Abstract—A shortening method for large polarization kernels is presented, which results in shortened kernels with the highest error exponent if applied to kernels of size up to 32. It uses lower and upper bounds on partial distances for quick elimination of unsuitable shortening patterns.

The proposed algorithm is applied to some kernels of sizes 16 and 32 to obtain shortened kernels of sizes from 9 to 31. These kernels are used in mixed-kernel polar codes of various lengths. Numerical results demonstrate the advantage of polar codes with shortened large kernels compared with shortened and punctured Arikan polar codes, and polar codes with small kernels.

I. INTRODUCTION

Polar codes are a novel class of error-correcting codes, which achieves the symmetric capacity of a binary-input memoryless channel $W$. They have low complexity construction, encoding, and decoding algorithms [1]. However, Arikan polar codes allow length $2^n$ only, which restricts their practical application. To cope with it, many length-compatible constructions were proposed, including shortened and punctured polar codes, chained polar codes [2] and asymmetric construction [3].

Polarization is a general phenomenon and is not restricted to Arikan matrix [4]. One can replace it by a larger matrix, called polarization kernel, which has better polarization properties obtaining thus better finite length performance. Polar codes with large kernels were shown to provide an asymptotically optimal scaling exponent [5]. By combining different kernels, one can construct mixed-kernel (MK) polar code [6], which may have wide range of possible lengths. For instance, hybrid design based on kernels of dimension 3 and 5 was suggested in [7]. Unfortunately, such small kernels have quite poor polarization properties. As a result, these codes do not outperform length-compatible constructions based on Arikan matrix.

At the same time, various kernels with good properties were recently introduced together with simplified processing algorithms, including $16 \times 16$ and $32 \times 32$ kernels for the window processing algorithm [8]. BCH kernels with reduced trellis processing complexity [9], approximate processing based on Box and Match algorithm [10], convolutive polar kernels [11] and some kernels [12] which allow low complexity processing by recursive trellis processing algorithm [13]. However, obtaining polar codes of arbitrary length requires application of code shortening and puncturing, which are not yet well understood for the case of non-Arikan kernels.

To address this issue we consider shortening of large polarization kernels [4]. This allows to obtain kernels of arbitrary smaller size, which can be processed by the same algorithm as the original kernel. In this paper we propose an algorithm for kernel shortening, resulting in shortened kernels with the highest error exponent if applied to kernels of size up to 32. The proposed algorithm examines all shortening patterns. Nevertheless, it avoids explicit evaluation of partial distance for all considered shortening pattern by using lower and upper bounds, substantially reducing thus optimization complexity.

We applied the proposed method to kernels of size 16 and 32, obtained shortened kernels of sizes from 9 to 31, and used them in MK polar codes of lengths $32 \cdot l$, $17 \leq l \leq 31$. Simulation results show that the obtained codes provide significant performance gain compared with shortened and punctured Arikan polar codes, and polar codes with small kernels.

II. BACKGROUND

A. Notations

For a positive integer $n$, we denote by $[n]$ the set $\{0, 1, \ldots, n-1\}$. For vectors $a$ and $b$ we denote their concatenation by $a \otimes b$. The vector $u^b_i$ is a subvector $(u_0, u_{a+1}, \ldots, u_b)$ of a vector $u$. $M[i]$ is the $i$-th row of the matrix $M$. By $M[j,i]$ we denote $j$-th element of $M[i]$. For matrices $A$ and $B$ we denote their Kronecker product by $A \otimes B$. By $A \otimes m$ we denote $m$-fold Kronecker product of matrix $A$ with itself.

B. Polarization transformation

Consider a binary-input memoryless channel with transition probabilities $W(y|c)$, $c \in \mathbb{F}_2$, $y \in \mathcal{Y}$, where $\mathcal{Y}$ is output alphabet. An $(n = l^m, k)$ polar code is a linear block code generated by $k$ rows of matrix $G_m = M(m)K \otimes m$, where $M(m)$ is a digit-reversal permutation matrix, corresponding to mapping $\sum_{i=0}^{m-1} t_i b^i \rightarrow \sum_{i=0}^{m-1} t_{m-1-i} b^i$, $t_i \in [l]$. The encoding scheme is given by $c_{0}^{n-1} = u_{0}^{n-1}G_{m}$, where $u_{i} \in \mathcal{F}$ are set to some pre-defined values, e.g. zero (frozen symbols), $|\mathcal{F}| = n-k$, and the remaining values $u_{i}$ are set to the payload data.

