Recursive Trellis Decoding Techniques of Polar Codes

Peter Trifonov
ITMO University, Russia
Email: pvtrifonov@itmo.ru

***Abstract*—Recursive trellis decoding of polar codes is considered. Polar codes are shown to have much lower recursive trellis decoding complexity compared to similar Reed-Muller codes. Furthermore, a low-latency decoding algorithm, which combines recursive trellis and successive cancellation decoding methods, is presented.**

I. Introduction

Arikan polar codes [1] were recently adopted for encoding of short data blocks, arising in the 5G control channel. One of the problems, which prevents their application for encoding of long data blocks, is huge decoding latency of the successive cancellation (SC) algorithms and its derivatives.

In this paper we show that polar codes admit efficient maximum likelihood decoding via the recursive trellis algorithm (RTA) [2]. We show that the trellises induced by polar codes have the number of states much lower compared to similar Reed-Muller codes. Furthermore, the recursive tree-like nature of RTA immediately results in substantial reduction of the decoding latency compared to the latency of SC based algorithms. This comes at the expense of the increased number of arithmetic operations performed by the decoder. To obtain a low-latency decoding algorithm for long polar codes with reasonable complexity, we further propose to combine the recursive trellis and SC decoding methods.

The paper is organized as follows. Section II provides a background on polar codes, as well as a brief overview of the recursive trellis decoding algorithm. The complexity of the latter algorithm in the case of polar codes is studied in Section III. A novel low-latency decoding algorithm for polar codes, which combines RTA and SC decoding, is presented in Section IV. The results illustrating its complexity and performance are presented in Section V.

II. Background

A. Polar codes

Consider the Arikan polarizing transformation given by $A_m = B_m \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}^{\otimes m} = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}^{\otimes m} B_m$, where B_m is a bit-reversal permutation[1] matrix. This construction can be also extended to the case of mixed-kernel polarizing transformations [3]. It is possible to show that a binary input memoryless symmetric channel $W(y|c) = W_0^{(0)}(y|c)$ together with matrix A gives rise to $n = 2^m$ bit subchannels

$$W_m^{(i)}(\mathbf{y}_0^{n-1}, u_0^{i-1}|u_i) = \frac{1}{2^{n-1}} \sum_{u_{i+1}^{n-1} \in \mathbb{F}_2^{n-i-1}} \prod_{j=0}^{n-1} W_0^{(0)}(\mathbf{y}_j\,|(u_0^{n-1}A_m)_j), i \in [n], \quad (1)$$

with capacities approaching 0 or 1 symbols per channel use, and fraction of noiseless subchannels approaching $I(W)$, the capacity of channel W, where $[n] = \{0, \dots, n-1\}$. The Bhattacharyya parameter $Z_{m,i}$ of subchannel $W_m^{(i)}$ satisfies

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

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

An $(n = 2^m, k)$ polar code over $\mathbb{F}_2$ is a set of codewords $c_0^{n-1} = u_0^{n-1}A_m$, where $u_i = 0, i \in \mathcal{F}$, $\mathcal{F} \subset [n]$ is the set of frozen symbol indices (frozen set), and $|\mathcal{F}| = n - k$. The set $\mathcal{F}$ is typically selected as the set of indices i of subchannels $\mathcal{W}_m^{(i)}$, such that $Z_{m,i} \ge \pi$ for some threshold π. Such codes will be referred to as *canonical polar codes*.

Observe that the (i, j)-th element of A_m is equal to 1 iff $R_m(j) \preceq i$, where

$$R_m\left(\sum_{s=0}^{m-1} i_s 2^s\right) = \sum_{s=0}^{m-1} i_s 2^{m-1-s}, i_s \in \{0, 1\},$$

is the bit-reversal operation, and

$$\sum_{s=0}^{m-1} i_s 2^s \preceq \sum_{s=0}^{m-1} j_s 2^s \Leftrightarrow i_s \le j_s.$$

It is possible to show that for canonical polar codes $\mathcal{F}$ satisfies the *domination contiguity* property, i.e. $i \in \mathcal{F}$ implies $j \in \mathcal{F}$ for any $j : j \preceq i$ [4].

