Twisted Polar Codes

Peter Trifonov, Vera Miloslavskaya
Saint-Petersburg State Polytechnic University
Email: {petert,veram}@dcn.icc.spbstu.ru

***Abstract*—A novel construction of polar codes with dynamic frozen symbols and twisted encoding scheme is proposed. It enables one to reduce the error probability and average complexity of the directed search successive cancellation decoding algorithm.**

I. Introduction

Polar codes were recently shown to be able to achieve the capacity of a wide class of communication channels [1]. However, the performance of polar codes appears to be quite poor due to very low minimum distance. It was suggested in [8] to concatenate a polar code with an outer CRC code. Observe that pre-encoding of the data with CRC introduces dependencies between information symbols of the inner polar code. It was suggested in [2] to derive such dependencies from parity check matrices of extended BCH codes, so that sufficiently high minimum distance is enforced, and better performance under list/stack successive cancellation (SC) decoding is achieved.

In the present paper this approach is extended by adjusting the polarization transformation. The only change is that internal bit subchannels induced by the Arikan transformation are permuted (twisted) while being combined to obtain a larger transformation. This enables one to obtain subchannels with better performance, while still being able to use the SC decoding algorithm or its variations. Numeric results show that the proposed construction enables one to reduce the decoding complexity at essentially no cost, provided that directed search SC algorithm is used [2]. The proposed approach is based on a generalization of the classical Plotkin construction, which can be used to represent any linear code of even length.

II. Background

A. Polar codes

Consider a binary input output symmetric memoryless channel with output probability density function $W(y|x), y \in \mathcal{Y}, x \in \mathbb{F}_2$. One can combine two copies of this channel into a synthetic channel $W_2(y_0^1|u_0^1) = W^2(y_0^1|u_0^1F)$, where $F = \begin{pmatrix}1 & 0\\ 1 & 1\end{pmatrix}$, $a_i^j = (a_i, a_{i+1}, \ldots, a_j)$, and $W^2(y_0^1|x_0^1) = W(y_0|x_0)W(y_1|x_1)$. This channel can be further decomposed into bit subchannels $W_2^{(0)}(y_0^1|u_0) = \frac{1}{2}\sum_{u_1} W_2(y_0^1|u_0^1)$ and $W_2^{(1)}(y_0^1, u_0|u_1) = \frac{1}{2}W_2(y_0^1|u_0^1)$. It was shown in [1] that the sum capacity of the transformed channel is equal to the capacity of the original vector channel W^2, and the capacities of bit subchannels satisfy $C(W_2^{(0)}) \leq C(W_2^{(1)})$, with equality only in the case of $C(W) \in \{0, 1\}$. This transformation can be applied recursively, and eventually the original channel decomposes into a number of either almost noise-free or almost pure-noise bit channels. This corresponds to the encoding scheme $c_0^{n-1} = u_0^{n-1}B_mF^{\otimes m}$, where B_m is the bit reversal permutation matrix, and $F^{\otimes m}$ denotes m-times Kronecker product of matrix F with itself. One can set to zero (static bit subchannel freezing) information symbols u_i, which should be transmitted over low-capacity subchannels to obtain a polar code. Let $\mathcal{F}$ be the set of frozen symbols. Observe that $(n = 2^m, k)$ polar code can be considered as a Plotkin concatenation $(u + v|u), u \in \mathcal{C}_0, v \in \mathcal{C}_1$, of polar codes $\mathcal{C}_i$ of length 2^{m-1}. Reed-Muller code $RM(r, m)$ can be considered as a special case of polar codes, obtained by freezing all subchannels with indices $i : \mathrm{wt}(i) < m - r$.