It is possible to show that a binary input memoryless channel $W$ together with matrix $G_{m}$ gives rise to bit subchannels $W_{m,K}(y_{0}^{n-1}, u_{0}^{n-1}u_{l})$ with capacities approaching 0 or 1, and fraction of noiseless subchannels approaching $I(W)$ [4]. Selecting $F$ as the set of indices of low-capacity subchannels enables almost error-free communication.

It is convenient to define probabilities $W_{m}^{(l)}(u_{0}^{l-1}|y_{0}^{n-1}) = \sum_{u_{l+1}^{n-1}}W_{m}^{(l)}((u_{0}^{l-1}G_{m})i|y_{l})$, where matrix $G_{m}$ will be clear from the context.

Due to the recursive structure of $G_{m}$, one has

$$W_{m}^{(l+t)}(u_{0}^{l+t}|y_{0}^{n-1}) = \sum_{u_{l+t+1}^{n-1}}\prod_{j=0}^{t-1}W_{m-1}^{(s)}(u_{l+t}^{l+j+1}G_{m}^{-1}i|y_{l+j+1})^{1/2-1},$$  \((1)\)
where $\theta_K[u_0^{(s+1)l-1}, j], \gamma = (u_l^{(r+1)l-1} K), r \in [s+1].$

At the receiver side, the successive cancellation (SC) decoding algorithm makes decisions

$$\hat{u}_i = \left\{ \arg \max_{u_i \in \mathbb{F}_2} W_m^{(i)}(\tilde{u}_0^{i-1}, u_i | y_0^{n-1}), \quad i \notin \mathcal{F}, \quad \text{the frozen value of } u_i \right\} \quad \text{if } i \in \mathcal{F}. \quad (2)$$

Since the probabilities $[3]$ are computed recursively, it is convenient to consider the probabilities for one layer of the polarization transform, which are given by

$$W_1^{(i)}(u_0^{i-1}| y_0^{j-1}) = \sum_{u_0^{i-1} \in \mathbb{F}_2} W_1^{(i-1)}(u_0^{i-1}| y_0^{j-1}). \quad (3)$$

The task of computing (3) is referred to as kernel processing. The value of $\phi$ is referred to as processing phase. Computing these probabilities reduces to soft-output decoding of nonsystematically encoded codes generated by the last $l - \phi - 1$ rows of $K$. This problem was considered in [14].

An $(n, k)$-mixed-kernel polar code [6, 15] is a linear block code generated by $k$ rows of matrix $G_m = K_1 \otimes K_2 \otimes \ldots \otimes K_m$, where $K_i$ is an $l_i \times l_i$ polarization kernel and $n = \prod_{i=1}^{m} l_i$.

C. Fundamental parameters of polar codes

1) Error exponent: Let $W : \{0, 1\} \rightarrow \mathcal{Y}$ be a symmetric binary-input discrete memoryless channel (B-DMC) with capacity $I(W)$. By definition,

$$I(W) = \sum_{y \in \mathcal{Y}} \sum_{x \in \{0, 1\}} 2^{-W(y|x)} \log \frac{W(y|x)}{2 W(y|0) + W(y|1)}.$$

The Bhattacharyya parameter of $W$ is

$$Z(W) = \sqrt{W(y|0) W(y|1)}.$$

Consider the polarizing transform $K^{\otimes m}$, where $K$ is an $l \times l$ polarization kernel, and bit subchannels $W_0^{(i)}(y_0^{i-1}, u_0^{i-1}|u_i)$, induced by it. Let $Z_m^{(i)} = Z(W_m^{(i)}(y_0^{n-1}, u_0^{n-1}|u_i))$ be a Bhattacharyya parameter of $i$-th subchannel, where $i$ is uniformly distributed on the set $[m]$. Then, for any B-DMC $W$ with $0 < I(W) < 1$, we say that an $l \times l$ matrix $K$ has error exponent (also known as a rate of polarization) $E(K)$ if

(i) For any fixed $\beta < E(K)$,

$$\liminf_{n \to \infty} \Pr[Z_n \leq 2^{-n^{\beta}}] = I(W).$$

(ii) For any fixed $\beta > E(K)$,

$$\liminf_{n \to \infty} \Pr[Z_n \geq 2^{-n^{\beta}}] = 1.$$
Le Theorem 1. Let $K^{(i)}_n$ be an $(l-t) \times (l-t)$ polarization kernel, obtained by $t$ times application of Alg. 1 to $l \times l$ kernel $K$. Let $r_i = (p_{r_0(0)}, p_{r_1(1)}, \ldots, p_{r_{l-1}(l-1)})$ be a sequence of removed columns indices, defined in (10). $\pi(j)$ is a permutation. Then, the kernel codes $C_{K^{(i)}_n}$, $i \in [l-t]$, are the same for any $\pi(j)$.