It is convenient to describe the SC algorithm in terms of probabilities $W_m^{(i)}\left\{v_0^i|\,\mathbf{y}_0^{n-1}\right\}$ of transmission of various vectors $v_0^{n-1}A_m$ with given values v_0^i, provided that the receiver observes a noisy vector $\mathbf{y}_0^{n-1}$, i.e.

$$W_m^{(i)}\left\{v_0^i|\,\mathbf{y}_0^{n-1}\right\} = \frac{W_m^{(i)}(\mathbf{y}_0^{n-1}, v_0^{i-1}|v_i)}{2W(\mathbf{y}_0^{n-1})}$$

$$= \sum_{v_{i+1}^{n-1}} W_m^{(n-1)}\left\{v_0^{n-1}|\,\mathbf{y}_0^{n-1}\right\} = \sum_{v_{i+1}^{n-1}} \prod_{j=0}^{n-1} W\left\{(v_0^{n-1}A_m)_j|\,\mathbf{y}_j\right\} \quad (4)$$

[1]Sometimes polar codes are introduced without the permutation matrix B_m. This matrix, however, is essential for obtaining reasonably low decoding complexity with the recursive trellis decoding algorithm.

These values can be approximated as $W_m^{(i)}\left\{v_0^i|\mathbf{y}_0^{n-1}\right\} \approx \mathcal{W}_m^{(i)}\left\{v_0^i|\mathbf{y}_0^{n-1}\right\} = \max_{u_{i+1}^{n-1}}\prod_{j=0}^{n-1} W\left\{(v_0^{n-1}A_m)_j|\mathbf{y}_j\right\}$. These probabilities can be recursively computed as

$$\mathcal{W}_\lambda^{(2i)}\left\{v_0^{2i}|\mathbf{y}_0^{2^\lambda-1}\right\} = \max_{v_{2i+1}} \mathcal{W}_{\lambda-1}^{(i)}\left\{v_{0,e}^{2i+1}\oplus v_{0,o}^{2i+1}|\mathbf{y}_0^{2^{\lambda-1}-1}\right\}\mathcal{W}_{\lambda-1}^{(i)}\left\{v_{0,o}^{2i+1}|\mathbf{y}_{2^{\lambda-1}}^{2^\lambda-1}\right\} \quad (5)$$

$$\mathcal{W}_\lambda^{(2i+1)}\left\{v_0^{2i+1}|\mathbf{y}_0^{2^\lambda-1}\right\} = \mathcal{W}_{\lambda-1}^{(i)}\left\{v_{0,e}^{2i+1}\oplus v_{0,o}^{2i+1}|\mathbf{y}_0^{2^{\lambda-1}-1}\right\}\mathcal{W}_{\lambda-1}^{(i)}\left\{v_{0,o}^{2i+1}|\mathbf{y}_{2^{\lambda-1}}^{2^\lambda-1}\right\}, \quad (6)$$

where $0<\lambda\le m$, $a_{0,e}^t$ and $a_{0,o}^t$ denote subvectors of a_0^t, consisting of elements with even and odd indices, respectively. By transforming these values into the corresponding LLRs, one obtains min-sum version of the SC algorithm [5].

At phase i the SC decoder makes decision

$$\widehat{u}_i = \begin{cases}\arg\max_{v_i\in\mathbb{F}_2}\mathcal{W}_m^{(i)}\left\{\widehat{u}_0^{i-1}\bullet v_i|\mathbf{y}_0^{n-1}\right\}, & i\notin\mathcal{F}\\ \text{the frozen value of } u_i, & \text{otherwise,}\end{cases}$$

where $a\bullet b$ denotes a vector obtained by appending b to a. The number of sequential steps (decoding latency) needed to compute $\mathcal{W}_m^{(i)}$ and make decisions $\hat{u}_i$ is given by