A simple method to decode a polar code is given by the SC algorithm, which makes decisions

$$\widehat{u}_i = \begin{cases} \arg\max_{u_i \in \{0,1\}} W_n^{(i)}(\widehat{u}_0^{i-1}, u_i|y_0^{n-1}), & i \notin \mathcal{F} \\ 0, & \text{otherwise,} \end{cases} \tag{1}$$

where

$$W_n^{(2i)}(u_0^{2i}|y_0^{n-1}) = \sum_{u_{2i+1}=0}^{1} W_{\frac{n}{2}}^{(i)}(u_{0,e}^{2i+1} + u_{0,o}^{2i+1}|y_0^{\frac{n}{2}-1})W_{\frac{n}{2}}^{(i)}(u_{0,o}^{2i+1}|y_{\frac{n}{2}}^{n-1}) \tag{2}$$

$$W_n^{(2i+1)}(u_0^{2i+1}|y_0^{n-1}) = W_{\frac{n}{2}}^{(i)}(u_{0,e}^{2i+1} + u_{0,o}^{2i+1}|y_0^{\frac{n}{2}-1})W_{\frac{n}{2}}^{(i)}(u_{0,o}^{2i+1}|y_{\frac{n}{2}}^{n-1}). \tag{3}$$

The SC decoding error probability of the polar code is

$$\Pi = 1 - \prod_{i=0, i\notin\mathcal{F}}^{2^m-1}(1 - P_i), \tag{4}$$

where P_i is the bit subchannel error probability, which can be computed via density evolution [6] or its Gaussian approximation [3]. In practice, set $\mathcal{F}$ is commonly constructed as a set of $n-k$ subchannel indices i, corresponding to the highest P_i. Observe that this method for construction of polar codes is not guaranteed to be optimal if list [8] or stack [4] SC decoding algorithms are used. Performance analysis of these algorithms still remains an open problem.

B. Dynamic frozen symbols

In order to improve minimum distance of polar codes, it was suggested in [2] to set some information symbols u_i not statically to 0, but to some linear combination of other symbols, i.e. enforce constraint

$$u_0^{n-1}B_mF^{\otimes m}H^T = 0, \tag{5}$$

where H is a $(n-k)\times n$ check matrix of some (n, K, d) parent code (e.g. extended BCH) with sufficiently high minimum distance d. Elementary row operations given by matrix Q can be used to obtain $V = QH(B_m F^{\otimes m})^T$, such that at most one row ends in each column of matrix V, i.e.

$$u_{j_i} = \sum_{s<j_i, V_{is}=1} u_s, 0 \le i < n-K, \tag{6}$$

where the i-th row ends in column j_i. Here j_i are the indices of dynamic frozen symbols. One can additionally freeze $K-k$ symbols corresponding to bad bit subchannels to obtain $(n, k, \ge d)$ polar code. The parent code needs to be selected so that $\{j_0, \dots, j_{n-K-1}\}$ does not include indices of good bit subchannels.

III. Generalized Plotkin decomposition

In this section we show that any linear code of even length can be represented using a construction similar to Plotkin one.

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

$$\tilde{G} = \begin{pmatrix} I_{k_1} & 0 & \tilde{I} \\ 0 & I_{k_2} & 0 \end{pmatrix} \begin{pmatrix} G_1 & 0 \\ G_2 & G_2 \\ G_3 & G_3 \end{pmatrix}, \tag{7}$$

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

Proof: Let $G = \begin{pmatrix} G' & G'' \end{pmatrix}$, where G' and G'' are some $k \times n$ matrices, be a generator matrix of the code, and let $H = \begin{pmatrix} H' & H'' \end{pmatrix}$ 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 $\tilde{G} = QG = \begin{pmatrix} G_2 & G_2 \\ G_4 & G_3 \\ G_5 & 0 \end{pmatrix}$, 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

$$\tilde{G} = \begin{pmatrix} I_{k_2} & 0 & 0 & 0 \\ 0 & I_{k_3} & I_{k_3} & 0 \\ 0 & 0 & 0 & I_{k_5} \end{pmatrix} \begin{pmatrix} G_2 & G_2 \\ G_3 & G_3 \\ G_4 - G_3 & 0 \\ G_5 & 0 \end{pmatrix}. \tag{8}$$

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

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 $k_3 = 0$. Therefore, the representation of code generator matrix in the form (7) will be referred to as a generalized Plotkin decomposition (GPD) of G or the corresponding code $\mathcal{C}$.