Proof. Lemma 1 implies that for any $\pi(j)$ the kernel codes $C_{K^{(i)}_n}$ are defined by (11), where the array $A = A^{(i)}$ is given by $\mathcal{P}$ and $\mathcal{P} = \{p_0, p_1, \ldots, p_{l-1}\}$.

One can shorten a kernel $K$ by consecutive application of Alg 1. Theorem 1 implies that the kernel codes, and, therefore, the polarization properties of the resulting kernel $K^{(i)}$ do not depend on the order in which we shorten the columns of $K$. Thus, we can consider the vector $p$ just as a set $\mathcal{P} = \{p_0, p_1, \ldots, p_{l-1}\}$, which is referred to as a shortened pattern.

Definition 2. Let $K$ be an $l \times l$ polarization kernel. Let $\mathcal{P} \subset [l]$, $|P| = t, 1 \leq |P| \leq l-2$. Thus, the kernel $s_{\mathcal{P}}(K)$ is called a shortened kernel of $K$ on coordinates $\mathcal{P}$ if any kernel code of $s_{\mathcal{P}}(K)$ is some shortened kernel code of $K$ on coordinates $\mathcal{P}$. More precisely, $C^{(i)}_{s_{\mathcal{P}}(K)}$, $i \in [l-t]$, is given by $s_{\mathcal{P}}(C^{(i)}_K)$. The array $A = A^{(i)}$, defined in (8), denotes the mapping of $s_{\mathcal{P}}(K)$ row indices $i \in [l-t]$ to row indices $i' \in [l]$ of $K$.

In addition, by $\mathcal{R}_{\mathcal{P}}(K) = \{r_0, r_1, \ldots, r_{l-1}\}$, where $r_i$ is defined by (11), we denote a set of removed rows indices from $K$ after shortening on coordinates $\mathcal{P}$.

B. Processing of shortened kernels

We have a given $l \times l$ polarization kernel $K$ together with some processing algorithm. We assume that it is given as a block box, i.e. it takes $u_0^{(i)}$ and $y_0^{(i-1)}$ as input and outputs the probability $W^{(i)}_1(u_0^{(i)}y_0^{(i-1)})$ of the kernel $K$. In this section we describe how to modify inputs $u_0^{(i)}$ and $y_0^{(i-1)}$ to obtain the probabilities $W^{(i)}_1(u_0^{(i)}y_0^{(i-1)})$ of shortened kernel $s_{\mathcal{P}}(K)$.

Let $\tilde{K}$ be an $l \times l$ polarization kernel obtained from kernel $K$ by application of equivalent additions, i.e. $\tilde{K} = TK$, where $T$ is an $l \times l$ upper triangular matrix. Let $\tilde{W}^{(i)}_1(u_0^{(i)}y_0^{(i-1)})$ be a probability $\tilde{u}$-th input symbol of kernel $\tilde{K}$. We have $\tilde{y}_0^{(i)} = u_0^{(i)}K = u_0^{(i)}\tilde{K}$, which implies $\tilde{y}_0^{(i)} = u_0^{(i)}T^{-1}$. Since $T^{-1}$ is also upper triangular, it is easy to verified that

$$\tilde{W}^{(i)}_1(u_0^{(i)}y_0^{(i-1)}) = W^{(i)}_1(u_0^{(i)}(T^{-1})(0)y_0^{(i-1)}),$$

where $(T^{-1})(0)$ is an $(\phi + 1) \times (\phi + 1)$ submatrix of $T^{-1}$, which contains its first $\phi + 1$ rows and columns.