$$\Lambda_{SC} = m + \sum_{i=1}^{m} 2^{m-i} i = 2n-2.$$

B. Recursive trellis decoding of binary linear codes

A recursive trellis-based maximum likelihood decoding algorithm (RTA) for linear codes was proposed in [2]. Given a linear code C, let $C_{h,h'}$ be its subcode, such that all its codewords have non-zero symbols only in positions $h\le i<h'$. Let $p_{h,h'}(C)$ be a linear code obtained by puncturing all symbols, except those in positions $h\le i<h'$, from codewords of C. Let us further define $s_{h,h'}(C)=p_{h,h'}(C_{h,h'})$, i.e. a code obtained from C by shortening it on all symbols except those with indices $h\le i<h'$. $s_{h,h'}(C)$ and $p_{h,h'}(C)$ will be referred to as *section codes*. Consider a minimal trellis of code C, and its section corresponding to symbols from x to y. It is possible to show that the paths between two adjacent states in this section correspond to a coset in $p_{x,y}(C)/s_{x,y}(C)$. Furthermore, there may be several paths in the trellis corresponding to the same coset. This enables one to reduce the complexity of maximum likelihood (ML) decoding of code C by computing the metrics of these paths once, and using several times. That is, for each coset $D\in p_{x,y}(C)/s_{x,y}(C)$ one needs to identify the most probable element $l(D)=c_x^{y-1}\in D$, and its log-probability $m(D)=\log W\left\{c_x^{y-1}|\mathbf{y}_x^{y-1}\right\}=\sum_{i=x}^{y-1}\log W\left\{c_y|\mathbf{y}_i\right\}$. Let the *composite branch table* $CBT_{x,y}$ be a data structure storing the pairs $(l(D),m(D))$. In the case of ML decoding of some (n,k) code, $p_{0,n}(C)/s_{0,n}(C)$ contains a single element, so $CBT_{0,n}$ contains a single entry, which corresponds to a solution of the ML decoding problem.

The straightforward approach to construction of a composite branch table for some code C is to enumerate all codewords of $p_{x,y}(C)$, and find the most probable one for each coset in $p_{x,y}(C)/s_{x,y}(C)$. However, more efficient approach was suggested in [2] for the case of $y-x\ge 2$.

Consider some $z: x<z<y$. Let the generator matrix of $p_{x,y}(C)$ be represented as

$$G_{x,y}^{(p)} = \begin{pmatrix} G_{x,z}^{(s)} & 0\\ 0 & G_{z,y}^{(s)}\\ G_{x,y}^{(00)} & G_{x,y}^{(01)}\\ \hline G_{x,y}^{(10)} & G_{x,y}^{(11)}\end{pmatrix}, \quad (7)$$

where $G_{x,y}^{(s)} = \begin{pmatrix} G_{x,z}^{(s)} & 0\\ 0 & G_{z,y}^{(s)}\\ G_{x,y}^{(00)} & G_{x,y}^{(01)}\end{pmatrix}$ is a generator matrix of $s_{x,y}(C)$, and $G_{x,y}^{(00)}, G_{x,y}^{(01)}$ are some $k''_{x,y}\times(z-x)$ and $k''_{x,y}\times(y-z)$ matrices, respectively. There is an one-to-one correspondence between vectors $vG'_{x,y}$, where $G'_{x,y}=\begin{pmatrix}G_{x,y}^{(10)} & G_{x,y}^{(11)}\end{pmatrix}$ is a $k'_{x,y}\times(y-x)$ matrix, and cosets $D\in p_{x,y}(C)/s_{x,y}(C)$. Here $k'_{x,y}, k''_{x,y}$ are some integers, which depend on code structure, and can be obtained from the minimum span form of its generator matrix.

Hence, we write $CBT_{x,y}[v].l := l(D)$ and $CBT_{x,y}[v].m = m(D)$, with D being a coset corresponding to v. It can be seen that

$$CBT_{x,y}[v].m = \max_{u\in\mathbb{F}_2^{k''_{x-y}}}\left(CBT_{x,z}[a].m + CBT_{z,y}[b].m\right), v\in\mathbb{F}_2^{k'_{x,y}} \quad (8)$$

where a and b are indices of the cosets $D'\in p_{x,z}(C)/s_{x,z}(C)$ and $D''\in p_{z,y}(C)/s_{z,y}(C)$, respectively, such that $\begin{pmatrix}u & v\end{pmatrix}\begin{pmatrix}G_{x,y}^{(00)}\\ G_{x,y}^{(10)}\end{pmatrix}\in D'$ and $\begin{pmatrix}u & v\end{pmatrix}\begin{pmatrix}G_{x,y}^{(01)}\\ G_{x,y}^{(11)}\end{pmatrix}\in D''$. Such values a,b can be identified from the following system of equations:

$$(a',a)\begin{pmatrix}G_{x,z}^{(s)}\\ G'_{x,z}\end{pmatrix} = \begin{pmatrix}u & v\end{pmatrix}\begin{pmatrix}G_{x,y}^{(00)}\\ G_{x,y}^{(10)}\end{pmatrix}$$

$$(b',b)\begin{pmatrix}G_{z,y}^{(s)}\\ G'_{z,y}\end{pmatrix} = \begin{pmatrix}u & v\end{pmatrix}\begin{pmatrix}G_{x,y}^{(01)}\\ G_{x,y}^{(11)}\end{pmatrix},$$