GPD enables one to perform hard-decision decoding of code $\mathcal{C}$ as follows. Let $\mathcal{C}_i$ is the code generated by G_i. Consider 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 using the techniques described in [5].

One can see from (7) that $\mathcal{C}_2$ has minimum distance $d_2 \ge d/2$. However, d_1 can be arbitrarily low. Therefore, the above described decoding algorithm may fail to correct even $t \le \lfloor (d-1)/2 \rfloor$ 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 vector $\tilde{y}''$ in $\mathcal{C}_2$, and select the codeword $(c'|c'')$ closest to the received sequence.

It must be recognized that applying GPD to equivalent codes may result in codes $\mathcal{C}_1, \mathcal{C}_2$ with substantially different performance. Furthermore, GPD can be applied recursively to $\mathcal{C}_1, \mathcal{C}_2$, and one can arbitrarily and independently permute columns of G_1 and G_2 prior to applying GPD to the corresponding codes. Ultimately, this process results in codes of length 1 and dimension at most 1, i.e. frozen and non-frozen symbols.

IV. Twisted polar codes

In this section we introduce interleaving into the construction of polar codes. This does not affect polarization properties, but enables construction of better codes with dynamic frozen symbols. Namely, one can search for permutations, such that the set of dynamic frozen symbols does not include indices of good bit subchannels.

A. Subchannel twisting

For $m > 1$, let the twisted polarizing transformation (TPT) be given by $2^m \times 2^m$ matrix S, which is recursively defined as $S_t = \begin{pmatrix} S_{t0} & 0 \\ S_{t1} & S_{t1} \end{pmatrix} P_t$, where t is the transformation index, P_t is a permutation matrix, S_{t0}, S_{t1} are (possibly different) TPTs of dimension $2^{m-1} \times 2^{m-1}$, and for $m = 1$ the transformation is given by matrix F. The encoding operation is given by $c_0^{n-1} = u_0^{n-1} S$. If codeword c_0^{n-1} is transmitted over a binary input memoryless output symmetric channel, then the permutations introduced above do not affect the properties of channels $W_n^{(i)}$ induced by transformation S_t.

One can use the TPT in the construction of polar codes with dynamic frozen symbols, i.e. impose constraint $u_0^{n-1} S H^T = 0$, where H is a check matrix of some $(n = 2^m, K, d)$ code $\mathcal{C}$. This can be seen as recursive application of the GPD to code $\mathcal{C}$, where at each step of the decomposition some permutation is applied to the generator matrix of the code.

To obtain $(n, k, \ge d)$ twisted polar code (TPC) corresponding to transformation S, one should construct the dynamic freezing constraints corresponding to parent (n, K, d) code, and freeze additional $K-k$ bit subchannels with the highest error probability. Methods proposed in [6], [3] can be used to compute error probabilities for subchannels $W_n^{(i)}$ induced by the TPT.

Employing different permutation matrices P_t at various steps of the GPD results in substantially different sets of

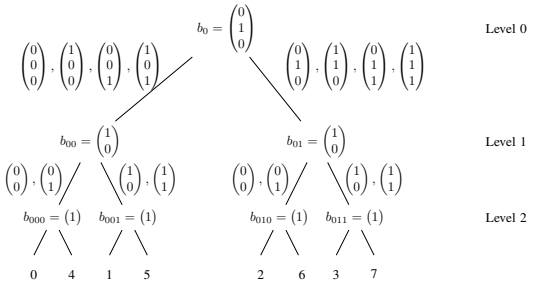

Fig. 1. Subspace splitting tree for $b_0 = (0,1,0)^T$

dynamic frozen symbols j_i (see (6)). One may be interested in finding such set of permutations, so that the error probability (4) of parent code $\mathcal{C}$ under SC decoding is minimized. It is computationally infeasible to check all possible permutations P_t. However, in the case of extended BCH codes much smaller set of permutations needs to be considered.