We consider shortening of $K$ on columns $\mathcal{P}$. We start from computing such kernel $\tilde{K} = TK$ as $\tilde{K}[i,j] = 0$ for $j \in \mathcal{P}$ and $i \in ([l] \setminus \mathcal{R}_{\mathcal{P}}(K))$. Next, for a given phase $\phi \in [l-t]$, input symbols $u_0^{(i)}$ of $s_{\mathcal{P}}(K)$ and input LLRs $y_0^{(i-1)}$ we define an extended phase $\tilde{\psi}_0 = A^{(i)}_0$, an extended input vector $\tilde{u}_0^{(i)}$ and an extended channel output vector $\tilde{y}_0^{(i)}$. More specifically, $\tilde{u}_0^{(i)} = u_i$ for $i \in [\phi]$, $\tilde{y}_0^{(i)} = 0$ for $j \in \mathcal{R}_{\mathcal{P}}(K)$, and $\tilde{y}_0^{(i)} = y_i$ for $i \in [\phi]$, $\tilde{y}_0^{(i)} = w_0$, $W(0)w_0 = 1$, for $j \in \mathcal{R}_{\mathcal{P}}(K)$. That is, the probabilities of the shortened kernel are given by

$$\tilde{W}^{(i)}_1(u_0^{(i)}y_0^{(i-1)}) = W^{(i)}_1(\tilde{u}_0^{(i)} \cdot (T^{-1})(\psi)y_0^{(i-1)}),$$

where

Algorithm 1: ShortenSingleCoordinate($K$, $j$, $l$)

1. $K' \leftarrow K$, $a \leftarrow l-1$; // initialization
2. for $i$ from $l-1$ down to $0$ do
3. if $K[i,j] = 1$ then
4. // index of the last 1 in column
5. $a \leftarrow i$;
6. break;
7. Obtain $\tilde{K}$ by removing $a$th row and $j$th column from $\tilde{K}$;
8. return $\tilde{K}$;
IV. OPTIMIZING THE SHORTENING PATTERN

Error exponent of the shortened kernel can substantially vary for different shortening patterns. Our goal is to find such a shortening pattern that results in a shortened kernel with as large error exponent as possible.

A. Bounds on partial distances after shortening

Lemma 2. Let $K$ be an $l \times l$ polarization kernel with partial distances profile (PDP) $D$. Let $D'$ be a PDP of a shortened kernel $s_{(j)}(K)$, and $R_{(j)}(K) = \{a\}$. We have

\[
\begin{align*}
\mathcal{D}'_k & \geq \mathcal{D}_k, & 0 \leq k \leq a - 1, \\
\mathcal{D}'_k & = \mathcal{D}_k, & a \leq k \leq l - 2.
\end{align*}
\]

Proof. If follows directly from definition 3.

Lemma 3. For any $l \times l$ polarization kernel $K$ with PDP $D$ we have $D_i \leq \text{wt}(K[i],i) \leq l$.

Proof. Terms (16)–(17) follows from Lemmas 2 and 3. By Lemma 2 we have $\mathcal{D}'_k \geq \mathcal{D}_k$. Let $k \in [a]$. For $k \in ([a] \setminus X_{(j)}(K))$ we have $\text{wt}(K'[k]) = \text{wt}(K[k]) = \mathcal{D}_k$, thus, by Lemma 2 we have $\mathcal{D}_k \geq \mathcal{D}'_k$, which results in (18).

Application of Lemma 3 to the case of shortening on $t$ columns results in the following

Corollary 1. Consider an $l \times l$ polarization kernel $K$ with PDP $D$, such that $\text{wt}(K[i]) = \mathcal{D}_i$, $i \in [l]$. Thus, for partial distances $D_i$, $i \in [l - t]$ of $K' = s_{(j)}(K)$ we have

\[
\begin{align*}
\mathcal{D}'_k & = \mathcal{D}_k + 1, & 0 \leq k \leq l - 2, \\
\mathcal{D}'_k & \leq \mathcal{D}_k \leq \text{wt}(K'[k]), & k \in X_{(j)}(K) \\
\mathcal{D}'_k & = \mathcal{D}_k, & k \in ([a] \setminus X_{(j)}(K))
\end{align*}
\]

Algorithm 2: FindOptimalShortening($K,l,t$)