where a', b' are some irrelevant values. Obviously, the solutions are given by $a=\begin{pmatrix}u & v\end{pmatrix}\widehat{G}_{x,y}$ and $b=\begin{pmatrix}u & v\end{pmatrix}\widetilde{G}_{x,y}$ for some matrices $\widehat{G}_{x,y}$ and $\widetilde{G}_{x,y}$. The corresponding most likely coset representatives are given by $CBT_{x,y}[v].l = CBT_{x,z}[\hat{a}].l\bullet CBT_{x,z}[\hat{b}].l$, where $\hat{a},\hat{b}$ are the values of a and b, which deliver maximum in (8).

The complexity of this calculation is $O(2^{k'_{x,y}+k''_{x,y}})$. It can be further reduced by exploiting the tricks suggested in [2]. The overall decoding complexity strongly depends on the sectionalization method being used, i.e. a rule for selection of the partitioning point z for some x,y. For the sake of simplicity, we assume here that $z=(x+y)/2$, although in some cases the decoding complexity may be reduced by more careful selection of z.

III. Recursive trellis decoding of polar codes

It was shown in [6] that the SC decoding algorithm for polar codes can be considered as successive application of the recursive trellis decoding algorithm to $(n+1, n-i)$ codes $\overline{\mathcal{C}}^{(i)}$, such that their generator matrices are obtained by taking rows $i, \ldots, n-1$ of matrix A_m and appending $(1,0,\ldots,0)^T$ column to it, with some intermediate results being reused for different phases $i = 0, \ldots, n-1$. These codes were shown to have the following properties:

1) For any $i < 2^m$, any $\tau < m$, and any x being multiple of 2^τ
$$p_{x,x+2^\tau}(\overline{\mathcal{C}}^{(i)}) = p_{x+2^\tau, x+2^{\tau+1}}(\overline{\mathcal{C}}^{(i)})$$
and
$$s_{x,x+2^\tau}(\overline{\mathcal{C}}^{(i)}) = s_{x+2^\tau, x+2^{\tau+1}}(\overline{\mathcal{C}}^{(i)}).$$
2) For any x being multiple of 2^τ one has $p_{x,x+2^\tau}(\overline{\mathcal{C}}^{(i)}) = p_{x,x+2^\tau}(\overline{\mathcal{C}}^{(i-1)})$ and $s_{x,x+2^\tau}(\overline{\mathcal{C}}^{(i)}) = s_{x,x+2^\tau}(\overline{\mathcal{C}}^{(i-1)})$, for any $i = 2^s i' > 0$, and $1 \leq \tau < m - s$.
3) If binary sectionalization is used, i.e. for any x, y the trellises are partitioned at time $z = \frac{x+y}{2}$, then $k''_{x,y} \leq 1$.

This implies that the SC algorithm is equivalent to successive application of the RTA to codes $\overline{\mathcal{C}}^{(i)}$ [6].

The above facts were established not for classical polar codes, but for artificial codes $\overline{\mathcal{C}}^{(i)}$, which arise in the course of SC decoding. Nevertheless, they suggest that the trellises introduced in [2] may be very simple in the case of polar codes.

In this section we show this explicitly by studying the codes $p_{x,y}(C)$ and $s_{x,y}(C)$, which are obtained from a polar code C in the case of binary sectionalization.

Theorem 1. *Given a length-2^m polar code C with a frozen set $\mathcal{F}$, $p_{x,y}(C)$ is a polar code with the frozen set $\mathcal{F}'_{\tau,t} = \{i \in [2^\tau] | \forall j \in [\nu] : R_{m-\tau}(t) \preceq j \Rightarrow i\nu + j \in \mathcal{F}\}$, where $x = 2^\tau t, y = 2^\tau(t+1), t \in [\nu], \nu = 2^{m-\tau}$.*

