binary linear codes," *IEEE Trans. On Inf. Theory*, vol. 39, no. 1, January 1993.

[32] A. Valembois and M. Fossorier, "Box and match techniques applied to soft-decision decoding," *IEEE Trans. on Inf. Theory*, vol. 50, no. 5, pp. 796–810, May 2004.

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

[34] S. H. Hassani, R. Mori, T. Tanaka, and R. Urbanke, "Rate-dependent analysis of the asymptotic behavior of channel polarization," *IEEE Trans. On Inf. Theory*, vol. 59, no. 4, April 2013.

[35] G. Sarkis, I. Tal, P. Giard, A. Vardy, C. Thibeault, and W. J. Gross, "Flexible and low-complexity encoding and decoding of systematic polar codes," 2015. [Online]. Available: http://arxiv.org/pdf/1507.03614v2.pdf

[36] G. Trofimiuk and P. Trifonov, "Block sequential decoding of polar codes," in *Proc. of Int. Symp. on Wireless Communication Systems*, 2015.

[37] P. Trifonov and P. Semenov, "Generalized concatenated codes based on polar codes," in *Proc. of IEEE Int. Symp. on Wireless Communication Systems*, 2011, pp. 442–446.

[38] G. Poltyrev, "Bounds on the decoding error probability of binary linear codes via their spectra," *IEEE Trans. on Inf. Theory*, vol. 40, no. 4, July 1994.

[39] M. Roder and R. Hamzaoui, "Fast tree-trellis list Viterbi decoding," *IEEE Trans. on Communications*, vol. 54, no. 3, pp. 453–461, March 2006.

[40] M. Mondelli, S. H. Hassani, and R. Urbanke, "From polar to Reed-Muller codes: a technique to improve the finite-length performance," *IEEE Trans. on Communications*, vol. 62, no. 9, September , 2014.

[41] B. Liesenfeld and B. Dorsch, "On the equivalence of some generalized concatenated codes and extended cyclic codes," in *Proc. of IEEE Int. Symp. on Inf. Theory*, 1993.

[42] V. Miloslavskaya and P. Trifonov, "Design of polar codes with arbitrary kernels," in *Proc. of IEEE Inf. Theory Workshop*, 2012, pp. 119–123.

[43] P. Trifonov, "Binary successive cancellation decoding of polar codes with Reed-Solomon kernel," in *Proc. of IEEE Int. Symp. on Inf. Theory*, 2014.

[44] ——, "Successive cancellation decoding of Reed-Solomon codes," *Problems of Inf. Transmission*, vol. 50, no. 4, pp. 303–312, 2014.

[45] ——, "Successive cancellation permutation decoding of Reed-Solomon codes," in *Proc. of IEEE Inf. Theory Workshop*, 2014, pp. 386–390.

[46] V. Miloslavskaya and P. Trifonov, "Sequential decoding of Reed-Solomon codes," in *Proc. of Int. Symp. on Inf. Theory and Applications*, 2014, pp. 424–428.

Peter Trifonov (S'02,M'05) was born in St.Petersburg, USSR in 1980. He received the MSc degree in computer science in 2003, and PhD (Candidate of Science) degree from Saint-Petersburg Polytechnic University in 2005. Currently he is an Associate Professor at the Distributed Computing and Networking department of the same university. His research interests include coding theory and its applications in telecommunications and other areas.

Vera Miloslavskaya (S12, M'15) received the B.Sc., M.Sc. and Ph.D. (Candidate of Science) degrees in computer science in 2010, 2012 and 2015, respectively, from Saint-Petersburg Polytechnic University, Russia. Her research interests include coding theory and its applications in telecommunications.