A Puncturing Algorithm for Mixing 2-Kernel and 3-Kernel Polar Codes

Xiaojun Zhang(✉), Chen Chen, Jianming Cui, Geng Chen, and Hengzhong Li

Shandong University of Science and Technology,
Qingdao, Shandong, People's Republic of China
zhangxiaojun@sdust.edu.cn

Abstract. In this paper, a puncturing algorithm for mixing 2-kernel and 3-kernel polar codes is presented. The puncturing sequence is generated based on the capacity of channels and the upper bound of minimum block error probability for successive cancellation (SC) decoding. We use the capacity-zero puncturing model, the decoding algorithm of mother codes can still be adopted. An improved greedy algorithm of computing the maximization of the minimum distance is proposed to select the information set. The maximum number of punctured bits is limited to $[1, 2^{n-2})$ when the length of subcodes $M \in (2^{n-2} * 3, 2^n)$. Simulation results show that the block error rate based on the mixing kernels is better than that based on 2-kernel.

Keywords: Polar codes · Multi-kernel · Length-compatible · Puncturing

1 Introduction

The original polar codes, proposed by Arikan [1], are capacity-achieving linear block codes. It is recursively constructed by the binary kernel matrix $\begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$. However, the code length is limit to be $N = 2^n$. [2] has proved that $l \times l$ kernel is granted for the polarization effect, where $l > 2$. It code length is $N = l^n$ [3]. To achieved more flexible code length, [4] presents construction methods based on multi-kernel, $N = l_1^n \times l_2^m$. This method is still restricted to achieve arbitrary code length.

In general, puncturing [5] and shorten [9] is a technique to implement higher code rate and wide range of code lengths. [6] proposed a puncturing pattern. Since the punctured channels don't transmit the information, they are treated as zero-capacity channels (completely noisy channels). We call this category as the capacity-zero (C0) puncturing model. The pattern punctures bits from the frozen set, which limits the performance of puncturing. [7] proposed a low-complexity puncturing algorithm, which selects the first m bits of the output as puncturing bits. It gets a better performance of the block error rate (BLER) than LET and WCDMA by using the successive cancellation list (SCL) decoder. However, all

M. Jia et al. (Eds.): WiSATS 2019, LNICST 281, pp. 323–332, 2019.
https://doi.org/10.1007/978-3-030-19156-6_30

the above puncturing schemes are based on the mother codes $N = 2^n$, and the performance is worse when too many bits are punctured.

This paper proposes a puncturing algorithm based on polar construction mixing kernels of sizes 2 and 3, when the length of subcodes $M \in (2^{n-1}, 2^{n-2} * 3)$. We can puncture the bits from mother code $N = 2^{n-2} * 3$ and combine the original puncturing algorithms when $M \in (2^{n-2} * 3, 2^n)$, the maximum number of punctured bits is limited to $[1, 2^{n-2})$. It is proved that the puncturing code is still applicable to the original SCL decoder. For short code lengths, the minimum distance has a large impact on performance, the polarization effect is not crucial. Theoretically, the punctured table can be acquired by optimization of distance spectrum or minimum Hamming distance. It is discovered that the maximal minimum-distance spectrum can be acquired by using recursion [8]. We propose the puncturing algorithm to select the punctured bits and use the improved greedy algorithm to select the information bits.

The rest of this paper is organized as follows. In Sect. 2, a brief review is given to the mixing 2-kernel and 3-kernel polar code. The proposed puncturing method is presented in Sect. 3. Numerical results are provided in Sect. 4. Finally, Sect. 5 concludes the paper.