1. Compute PDP $D$ of $K$;
2. Transform $K$ into $K'$ with $\text{wt}(K'[i]) = D_i$ by equivalent additions;
3. $E^* = 0$; $P^* = \emptyset$; $l' \leftarrow l - t$;
4. for each $P$ in $B(l,t)$ do
5. Compute $s_P(K)$;
6. /* lower bound on error exponent */
7. $\check{E} \leftarrow \frac{1}{l} \sum_{i=l+1}^{l-1} \text{log}_2 D_i$;
8. if $\check{E} > E^*$ then
9. $E^* \leftarrow \check{E}; P^* \leftarrow P$;
10. end
11. for each $P$ in $B(l,t)$ do
12. /* Upper bound from Corollary 1 */
13. Compute $D_0^{-1}$, where
14. $D_i = \begin{cases} \text{wt}(s_P(K'[i])), & A_i \in X_P(K'), \\ \text{null}, & \text{otherwise}; \end{cases}$
15. $E \leftarrow \frac{1}{l} \sum_{i=0}^{l-1} \text{log}_2 D_i$;
16. if $E > E^*$ then
17. $E^* = E; P^* = P$;
18. return $P^*$;

To compute (21) one should run FindOptimalShortening($K,l,t$) described in Alg. 2. Note that the complexity of $s_P(K)$ computation is $O(n \cdot |P|)$ row additions. Thus, we propose to initially compute lower bound on the optimal error exponent, which is done at lines (15) of Alg. 2. This allows us to skip costly calculation of $s_P(K)$ PDP if the upper bound $E$ is less than current $E^*$. Moreover, at line (15) one could compute partial distances $D_i$ only for indices $i$ given in (20).

C. Shortening of some known kernels

In this work we consider MK polar codes of length $l \cdot 32$ for $17 \leq l \leq 31$. Recently, $K_{32}$ kernel with $E(K_{32}) = 0.522$ and $\mu(K_{32}) = 3.417$, together with efficient window processing algorithm was introduced in [3]. So we basically consider $K_{32} \otimes F_2$ polarizing transform, where $F_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ is an $2^t \times 2^t$ Arikan matrix. Therefore, to obtain codes of length $l \cdot 32$, $17 \leq l \leq 31$, we shorten either $K_{32}$ or $F_2$, which results in polarizing transforms $s_{P}(K_{32}) \otimes F_2$ or $K_{32} \otimes s_{P}(F_2)$. In addition, we shortened kernels $F_4$, $K_{16}$, which was also
proposed for window processing algorithm [8], and 32 × 32 BCH kernel $B_{32}$ proposed in [9].

Table I: Properties of the shortened polarization kernels

| $K$ | $F_2$ | $F_5$ | $K_{32}$ | $B_{32}$ | $F_1$ |
|-----|-----|-----|-------|-------|-----|
| $2^1$ | $110$ | $111$ | $1110$ | $11110$ | $1111$ |
| $2^2$ | $1110$ | $1111$ | $11110$ | $111110$ | $11111$ |
| $2^3$ | $11110$ | $11111$ | $111110$ | $1111110$ | $111111$ |
| $2^4$ | $111110$ | $111111$ | $1111110$ | $11111110$ | $1111111$ |
| $2^5$ | $1111110$ | $1111111$ | $11111110$ | $111111110$ | $11111111$ |

Table I presents the optimal shortening patterns and polarization properties for above mentioned kernels. The shortening patterns are provided in big-endian hexadecimal representation of the number $\sum_{p \in P} 2^p$. For comparison, we included the error exponents of BCH kernels, shortened by the greedy algorithm [4]. It can be observed that in many cases the proposed Alg. 2 results in shortened BCH kernels with higher error exponent compared with the greedy algorithm.

It turns out that different shortening patterns may result in polarization kernels with the same error exponent but different scaling exponents. Thus, for each kernel and shortening size we tried to find a shortening pattern resulting in the minimal scaling exponent. Note that the values $\mu^*$ from Table I are most likely not optimal. Unfortunately, explicit minimization of scaling exponent after shortening is an open problem.

V. NUMERICAL RESULTS

The performance of all codes was investigated for the case of AWGN channel with BPSK modulation. The sets of frozen symbols were obtained by method proposed in [23].

Fig. 1 presents the simulation results for MK polar subcodes (PSCs) [24], chained [25], shortened and punctured PSCs of different lengths with rate 1/2. All codes were decoded by successive cancellation list (SCL) decoder [26] with the list size 8. It can be observed that MK PSCs with $K_{32} \otimes s_{p^2}(F_3)$ provide approximately 0.2 dB gain compared with all polar code constructions based on Arikan kernel.