Proof. The symbols of codeword of the polar code c_0^{n-1} are given by
$$c_j = \sum_{\substack{i \in \mathcal{F} \\ R_m(j) \preceq i}} u_i.$$
The codewords $\widetilde{c}_0^{2^\tau - 1}$ of the punctured code are given by
$$\begin{aligned}\widetilde{c}_s =& c_{2^\tau t+s} = \sum_{j=0}^{2^{m-\tau}-1} \sum_{i=0}^{2^\tau-1} u_{i2^{m-\tau}+j} \chi_{R_m(2^\tau t+s) \preceq i2^{m-\tau}+j} \\ =& \sum_{j=0}^{2^{m-\tau}-1} \sum_{i=0}^{2^\tau-1} u_{i2^{m-\tau}+j} \chi_{R_\tau(s) \preceq i \wedge R_{m-\tau}(t) \preceq j} \\ =& \sum_{i=0}^{2^\tau-1} \chi_{R_\tau(s) \preceq i} \underbrace{\sum_{j=0}^{2^{m-\tau}-1} u_{i2^{m-\tau}+j} \chi_{R_{m-\tau}(t) \preceq j}}_{\widetilde{u}_i}, 0 \leq s < 2^\tau,\end{aligned}$$
where $\chi_Z = 1$ iff Z is true. Obviously, $\widetilde{u}_i$ is identically zero iff for all $j \in [2^\tau]$, such that $R_{m-\tau}(t) \preceq j$, one has $i2^{m-\tau} + j \in \mathcal{F}$. □

For canonical polar codes $i\nu + \nu - 1 \in \mathcal{F}$ implies $i\nu + j \in \mathcal{F}$ for all $j \in [\nu]$, so one obtains $\mathcal{F}'_{\tau,t} = \{i \in [2^\tau] | i\nu + \nu - 1 \in \mathcal{F}\}$. This set appears to be independent of t.

Lemma 1. *Consider a canonical polar code with frozen set $\mathcal{F}$. Then the check matrix of this code is given by rows $n - 1 - i, i \notin \mathcal{F}$, of matrix A_m.*

Proof. It can be seen that A_m^T is obtained by reversing the order of columns and rows in A_m. This implies that the dual of a polar code with frozen set $\mathcal{F}$ is a code generated by rows $n-1-i, i \notin \mathcal{F}$ of matrix A_m, written in the reverse order [7]. To conclude the proof, observe that the automorphism group of polar codes includes permutation $\pi(i) = n - 1 - i$ [4]. □

Theorem 2. *Given a length-2^m canonical polar code C with a frozen set $\mathcal{F}$, $s_{x,y}(C)$ is a polar code with the frozen set $\mathcal{F}''_{\tau,t} = \{i \in [2^\tau] | i\nu \in \mathcal{F}\}$, where $x = 2^\tau t, y = 2^\tau(t+1), 0 \leq t < \nu, \nu = 2^{m-\tau}$.*

Proof. Observe that a check matrix of a shortened code $s_{x,y}(C)$ can be obtained as a generator matrix of the punctured dual code $p_{x,y}(C^\perp)$. Lemma 1 implies that the dual of C is a polar code with frozen set $\mathcal{F}^\perp = \{i | n-1-i \notin \mathcal{F}\}$. Theorem 1 implies that $p_{x,y}(C^\perp)$ is a polar code with the frozen set $\mathcal{F}' = \{i | \forall j \in [2^{m-\tau}] : R_{m-\tau}(t) \preceq j \Rightarrow n - 1 - (i2^{m-\tau} + j) \notin \mathcal{F}\}$. Hence, $s_{x,y}(C)$ is a polar code given by frozen set $\mathcal{F}'' = \{i | 2^\tau - 1 - i \notin \mathcal{F}'\} = \{i | \exists j \in [2^{m-\tau}] : R_{m-\tau}(t) \preceq j \wedge n - 1 - ((2^\tau - 1 - i)2^{m-\tau} + j) \in \mathcal{F}\}$. Here it is sufficient to take $j = \nu - 1$. □

The above theorem implies that for polar code C one has
$$\begin{aligned}&k''_{2^\tau t, 2^\tau(t+1)} = \dim(s_{2^\tau t, 2^\tau(t+1)}(C)) - \dim(s_{2^\tau t, 2^\tau t + 2^{\tau-1}}(C)) \\ &- \dim(s_{2^\tau t+2^{\tau-1}, 2^\tau t + 2^\tau}(C)) = |\mathcal{F}''_{\tau-1,2t}| + |\mathcal{F}''_{\tau-1,2t+1}| - |\mathcal{F}''_{\tau,t}| \\ &= 2|\left\{i \in [2^{\tau-1}] | i2^{m-\tau+1} \in \mathcal{F}\right\}| - |\left\{i \in [2^\tau] | i2^{m-\tau} \in \mathcal{F}\right\}|,\end{aligned}$$
where $\dim(C)$ is the dimension of code C. Furthermore,
$$\begin{aligned}k'_{2^\tau t, 2^\tau(t+1)} =& \dim(p_{2^\tau t, 2^\tau(t+1)}(C)) - \dim(s_{2^\tau t, 2^\tau(t+1)}(C)) \\ =& |\left\{i \in [2^\tau] | i2^{m-\tau} \in \mathcal{F}\right\}| \\ & - |\left\{i \in [2^\tau] | (i+1)2^{m-\tau} - 1 \in \mathcal{F}\right\}|\end{aligned}$$

Figure 1 presents an example of the recursion tree for the recursive trellis decoding algorithm. Each node in the tree shows the values of x, y, $k'_{x,y}$, and $k''_{x,y}$.