B. Subcodes of extended BCH codes

It is possible to show that any extended primitive narrow-sense BCH code $\mathcal{C}$ satisfies $RM(r,m) \subset \mathcal{C} \subset RM(r+1,m)$ for some r, where $RM(r,m)$ is a Reed-Muller code of length 2^m and order r [7]. Therefore, setting all P_t to identity matrices, one obtains that many of the dynamic frozen bit subchannels induced by $\mathcal{C}$ are low-capacity ones. One may be interested in preserving this property in twisted polar codes.

Recall that codewords of $RM(r+1,m)$ are obtained by evaluating Zhegalkin polynomials $f(X)$ of degree at most $r+1$ at distinct points $X \in \mathbb{F}_2^m$. Plotkin decomposition of such code gives codes $\mathcal{C}_1 = \left\{(f'_b(X^{(1)}),\dots,f'_b(X^{(2^{m-1})}))|X^{(i)} \in \mathcal{L}\right\}$, and $\mathcal{C}_2 = \left\{(f(X^{(1)}),\dots,f(X^{(2^{m-1})}))|X^{(i)} \in \mathcal{L}\right\}$, where $\mathcal{L}$ is a $(m-1)$-dimensional linear subspace of $\mathbb{F}_2^m$, b is a splitting vector such that $\mathcal{L} \cup (b+\mathcal{L}) = \mathbb{F}_2^m$, and $f'_b(X) = f(X+b) + f(X)$. Observe that $\deg g(X) \leq r$. Codewords of an extended BCH code correspond to some linear subspace of the space of degree-$(r+1)$ Zhegalkin polynomials.

Each coordinate of codewords of $\mathcal{C}$ can be identified with vector $X = (x_0,\dots,x_{m-1}) \in \mathbb{F}_2^m$, and permutation matrix P_t can be given by a bijective mapping $\psi : \mathbb{F}_2^m \to \mathbb{F}_2^m$. In order to construct GPD of $\mathcal{C}$ with $\mathcal{C}_1 \subset RM(r,m-1), \mathcal{C}_2 \subset RM(r+1,m-1)$ we propose to set $\psi(X) = x_{m-1}b + A(x_0,\dots,x_{m-2})^T$, where rows of $m \times (m-1)$ matrix A represent a basis of $\mathcal{L}$, and b is the corresponding splitting vector. TPT with permutation matrices P_t on all levels of recursion defined in this way will be referred to as linear TPT. Let b_t denote the splitting vector corresponding to matrix P_t. These vectors can be arranged into a subspace splitting tree, as shown in Figure 1. Observe that the Arikan polarizing transformation is obtained by setting $b_{i_0 i_1 \dots i_{s-1}} = (1,0,\dots,0) \in \mathbb{F}_2^{m-s}$.

Example 1. *Consider the subspace splitting tree shown in Figure 1. Splitting vectors* $b_{00} = b_{01} = b_{10} = b_{11} = (1,0)^T$ *correspond to* $S_{00} = S_{01} = S_{10} = S_{11} = \begin{pmatrix} 1&0&0&0\\1&0&1&0\\1&1&0&0\\1&1&1&1 \end{pmatrix}$. *Splitting vectors* $b_0 = (0,1,0)^T$ *and* $b_1 = (1,0,0)^T$ *correspond to*

$$S_0 = \begin{array}{c} \begin{matrix} 0&2&1&3&4&6&5&7 \end{matrix} \\ \begin{pmatrix} 1&0&0&0&0&0&0&0\\1&0&0&0&1&0&0&0\\1&1&0&0&0&0&0&0\\1&1&0&0&1&1&0&0\\1&0&1&0&0&0&0&0\\1&0&1&0&1&0&1&0\\1&1&1&1&0&0&0&0\\1&1&1&1&1&1&1&1 \end{pmatrix} \end{array}, S_1 = \begin{array}{c} \begin{matrix} 0&1&2&3&4&5&6&7 \end{matrix} \\ \begin{pmatrix} 1&0&0&0&0&0&0&0\\1&0&0&0&1&0&0&0\\1&0&1&0&0&0&0&0\\1&0&1&0&1&0&1&0\\1&1&0&0&0&0&0&0\\1&1&0&0&1&1&0&0\\1&1&1&1&0&0&0&0\\1&1&1&1&1&1&1&1 \end{pmatrix} \end{array}.$$