2 Preliminaries

Since the code length is $N = n_1 \times \cdots \times n_s$, the generator matrix is defined as $G_N \triangleq T_{n_1} \otimes \cdots \otimes T_{n_s}$, where $\otimes$ denote Kronecker product, n_i denote the dimension of kernel and $T_2 = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$, $T_3 = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$, the dimensions of kernels 2×2, 3×3 respectively. If we select the first bit as the information bit, then the largest minimum distance is 3. For the size of the information set is 2, the last two bits are selected and the minimum distance is 2. If we select all the bits as information bits, the minimum distance is 1. The minimum-distance spectrum is $S_{T_3} = (3, 2, 1)$. For example, because of the recursive operation of generator matrix, it is cleared that $N_6 = T_2 \otimes T_3 = \begin{bmatrix} T_3 & 0 \\ T_3 & T_3 \end{bmatrix}$. The minimum-distance spectrum $S_{T_6} = sort(T_3, T_3 + T_3)$, where $sort(x)$ denotes the sorting of vector x in the decreasing ordering. The information set A and the frozen set A^c are acquired by the greedy algorithm, which is based on maximize minimum-distance spectrum. The binary output vector $X = \{x_1, x_2, \cdots, x_N\}$ can be acquired form the binary output vector $U = \{u_1, u_2, \cdots, u_N\}$ with G_N, $X = UG_N$. As in the binary case, the indices of X are required to be reshuffled. For each stage $i > 1$, the permutation matrix P_{i} can be calculated as

$$P_i = (Q_i | Q_i + N_{i+1} | Q_i + 2N_{i+1} \cdots | Q_i + (N/N_{\text{i}+1})N_{i+1}) \quad (1)$$

Where Q_i is the canonical permutation, it is introduced in [4]. Assume that the $n_j \times n_j$ is the dimension of the j-th kernel of the Kronecker product, $N_{\text{i}} = \prod_{j=1}^{i-1} n_j$, the last stage $P_s = Q_s$. Then the output indices are acquired in the first stage by P_1. Then we can get the X permutation in the final, in Sect. 3 we will show an example in detail.

The reliability of output is calculated with Bhattacharyya parameter over BEC channels. The subchannels' capacity of T_3 is

$$\begin{cases} I(W_3^1)=I(W_1)I(W_2)I(W_3) \\ I(W_3^2)=I(W_1)+I(W_2)I(W_3)-I(W_1)I(W_2)I(W_3) \\ I(W_3^3)=I(W_1)+I(W_2)-I(W_1)I(W_2) \end{cases} \quad (2)$$

Multi-kernel polar code can calculate the error probability of subchannel by using Gaussian approximation (GA). In the case of binary additive white Gaussian noise channel (B-AWGNC), it assumes that all-zero bits are transmitted, and the probability density function (PDF) of likelihood ratios (LLRs) is $a_N^i \sim N(m_N^i, 2m_N^i)$, where $m_1^1 = 2/\sigma^2$. The recursive formulas of LLRs are calculated by

$$m_{3N}^{3i-1} = \varphi^{-1}(1-(1-m_N^i)^3) \quad (3)$$

$$m_{3N}^{3i-2} = \varphi^{-1}(1-(1-m_N^i)^2) + m_N^i \quad (4)$$

$$m_{3N}^{3i} = 2m_N^i \quad (5)$$

Where

$$\varphi(x) = \begin{cases} 1 - \frac{1}{\sqrt{4\pi x}} \int_{-\infty}^{\infty} \tanh\frac{u}{2} \cdot exp(-\frac{(u-x)^2}{4x})du & x \geq 10 \\ 0 & x = 0 \end{cases} \quad (6)$$

$\varphi(x)$ can be calculated by

$$\varphi(x) = \begin{cases} \sqrt{\frac{\pi}{x}(1-\frac{10}{7x})}exp(-\frac{x}{4}) & x \geq 10 \\ exp(-0.4527x^{-0.86}+0.00218) & 0 < x < 10 \end{cases} \quad (7)$$

Assume all-zero vectors are transmitted, the estimate of u is determined to 1 when the LLR is less than 0. The probability over $(-\infty, 0)$ is the error probability of SC decoding

$$p_e(i) = \int_{-\infty}^{0} \frac{1}{2\sqrt{\pi m_N^{(i)}}} \cdot exp(\frac{-(x-m_N^{(i)})^2}{4m_N^{(i)}})dx \quad (8)$$

3 Puncturing Algorithm for Mixing 2-Kernel and 3-Kernel Polar Codes

In this section, the paper analyzes the impact of each punctured bit on the other bits and the feasibility of puncturing algorithm for mixing 2-kernel and 3-kernel polar code. We propose the method of puncturing input bits for mixing of kernel polar codes, and apply the quasi-uniform puncturing (QUP) algorithm to the proposed mixing 2-kernel and 3-kernel construction method. After puncturing, the minimum-distance spectrum is used to select the information set A and the frozen set A^c.