It is noticeable that MK PSCs of lengths $12 \cdot 64 = 24 \cdot 32 = 768$ and $13 \cdot 64 = 26 \cdot 32 = 832$ with $s_{p^2}(K_{16}) \otimes F_0$ has the same performance as the codes with $K_{32} \otimes s_{p^2}(F_3)$. This fact shows that kernels of size $l \leq 16$ and even $E < 0.5$ are useful for MK polar code construction.

It can be seen that for some lengths MK PSCs with $K_{32} \otimes s_{p^2}(F_3)$ significantly outperforms codes with $s_{p^2}(K_{32}) \otimes F_3$. To the best of our knowledge, there are no analytical tools for analysis of MK polar codes with arbitrary kernels. The problem of MK polar code construction, which includes the selection of kernels included in the transform and its permutation, is an open problem and out of scope of this paper.

Observe that one can obtain polar codes of considered lengths by shortening of the entire polar code with polarizing transform $K_{32} \otimes F_3$. Unfortunately, there are no efficient methods for shortening of polar codes with arbitrary large kernels. Moreover, there are no methods for efficient estimation of bit subchannels reliability for such codes. On the contrary, method [23] allows one to estimate the bit subchannel capacities of
MK polar codes with transform $K_{1}^{\otimes t_{1}} \otimes K_{2}^{\otimes t_{2}} \otimes ... \otimes K_{m}^{\otimes t_{m}}$, $t_{i} \geq 1$, if subchannel capacity function is provided for each kernel $K_{i}$, $i \in m$. To compare the performance of these two approaches, we included the simulation results for transforms $s_{p}(F_{3}) \otimes F_{5}$ and $s_{p}(F_{3}) \otimes F_{6}$. It can be observed from Fig. 1 that the performance of the shortened Arikan polar subcodes and polar subcodes with shortened Arikan kernels are similar.

Fig. 2 illustrates the performance of different (768, 384) polar codes decoded by SCL decoder with list size $L$. We report the results for $s_{p}(K_{32}) \otimes F_{5}$, $K_{24}$, $F_{8}$, $K_{3}$, and $T_{3}$, where $K_{24}$ is a $24 \times 24$ kernel and $T_{3}$ is a $3 \times 3$ kernel introduced in [22] and [15], respectively. We also include the results for descending (DES) asymmetric construction with 16-bit CRC and improved hybrid design of MK polar codes. It can be seen that MK polar code with the proposed shortened kernel $s_{p}(K_{32})$ outperforms all other code constructions decoded with the same list size ($L = 8$) and demonstrates the same performance under SCL with $L = 4$.

The $K_{24}$ kernel was obtained by recently introduced search algorithm [12], which can also be used to obtain polarization kernels which admit low processing complexity. However, for each desired size of the kernel, this method requires complicated search over different partial distances. Moreover, such kernels do not have a common structure, whereas the proposed shortening method can be used to obtain a kernel of an arbitrary size (less than base kernel) and shortened polarization kernels can be processed by the same algorithm.

VI. CONCLUSIONS

In this paper a shortening method for large polarization kernels was proposed. This algorithm uses lower and upper bounds on partial distances after shortening to quickly eliminate unsuitable shortening patterns. The algorithm results in a shortened polarization kernel with the highest error exponent if applied to kernels of size up to 32.

We demonstrated application of the proposed algorithm to some $16 \times 16$ and $32 \times 32$ kernels and obtained kernels of size $9$ to $31$. Polar codes based on the obtained kernels were shown to outperform shortened and punctured polar codes with Arikan kernel, and polar codes with small kernels.

ACKNOWLEDGEMENT

This work is partially supported by the Ministry of Science and Higher Education of Russian Federation, passport of goszadanie no. 2019-0898.