Figure 2 presents the transformation given by S_0. *Observe that it differs from the Arikan's one by an output permutation. Finally, the top-level splitting vector* $b = (1,1,0,0)^T$ *produces*

$$S = \begin{pmatrix} 1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\1&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0\\1&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0\\1&0&1&0&0&0&0&0&1&0&1&0&0&0&0&0\\1&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0\\1&0&0&0&1&0&0&0&1&0&0&0&1&0&0&0\\1&0&1&0&1&0&1&0&0&0&0&0&0&0&0&0\\1&0&1&0&1&0&1&0&1&0&1&0&1&0&1&0\\1&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0\\1&0&0&1&0&0&0&0&1&0&0&1&0&0&0&0\\1&0&0&1&1&0&0&1&0&0&0&0&0&0&0&0\\1&0&0&1&1&0&0&1&1&0&0&1&1&0&0&1\\1&1&1&1&0&0&0&0&0&0&0&0&0&0&0&0\\1&1&1&1&0&0&0&0&1&1&1&1&0&0&0&0\\1&1&1&1&1&1&1&1&0&0&0&0&0&0&0&0\\1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1 \end{pmatrix}$$

Given $(16,7,6)$ *extended BCH code, one obtains*

$$uS \underbrace{\begin{pmatrix} 1&0&0\\1&1&1\\1&\alpha&\alpha^3\\1&1+\alpha&(1+\alpha)^3\\1&\alpha^2&1+\alpha+\alpha^2+\alpha^3\\1&1+\alpha^2&1+\alpha\\1&\alpha(1+\alpha)&1+\alpha^2\\1&1+\alpha+\alpha^2&1+\alpha+\alpha^2+\alpha^3\\1&\alpha^3&1+\alpha^2\\1&1+\alpha^3&1+\alpha\\1&\alpha(1+\alpha^2)&1\\1&1+\alpha+\alpha^3&1\\1&\alpha^2(1+\alpha)&1+\alpha\\1&1+\alpha^2+\alpha^3&\alpha^3\\1&\alpha(1+\alpha+\alpha^2)&1+\alpha^2\\1&1+\alpha+\alpha^2+\alpha^3&\alpha^3 \end{pmatrix}}_{H^T} = 0,$$

where α *is a primitive root of* x^4+x^3+1. *Expanding matrix* H *in the standard basis, and applying column operations to matrix* SH^T, *one obtains :*

$$\mathbf{u} \begin{pmatrix} 0&0&0&0&0&1&1&0&0&0&0&0&1&0&0&0\\0&0&0&0&0&1&1&0&0&0&1&0&0&0&0&0\\0&0&0&0&0&0&1&0&0&1&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0\\0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0\\0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0\\0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0\\0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \end{pmatrix}^T = 0,$$

i.e. $u_0 = u_1 = u_2 = u_3 = u_4 = u_8 = 0$, $u_9 = u_6$, $u_{10} = u_5 + u_6, u_{12} = u_5 + u_6$. *In the case of Arikan transformation* $S = B_m F^{\otimes m}$ *symbol* u_6 *would be frozen instead of* u_3.

In the case of BEC with erasure probability 0.5 *the Bhattacharyya parameters of subchannels corresponding to* u_3 *and* u_6 *are* 0.77 *and* 0.53, *respectively. Hence, TPT enables one to freeze a bad subchannel, and unfreeze a better one.*

One can still use (1)–(3) to perform SC decoding of TPCs, except that u_0^{2i+1} should be partitioned into subvectors corresponding to $\mathcal{L}$ and $\mathcal{L}+b$, instead of subvectors of elements with even and odd indices. Furthermore, (6) should be used in (1) for $i \in \mathcal{F}$.