Theorem 1. *Regardless of the puncturing from the input bits or the output bits, it can receive the same number of channel with zero-capacity at the other side by using* T_3 *kernel.*

Proof. For T_3, the input bits of LLRs with SC decoder are calculated by

$$L(u_1) = 2tanh^{-1}(tanh\frac{L(y_1)}{2}tanh\frac{L(y_2)}{2}tanh\frac{L(y_3)}{2}) \tag{9}$$

$$L(u_2) = 2tanh^{-1}(tanh\frac{L(y_2)}{2}tanh\frac{L(y_3)}{2})+(-1)^{\mathrm{u}_1}L(y_1) \tag{10}$$

$$L(u_3) = (-1)^{\mathrm{u}_1}L(y_2) + (-1)^{\mathrm{u}_1 \oplus \mathrm{u}_3}L(y_3) \tag{11}$$

If one bit is punctured, one of the output sequence y_1^3 does not have any message. We will puncture one bit and make one of $\{L(y_1); L(y_2); L(y_3)\}$ be 0. No matter what the situation is, that will lead to $L(u_1) = 0$. The capacity of subchannels can also be calculated by (2). When puncturing one bit from input bits, the result of $I(W_3^1) = 0$ remains no matter which one is chosen in $\{W_1, W_2, W_3\}$. For other stages, due to T_2, it can acquire the same number of channel with zero-capacity at both sides. In every stage, it can get the same number of channels with zero-capacity at the both side. And the one which is puncturing the first bit in T_3 as the optimal puncturing pattern. It can be noticed that the input bit u_1 corresponds exactly to the output bit y_1. When the subcodes $M \in (2^{n-1}, 2^{n-2} * 3)$, it just needs to puncture 2^{n-2} bits at most.

Theorem 2. *Suppose that the length of subcodes* $M \in (2^{n-1}, 2^{n-2} * 3)$*, the number of puncturing codes* m*. Puncturing the bits* $\{y_1, \cdots, y_m\}$ *from the output bits, the corresponding input bits must be the bits with zero-capacity.*

Proof. Assume that the error probability of the sub-channel is $p_e(i)$. The overall block error probability upper bound of polar code is obtained by

$$\text{BLER} \leq \sum_{i \in A} p_e(i) \tag{12}$$

Based on the minimum estimate of $\sum p_e(i)$, we can get the structure and the best performance. Therefore, when a bit in T_3 is punctured, we can select the optimal puncturing pattern which minimizes BLER. The punctured bits can be regarded as $L(y_i) \sim N(0, 0)$, where $E(x)$ denotes the mean of LLRs

$$E(u_1) = \varphi^{-1}(1 - (1 - E(y_1))(1 - E(y_2))(1 - E(y_3))) \tag{13}$$

$$E(u_2) = \varphi^{-1}(1 - (1 - E(y_2))(1 - E(y_3))) + E(y_1) \tag{14}$$

$$E(u_3) = E(y_1) + E(y_2) \tag{15}$$

Table 1 enlists the error probability of subchannels for T_3 after puncturing one bit. In case of puncturing the first bit or third bit of T_3, the minimum block error probability upper bound can be obtained.

Considering of the Theorem 1, we determine that the first bit in the T_3 is the unique puncturing position. The output sequence after polarization can be acquired using (1) when one bit is punctured. An example is given in Fig. 1.

Table 1. The error probability of subchannels for T_3 after puncturing one bit

Punctured bit	u_1	u_2	u_3	$Total$
u_1		0.2605	0.0786	0.3391
u_2	0.2606		0.1587	0.4193
u_3	0.2606	0.0786		0.3391