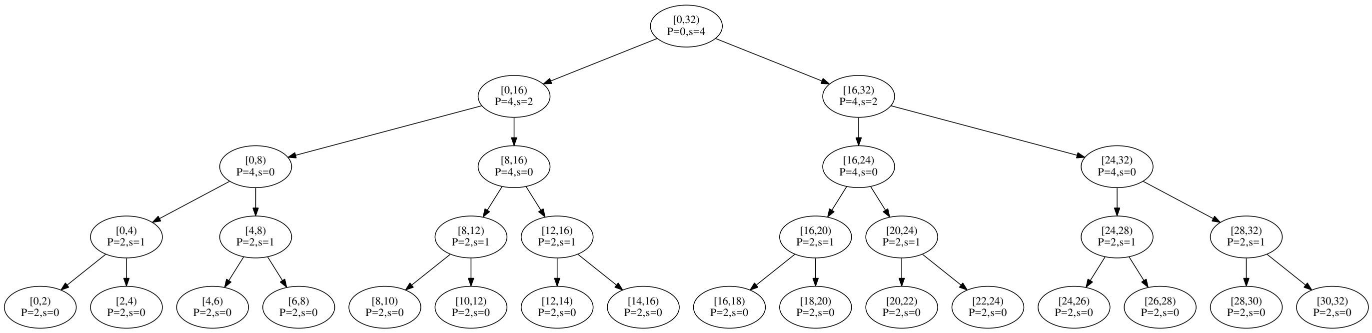

Fig. 1: Decomposition tree for $(32, 16)$ polar code with $\mathcal{F} = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 16, 17, 18, 20\}$

The total complexity of computing (8) for sections of length $y - x = 2^\tau$ is proportional to $2^\delta = 2^{m-\tau} \cdot 2^{k_{2^\tau t, 2^\tau(t+1)} + k''_{2^\tau t, 2^\tau(t+1)}}$, where

$$\delta =(m-\tau) + (k''_{2^\tau t, 2^\tau(t+1)} + k'_{2^\tau t, 2^\tau(t+1)})$$
$$=(m-\tau) + \Bigg(2 \underbrace{\left|\left\{i \in [2^{\tau-1}] | i2^{m-\tau+1} \in \mathcal{F}\right\}\right|}_{N_1} - \underbrace{\left|\left\{i \in [2^{\tau}] | (i+1)2^{m-\tau} - 1 \in \mathcal{F}\right\}\right|}_{N_2} \Bigg)$$

Since $i2^{m-\tau+1} \preceq (2i+1)2^{m-\tau}$, the domination contiguity property implies that

$$N_1 \geq N_1' = \left|\left\{i \in [2^{\tau-1}] | (2i+1)2^{m-\tau} \in \mathcal{F}\right\}\right|,$$

and $2N_1 \geq N_1 + N_1' = |\{i \in [2^\tau] | i2^{m-\tau} \in \mathcal{F}\}| = |\{i \in [2^\tau] | Z_{m, i2^{m-\tau}} \geq \pi\}| \overset{(2)}{\geq} \{i \in [2^\tau] | Z_{\tau,i} \geq \pi 2^{\tau - m}\}$. Furthermore, $N_2 = \left\{i \in [2^\tau] | Z_{m,(i+1)2^{m-\tau}-1} < \pi\right\} = \left\{i \in [2^\tau] | Z_{\tau,i} \overset{(3)}{<} \pi^{2^{\tau-m}}\right\}$. Hence, for sufficiently small π, $2N_1 - N_2$ is lower bounded by the number $\hat{N}$ of imperfectly polarized subchannels $W_\tau^{(i)}$. This number is known [8] to satisfy, at least for the case of the binary erasure channel,

$$\lim_{\tau \to \infty} 2^{\tau/\mu} \frac{\hat{N}}{2^\tau} = q(Z_{0,0}, \pi),$$

where μ is the scaling exponent of polar codes, and $q(Z_{0,0}, \pi)$ is some positive value. That is, the number of imperectly polarized subchannels grows with code length. Hence, the complexity of the recursive trellis decoding algorithm is dominated by a few top layers of the recursion.