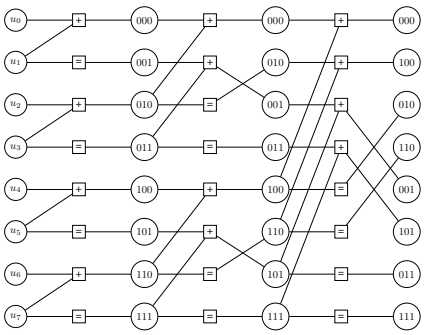

Fig. 2. Twisted 8-dimensional polarizing transformation S_0

C. Optimizing linear TPT

Consider construction of a linear TPC based on some parent code $\mathcal{C}$. This requires one to find a TPT S, such that the error probability (see (4)) of the obtained code under SC decoding is minimized. There is no explicit expression relating splitting vectors and error probability under SC decoding. Therefore, we propose a suboptimal algorithm for its minimization.

The first step is to perform randomized search for a linear TPT S, such that the SC decoding error probability of the code, obtained from $\mathcal{C}$ by applying to it S, and additionally freezing all subchannels with bit error probability more than π for some $\pi \in (0, 0.5)$, is minimized. Every leaf of the subspace slitting tree corresponds to a subchannel of the TPT. Observe that the bit error rate for these subchannels under SC decoding does not depend on particular splitting vectors. Furthermore, one does not need to optimize splitting vectors for the subtrees having all leafs corresponding to subchannels with error probability more than π. One does not need also to optimize splitting vectors in the subtrees, where matrix G_2 arising at the corresponding level of the recursive GPD of code $\mathcal{C}$ is equal to identity matrix. This reduces search space significantly. The second step is to freeze a number of additional subchannels with the highest error probability to obtain a code of dimension k.

Figure 3 illustrates the proposed code construction algorithm. It accepts as input integer m, generator matrix G of $(2^m, K, d)$ parent code $\mathcal{C}$, and vector P of subchannel error probabilities of 2^m-dimensional polarizing transformation, as well as threshold π. At lines 3–5 upper bound on subchannel non-error probabilities is computed. Observe that all subchannels with $P_i \geq \pi$ are assumed to be frozen, so they are error-free. At lines 6–9 lower bound $p_{i,j}$ on bit error rate of subchannels corresponding to subtrees of the splitting tree are computed, as well as lower bounds on the probability of correct SC decoding

$$\gamma_{m-i,j} \leq \prod_{s=j2^{m-i},s\notin\mathcal{F}}^{(j+1)2^{m-i}-1} (1-P_s)$$

over all possible sets of frozen symbols $\mathcal{F}$. At line 10 randomized search for a TPT, that would maximize probability λ of correct SC decoding of a code obtained from $\mathcal{C}$ by

TWISTEDCONSTRUCTOR(m, G, k, P, π)
1 **for** $j \leftarrow 0$ **to** $2^m - 1$
2 **do** $p_{0,j} \leftarrow P_i$
3 **if** $P_j < \pi$
4 **then** $\gamma_{0,j} \leftarrow 1 - P_j$
5 **else** $\gamma_{0,j} \leftarrow 1$
6 **for** $i \leftarrow 1$ **to** $m - 1$
7 **do for** $j \leftarrow 0$ **to** 2^{m-i}
8 **do** $p_{i,j} \leftarrow \min(p_{i-1,2j}, p_{i-1,2j+1})$
9 $\gamma_{i,j} \leftarrow \gamma_{i-1,2j}\gamma_{i-1,2j+1}$
10 $[\lambda, b, \mathcal{F}] \leftarrow$ RANDOMSPLIT$(m, G, p, \pi, \gamma, 0)$
11 $Z \leftarrow$ SORT$(([P_i, i], 0 \leq i < 2^m)$
12 $j \leftarrow 0$
13 **while** $|\mathcal{F}| < 2^m - k$
14 **do if** $Z_{j,1} \notin \mathcal{F}$
15 **then** $\mathcal{F} \leftarrow \mathcal{F} \cup \{Z_{j,1}\}$
16 $j \leftarrow j + 1$
17 **return** $(\mathcal{F}, b)$