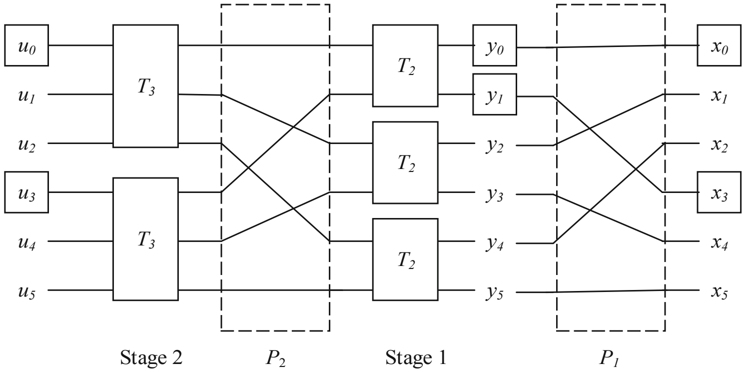

Fig. 1. Tanner graph of mixing 2-kernel and 3-kernel polar code with $N = 6$

Since stage 2 is the last stage, $P_2 = Q_2$, $Q_2 = \begin{pmatrix} 1\,2\,3\,4\,5\,6 \\ 1\,3\,5\,2\,4\,6 \end{pmatrix}$, we can get P_1 by all the other P_i, $P_1 = (P_2)^{-1} = \begin{pmatrix} 1\,2\,3\,4\,5\,6 \\ 1\,4\,2\,5\,3\,6 \end{pmatrix}$. The output order can be calculated with P_1. The polar code for $N = 6, M = 4, m = 2$, the puncturing vector $\mathbf{P} = (p_1, p_2, p_3, p_4, p_5, p_6)$, where $p_i \in \{0, 1\}$ denotes the corresponding output bits. $\mathbf{P}$ is set to all ones initially, the output x_i is punctured when $p_i = 0$. The algorithm is described as:

Step1: Set the first m bits as zeros of $\mathbf{P}$.
Step2: Update permutation by P_1 and obtain the puncturing set of input bits, B.

We find that the bits, which is the first bit of each T_3, are already in the first third of the output sequence. So any bits in the first third of output bits can be punctured. If the punctured bits from output sequence have zero-capacity, the corresponding input bits must be the bits with zero-capacity calculating by (2).

After puncturing the bit of T_3, the generator matrix is $\begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}$, the minimum-distance spectrum is $S'_{T_3} = (2, 1, 0)$, the corresponding position of input bits are $\{u_2; u_2, u_3; \phi\}$. If we delete columns corresponding to the punctured positions from, $G_N = T_2^{\otimes n} \otimes T_3$ and the vector $r'_N = (2, 1)^{\otimes n} \otimes S'_{T_3}$, where r'_N is an unsorted version of the minimum-distance spectrum. We set $I = \{\}$, at each step we will select one row add to I.

Algorithm 1. Information set to minimum distance

```
Initialize the set I = {}
Load N-vector r_N
Load B
for i = 1, ..., K do
    l = argmax(r_N)
    q = (l mod 3)
    r_N(l) = 0;
    if q == 1 then
        if r_N(l + 2) == 0 then
            I = I ∪ {l + 2}
        else
            I = (I\{l, l + 1, l + 2}) ∪ {l}
        end if
    else if q == 2 then
        if r_N(l + 1) == 0 then
            I = I ∪ l
        else
            I = (I\{l − 1, l, l + 1}) ∪ {l, l + 1}
        end if
    else
        I = (I\{l − 2, l − 1, l}) ∪ {l − 2, l − 1, l}
    end if
end for
```

For instance, $M = 5$ and $K = 3$. When $M \in (2^2, 2^1 * 3)$, we will use the puncturing algorithm for mixing 2-kernel and 3-kernel polar code. When the code lengths is $N = 2^1 * 3^1 = 6$, the puncturing vector is initialized as $\mathbf{P} = (1, 1, 1, 1, 1, 1)$. After the first step and reshuffling permutations, $\mathbf{P} = (0, 1, 1, 1, 1, 1)$, the punctured position is u_1. The generator matrix G_6 is depicted as

$$G_6 = \begin{bmatrix} 0&0&0&0&0&0 \\ 0&0&1&0&0&0 \\ 0&1&1&0&0&0 \\ 0&1&1&1&1&1 \\ 0&0&1&1&0&1 \\ 0&1&1&0&1&1 \end{bmatrix} \tag{16}$$

The sequence of the minimum-distance spectrum is $S_{T_6} = (5, 3, 2, 2, 1, 0)$. The information set I is initially empty.