The latency of the recursive trellis decoding algorithm is given by

$$\Lambda_{RTA} = \sum_{\tau=1}^{m} \lambda_\tau,$$

where $\lambda_\tau = O(k''_{x,x+2^\tau})$ is the latency of computing (8).

The complexity of the recursive trellis decoding algorithm can be substantially reduced by employing the techniques introduced in [9]. This would, however, come at the price of substantially higher decoding latency.

TABLE I: RTA latency and complexity, $n = 128$

	Reed-Muller		Polar		
k	Complexity	Latency	Design E_b/N_0, dB	Complexity	Latency
8	895	8	0	383	6
29	4334911	23	1	8575	14
64	4425123935	38	2	142655	22
99	151 111 775	37	3	43583	20

IV. Hybrid trellis-SC decoding of polar codes

Since the complexity of the recursive trellis decoding algorithm grows exponentially with code length, it is natural to implement a hybrid decoding method, which would combine low latency of RTA, and low complexity of SC decoding.

Any polar code of length 2^m can be considered as a generalized concatenated code [10] with inner $(2^\eta, 2^\eta - i)$ polar codes, and outer polar codes of length 2^τ, where $\tau = m - \eta$, and the i-th outer code has frozen set $\mathcal{F}_i = \{j \in [2^\tau] | i2^\tau + j \in \mathcal{F}\}$ [11]. The SC decoding algorithm can be considered as an instance of the multistage decoding method for multilevel (generalized concatenated) codes [12], where outer codes of length 1 are used. However, by considering the polar code as a generalized concatenated code with longer outer codes, one can obtain performance or complexity gain by employing the multistage decoding method together with the appropriate decoding algorithms for outer codes.

We propose to employ the recursive trellis algorithm for decoding of outer codes. Due to polarization provided by the inner polarizing transformation A_η, most of the outer codes arising in the GCC representation of the polar code are either low-, or high-rate ones. This ensures that their RTA decoding complexity is sufficiently low, while the decoding latency is equal to $\Lambda_{SCT} = \sum_{i=0}^{2^\eta - 1} \Lambda_i l_i$, where Λ_i is the RTA decoding latency of the i-th outer code, and l_i is the latency of computing input LLRs or probabilities based on (5)–(6). If the intermediate results of these calculations are appropriately reused [13], then l_i is equal to $\min(\eta, b(i) + 1)$, where $b(i)$ is the number of least significant bits of i equal to 0.

The proposed approach can be considered as an improvement of [14]. Indeed, RTA enables efficient decoding of sufficiently long non-trivial outer codes.

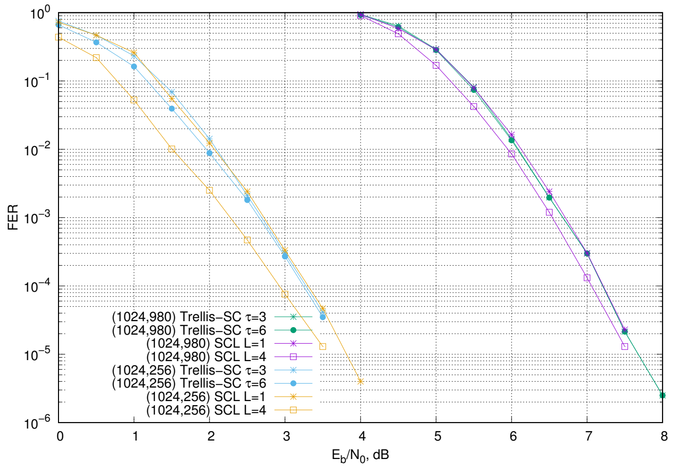

Fig. 2: Performance of polar codes under Trellis-SC decoding

TABLE II: Decoding complexity and latency for polar codes of length 1024

	(1024, 256)	(1024, 980)
SCL complexity, $L = 1$	10240	10240
SCL complexity, $L = 4$	40960	40960
Trellis-SC complexity, $\tau = 3$	7764	7830
Trellis-SC latency, $\tau = 3$	361	401
Trellis-SC complexity, $\tau = 6$	43248	11109
Trellis-SC latency, $\tau = 6$	122	81
RTA complexity		9101823