Fig. 3. Construction of TPC

RANDOMSPLIT(m, G, p, π, l)
1 $K \leftarrow$ DIMENSION(G)
2 **if** $K = 0 \vee p_{m,l} > \pi$
3 **then** $\mathcal{F} \leftarrow \{0, \ldots, 2^m - 1\}; b_s \leftarrow \mathbf{e}_0, s \in \bigcup_{p=0}^{m-1} \mathbb{F}_2^p$
4 **return** $[1, b, \mathcal{F}]$
5 **if** $K = 2^m$
6 **then** $b_s \leftarrow \mathbf{e}_0, s \in \bigcup_{p=0}^{m-1} \mathbb{F}_2^p; \mathcal{F} = \emptyset$
7 **return** $[\gamma_{m,l}, b, \mathcal{F}]$
8 $\lambda \leftarrow 0$
9 **for** $t \leftarrow 1$ **to** T
10 **do** $\beta \leftarrow$ RAND(m)
11 Construct partitioning $\mathbb{F}_2^m = \mathcal{L} \cup (\mathcal{L} + \beta)$
12 $(G_1, G_2, G_3) \leftarrow$ GPD$(G, \mathcal{L}, \mathcal{L} + \beta)$
13 $[\tilde{p}, \tilde{b}, \tilde{\mathcal{F}}] =$ RANDOMSPLIT$(m - 1, G_1, p, \pi, 2l)$
14 $[\hat{p}, \hat{b}, \hat{\mathcal{F}}] =$ RANDOMSPLIT$(m - 1, G_2, p, \pi, 2l + 1)$
15 **if** $\tilde{p}\hat{p} > \lambda$
16 **then** $\lambda = \tilde{p}\hat{p}; \mathcal{F} = \tilde{\mathcal{F}} \cup \left\{2^{m-1} + j | j \in \hat{\mathcal{F}}\right\}$
17 $b \leftarrow (\beta, \tilde{b}, \hat{b})$
18 **return** $[\lambda, b, \mathcal{F}]$

Fig. 4. Randomized search for linear TPT

freezing all bit subchannels with error probability more than π, is performed. The result is a tree of splitting vectors b and the set of (static or dynamic) frozen channels $\mathcal{F}$. Then error probabilities of bit subchannels are re-arranged in descending order at line 11, and indices of additional frozen channels are identified. Parameter π should be set so that all subchannels with bit error rate more than π are frozen after this step.

Figure 4 presents a randomized algorithm for construction of linear TPT. It returns probability λ on correct SC decoding of the code, obtained from the one generated by G by freezing all bit subchannels with error probability more than π, a tree