Step1: $l = \{4\}$, $q = 1$,$r_N(6) \neq 0$, $I = 4$.
Step2: $l = 5$, $q = 2$,$r_N(6) \neq 0$, $I = \{5,6\}$.
Step3: $l = 1$, $q = 1$,$r_N(3) = 0$, $I = \{3,5,6\}$.

4 Numerical Results

When $M \in (2^{n-1}, 2^n)$, the method of this paper combines with the QUP algorithm based on the original polar code, which can effectively reduce the number of punctured bits, and improve the decoding performance. For instance, when $M \in [128, 192]$, the number of punctured bits based on proposed algorithm are represented in Fig. 2(a). And Fig. 2(b) shows the number of punctured bits based on the original polar codes. It shows that the proposed algorithm less 64 bits than that based on the original polar codes.

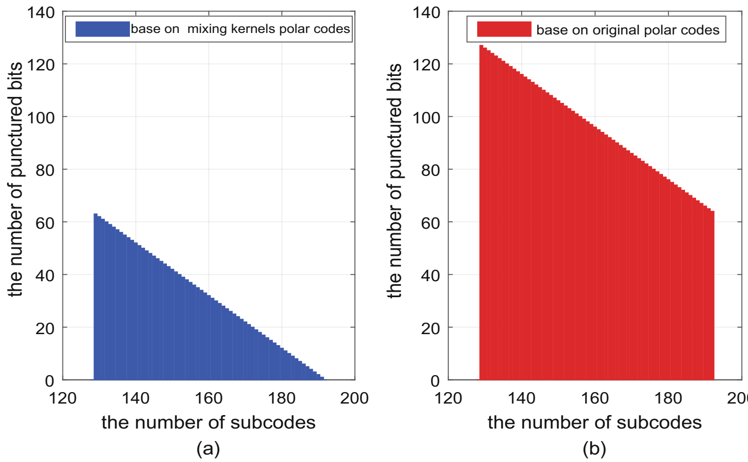

Fig. 2. The number of punctured bits needed for different subcodes

When $M \in (2^{n-1}, 2^{n-2} * 3)$, the proposed method effectively reduces the punctured bits based on the mother codes being equal to 1/4 of the original codes. When $M \in (2^n * 3, 2^{n+2})$, it can still puncture the bits based on original polar code. Because of the above improvements, the research can limit the maximal number of punctured bits to 2^{n-1}. Moreover, compared with the original polar code puncturing algorithm based on the 2-kernel polar code, the decoding based on the mother decoder, and complexity of the multi-kernel polar code decoder is lower than that of the original polar code. Therefore, the proposed algorithm has lower complexity compared to the decoder and encoder of the state-of-the-art punctured polar code.

In Figs. 3 and 4, the paper compares BLER performance of the proposed puncturing algorithm over AWGN channel against the punctured and shortened polar code based on the original polar code. We adopt the SCL decoding with $L = 8$. When $M = 191$ and dimension $K = 96$, the mother code based on multi-kernel of length $N = 192$, and others use mother code based on the original polar code of length $N = 256$.

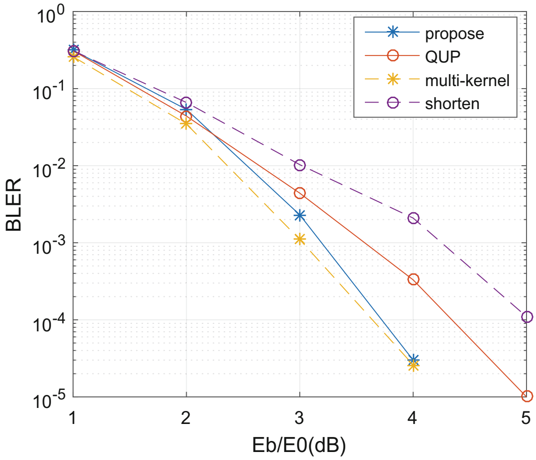

Fig. 3. Block error rates for length $N = 191$ and $K = 96$ under SCL decoding with list size $L = 8$

Figure 3 shows that the performance of $M = 191$, $K = 96$ punctured codes. The proposed outperform the other codes. The proposed code provides a gain of about 0.55 dB at BLER of 10^{-4}.