V. Numeric results

Table I presents decoding complexity and latency for some polar and Reed-Muller codes of length 128 under the recursive trellis decoding algorithm. Polar codes were constructed via the Gaussian approximation method [11] for AWGN channel with SNR shown in the table. The complexity is reported in terms of the number of summation and comparison operations. It can be seen that polar codes have much lower decoding complexity and latency compared to Reed-Muller codes, although the complexity appears to be higher compared to $n \log_2 n = 896$, the complexity of the SC algorithm, except for the case of $k = 8$, where the RTA appears to be simpler. Recall, that the RTA always provides maximum likelihood decoding, while the SC algorithm is highly suboptimal.

Figure 2 illustrates the performance of polar codes of length 1024 under the proposed trellis-SC algorithm. For comparison, we report also the performance of SC list decoding algorithm [13] with various values of list size L. It can be seen that the trellis-SC algorithm provides somewhat better performance compared to the case of SC decoding ($L = 1$). However, its performance is inferior compared to the case of Tal-Vardy list decoding with $L > 1$. The complexity and latency of the considered algorithms is reported in Table II. It can be seen that the complexity of the proposed trellis-SC decoding method in the worst case is close to that of the SC list decoder, an in some cases is much lower. It is also much lower compared to the case of immediate application of the RTA to the considered polar code. The latency of the trellis-SC algorithm is much lower compared to $\Lambda_{SC} = 2n - 2$, latency of the SC algorithm. Observe that a recent reduced-latency decoding algorithm [15] requires 226 time steps for $(1024, 256)$ polar code, 85% more compared to the proposed algorithm.

VI. Conclusions

In this paper it was shown that polar codes admit very efficient decoding by the recursive trellis decoding algorithm. Furthermore, a hybrid low-latency decoding method based on the recursive trellis and successive cancellation algorithms was proposed.

In general, the recursive trellis algorithm, while providing maximum likelihood decoding, has much higher complexity compared to the successive cancellation list decoding method with comparable performance. Reducing the recursive trellis algorithm complexity requires exploiting the techniques introduced in [9]. Implementation of these tricks without heavy latency penalty is the subject of further study.

On the other hand, the performance of the proposed trellis-SC algorithm may be improved by employing list decoding techniques. This, however, requires developing an extension of the recursive trellis algorithm, capable of producing a list of most probable codewords. This would enable one to perform low-latency near-ML decoding of improved versions of polar codes, such as polar codes with CRC [13] and polar subcodes [16].

Acknowledgements

This work was supported by the Ministry of Science and Higher Education of Russian Federation, project (Goszadanie) no. 2019-0898.

References

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

[2] T. Fujiwara, H. Yamamoto, T. Kasami, and S. Lin, “A trellis-based recursive maximum-likelihood decoding algorithm for binary linear block codes,” *IEEE Transactions On Information Theory*, vol. 44, no. 2, March 1998.

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

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

[5] P. Trifonov, “A score function for sequential decoding of polar codes,” in *Proceedings of IEEE International Symposium on Information Theory*, Vail, USA, 2018.

[6] ——, “Trellis-based decoding techniques for polar codes with large kernels,” in *Proceedings of IEEE Information Theory Workshop*, 2019, invited paper.

[7] M. P. Fossorier, “Polar codes: Graph representation and duality,” *IEEE Communications Letters*, vol. 19, no. 9, September 2015.

[8] S. H. Hassani, K. Alishahi, and R. Urbanke, “Finite-length scaling for polar codes,” *IEEE Transactions On Information Theory*, vol. 60, no. 10, October 2014.

[9] T. Kasami, "Adaptive recursive mld algorithm based on parallel concatenation decomposition for binary linear codes," in *Coding, Cryptography and Combinatorics*, ser. Progress in Computer Science and Applied Logic, K. Feng, H. Niederreiter, and C. Xing, Eds. Basel: Birkhäuser, 2004, vol. 23.

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

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

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

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

[14] A. Alamdar-Yazdi and F. Kschischang, "A simplified successive-cancellation decoder for polar codes," *IEEE Communications Letters*, vol. 15, no. 12, December 2011.

[15] M. H. Ardakani, M. Hanif, M. Ardakani, and C. Tellambura, "Fast successive-cancellation-based decoders of polar codes," *IEEE Transactions On Communications*, vol. 67, no. 7, July 2019.

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