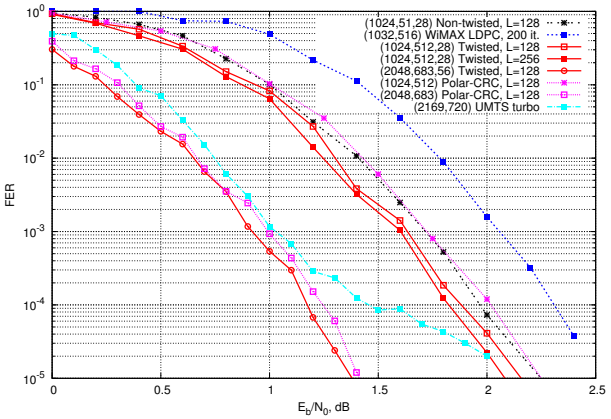

Fig. 5. Performance of polar codes

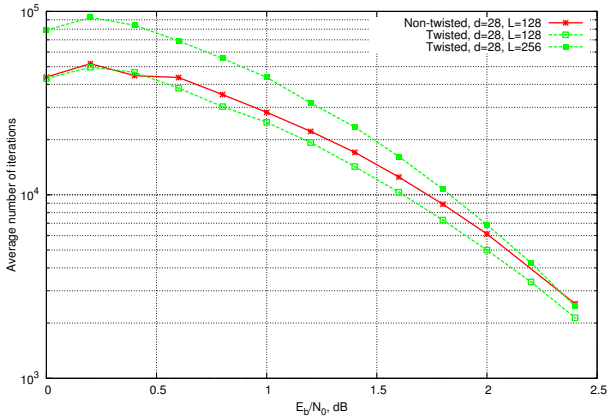

Fig. 6. Number of decoder iterations for $(1024, 512)$ codes

of splitting vectors b (its nodes are denoted b_s), and the set of frozen channels $\mathcal{F}$. For codes of dimension 0 and subtrees of the splitting tree, such that all the corresponding subchannels have too high error probability, all subchannels are assumed to be frozen, and Arikan splitting vectors $e_0 = (1, 0, \ldots, 0)$ are selected. These vectors are used also for codes of rate 1. In this case selection of frozen symbols is not performed, since this will be done later by algorithm $TwistedConstructor$.

In all other cases a number of attempts (given by parameter T) to construct a GPD of code generated by G is performed. Random vector $\beta \in \mathbb{F}_2^m \setminus \{0\}$ is used to partition $\mathbb{F}_2^m$ into a linear space $\mathcal{L}$ and its coset. Columns of G are indexed with binary m-vectors in the standard bit order, and permuted, so that those indexed with elements in $\mathcal{L}$ are placed on the first 2^{m-1} positions, and the corresponding columns indexed with elements in $\mathcal{L} + b$ are placed on the last positions. GPD results in matrices G_1, G_2, G_3. The algorithm is recursively applied to G_1 and G_2 to obtain probabilities $\tilde{p}, \hat{p}$ of correct SC decoding of the corresponding polar codes of length 2^{m-1}. The decomposition providing the highest probability $\tilde{p}\hat{p}$ of correct decoding of the polar code obtained from G is identified, the sets of frozen symbols obtained for G_1 and G_2 are combined, and the corresponding splitting trees $\tilde{b}, \hat{b}$ are attached as children to the root node given by β.

It appears that two TPCs with the same error probability under SC decoding may have substantially different performance under improved versions of SC decoding. Therefore, we propose to execute $TwistedConstructor$ a number of times, run simulations for each obtained code to identify their performance under a given decoding algorithm, and select the one with the lowest decoding error probability.

V. Numeric results

Figure 5 presents the performance of twisted and non-twisted polar codes obtained from extended BCH codes, polar code concatenated with outer CRC one [8], LDPC and turbo codes. Decoding of polar codes was performed using the directed search algorithm introduced in [2], where the maximal number of paths L of any given length in the code tree considered simultaneously by the decoder was set to 128. It can be seen that TPCs provide approximately 0.1 dB gain compared to non-twisted and CRC-based ones. Furthermore, there is no error floor, as in the case of the turbo code.

Figure 6 shows that the average number of decoder iterations is lower by approximately 20% compared to the case of non-twisted codes. For high SNR this enables one to increase L and obtain better performance with the same complexity.

VI. Conclusions

In this paper a novel construction of polar codes was proposed. It enables one to ensure that the code has sufficiently high minimum distance by employing an appropriate extended BCH parent code. Subchannel twisting enables one to obtain better set of unfrozen bit subchannels. This results not only in lower decoding error probability, but also in lower average decoding complexity.

Acknowledgements

This work was supported by Samsung Electronics, and partially by the grant of the President of Russia MK-5407.2013.9.

References

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

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

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

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

[5] I. Dumer and R. Krichevskiy, "Soft-decision majority decoding of Reed-Muller codes," *IEEE Transactions On Information Theory*, vol. 46, no. 1, January 2000.

[6] I. Tal and A. Vardy, "How to construct polar codes," *IEEE Transactions On Information Theory*, vol. 59, no. 10, October 2013.

[7] T. Kasami, S. Lin, and W. Peterson, "New generalizations of the Reed-Muller codes part i: Primitive codes," *IEEE Transactions on Information Theory*, vol. 14, no. 2, March 1968.

[8] I. Tal and A. Vardy, "List decoding of polar codes," in *Proceedings of IEEE International Symposium on Information Theory*, 2011.