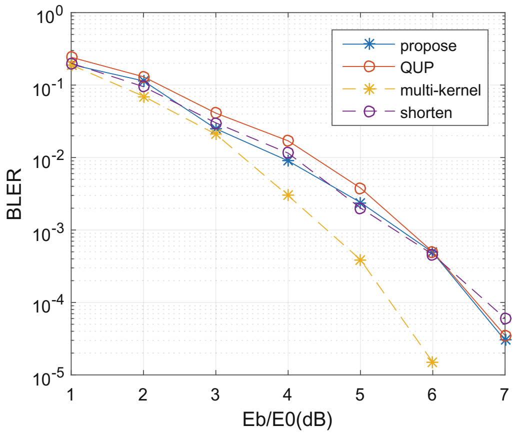

Fig. 4. Block error rates for length $N = 40$ and $K = 20$ under SCL decoding with list size $L = 8$

Figure 4 depicts that the performance of a code of length $M = 40$ and dimension $K = 20$. The BLER of the proposed is close to that of the shorten polar code. This shows that the proposed algorithm should be further optimized in adopting larger-scale and varied kernel not only use T_3 to construct the multi-kernel. The proposed puncturing algorithm based on multi-kernel significantly outperforms QUP puncturing algorithm which is based on original 2-kernel polar code. It is expected that this heuristic algorithm can apply puncturing algorithm to multi-kernel polar code, when larger puncturing ratio in practical applications is required.

5 Conclusions

In this paper, we present a puncturing algorithm for mixing 2-kernel and 3-kernel polar code. When $M \in (2^{n-1}, 2^{n-2} * 3)$, the proposed algorithm for mixing codes can puncture less 2^{n-2} bits than the others. We further improve the greedy algorithm, which is suitable for puncturing algorithm, to find the information set. Simulation results show that the proposed algorithm can have similar or even better performance than that based on 2-kernel polar code.

Acknowledgment. This work was supported in part by the NSFC (61701284, 61472229, 61471224), Project funded by China Postdoctoral Science Foundation (2016M592216) and Qingdao postdoctoral research project (2016125), Sci. & Tech. Development Fund of Shandong Province of China (2016ZDJS02A11), and SDUST Research Fund (2015TDJH102). 2017 Humanities and Social Sciences Research Program of University in Shandong (J17ZZ27).

References

1. Arikan, E.: Channel polarization: a method for constructing capacity-achieving codes for symmetric binary-input memoryless channels. IEEE Press **55**(7), 3051–3073 (2009)
2. Korada, S.-B., Sasoglu, E., Urbanke, R.: Polar codes: characterization of exponent, bounds, and constructions. IEEE Trans. Inf. Theory **56**(12), 3253–6264 (2010)
3. Wang, X., Zhang, Z., Zhang, L.: On the SC decoder for any polar code of length $N = l^n$. In: Wireless Communications and Networking Conference, pp. 485–489. IEEE, Istanbul (2014)
4. Gabry, F., Bioglio, V., Land, I., Belfiore, J.C.: Multi-kernel construction of polar codes. In: 2017 IEEE International Conference on Communications Workshops (ICC Workshops), pp. 761–765. IEEE, Paris (2017)
5. Eslami, A., Pishro-Nik, H.: A practical approach to polar codes. In: IEEE International Symposium on Information Theory Proceedings, pp. 16–20. IEEE, St. Petersburg (2011)
6. Zhang, L., Zhang, Z., Wang, X., Yu, Q.: On the puncturing patterns for punctured polar codes. In: IEEE International Symposium on Information Theory, pp. 121–125. IEEE, Honolulu (2014)
7. Niu, K., Chen, K., Lin, J.-R.: Beyond turbo codes: rate-compatible punctured polar codes. In: 2013 IEEE International Conference on Communications (ICC), pp. 3423–3427. IEEE, Budapest (2013)

8. Bioglio, V., Gabry, F., Land, I., Belfiore, J.-C.: Minimum-distance based construction of multi-kernel polar codes. In: 2017 IEEE Global Communications Conference, GLOBECOM 2017, pp. 1–6 (2016). IEEE Global Communications Conference 18(12), 2081–2084
9. Wang, R., Liu, R.: A novel puncturing scheme for polar codes. IEEE Commun. Lett. **18**(12), 2081–2084 (2016)