REFERENCES

[1] E. Arıkan, “Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels,” IEEE Trans. Inf. Theory, vol. 55, no. 7, pp. 3051–3073, July 2009.
[2] P. Trifonov, “Chained polar subcodes,” in Proceedings of 11th International ITG Conference on Systems, Communications and Coding, Hamburg, Germany, 2017, pp. 1–6.
[3] A. Cavatassi, T. Tonnellier, and W. J. Gross, “Asymmetric construction of low-latency and length-flexible polar codes,” in IEEE ICC, 2019.
[4] S. B. Korada, E. Sasoglu, and R. Urbanke, “Polar codes: Characterization of exponent, bounds, and constructions,” IEEE Transactions on Information Theory, vol. 56, no. 12, pp. 6253–6264, December 2010.
[5] A. Fazeli, S. H. Hassani, M. Mondelli, and A. Vardy, “Binary linear codes with optimal scaling: Polar codes with large kernels,” in Proceedings of IEEE Information Theory Workshop, 2018.
[6] N. Presman, O. Shapira, and S. Litsyn, “Mixed-kernels constructions of polar codes,” IEEE J. Sel. Areas Commun., vol. 34, no. 2, Feb. 2016.
[7] V. Bioglio, I. Land, and C. Conto, “Improved hybrid design of polar codes and multi-kernel polar codes,” in 2019 IEEE ISIT, 2019.
[8] G. Trofimuk and P. Trifonov, “Window processing of binary polarization kernels,” IEEE Trans. Commun., vol. 69, no. 7, pp. 4294–4305, 2021.
[9] E. Moskovskaya and P. Trifonov, “Design of bch polarization kernels with reduced processing complexity,” IEEE Communications Letters, vol. 24, no. 7, pp. 1383–1386, 2020.
[10] V. Miloslavskaya and P. Trifonov, “Sequential decoding of polar codes with arbitrary binary kernel,” in Proceedings of IEEE ITW, 2014.
[11] R. A. Morozov, “Convolutional polar kernels,” IEEE Transactions on Communications, vol. 70, no. 6, pp. 1752–1761, 2020.
[12] G. Trofimuk, “A search method for large polarization kernels,” in Proceedings of IEEE ISIT, 2021.
[13] P. Trifonov, “Recursive trellis processing of large polarization kernels,” in Proceedings of IEEE ISIT, 2021.
[14] H. Griesser and V. R. Sidorenko, “A posteriori probability decoding of nonsystematically encoded block codes,” Problems of Information Transmission, vol. 38, no. 3, 2002.
[15] V. Bioglio, G. Fabry, I. Land, and J. C. Belfiore, “Multi-kernel polar codes: Concept and design principles,” IEEE Transactions on Communications, vol. 68, no. 9, pp. 5350–5362, 2020.
[16] N. Presman, O. Shapira, S. Litsyn, T. Etzion, and A. Vardy, “Binary polarization kernels from code decompositions,” IEEE Transactions On Information Theory, vol. 61, no. 5, May 2015.
[17] H.-P. Lin, S. Lin, and K. A. Abdel-Ghaffar, “Linear and nonlinear binary kernels of polar codes of small dimensions with maximum exponents,” IEEE Trans. Inf. Theory, vol. 61, no. 10, October 2015.
[18] S. H. Hassani, K. Alshahi, and R. Urbanke, “Finite-length scaling for polar codes,” IEEE Trans. Inf. Theory, vol. 60, no. 10, October 2014.
[19] A. Fazeli and A. Vardy, “On the scaling exponent of binary polarization kernels,” in Proceedings of 52nd Annual Allerton Conference on Communication, Control and Computing, 2014, pp. 797 – 804.
[20] H. Pfister and R. Urbanke, “Near-optimal finite-length scaling for polar codes over large alphabets,” in Proceedings of IEEE ISIT, 2016.
[21] A. Fazeli, H. Hassani, M. Mondelli, and A. Vardy, “Binary linear codes with optimal scaling: Polar codes with large kernels,” IEEE Transactions on Information Theory, vol. 67, no. 9, September 2021.
[22] H. Yao, A. Fazeli, and A. Vardy, “Explicit polar codes with small scaling exponent,” in Proceedings of IEEE ISIT, Paris, France, 2019.
[23] P. Trifonov, “On construction of polar subcodes with large kernels,” in Proceedings of IEEE ISIT, Paris, France, 2019.
[24] P. Trifonov and G. Trofimuk, “A randomized construction of polar subcodes,” in Proceedings of IEEE ISIT, Aachen, Germany, 2017.
[25] P. Trifonov, “Randomized chained polar subcodes,” in Proceedings of IEEE WCNCW. Barcelona, Spain: IEEE, 2018, pp. 292–297.
[26] I. Tal and A. Vardy, “List decoding of polar codes,” IEEE Transactions On Information Theory, vol. 61, no. 5, pp. 2213–2226, May 2015.
[27] G. Trofimuk, “A search method for large polarization kernels,” arXiv:2101.10269, 2021.