

A Modified Approach to Punctured Product Polar Codes

Mohammad Abu Hanif and Sina Vafi

College of Engineering, IT and Environment, Charles Darwin University, Darwin, Australia

<https://doi.org/10.26636/jtit.2019.132219>

Abstract—A high-performance polar code introduced as a product polar code can be constructed by concatenating two short length polar codes. The punctured structure of this code was achieved by puncturing one of its constituent codes. The constructed polar code provided better performance than a single polar code in the error floor region. However, its performance in the waterfall region was not as good as that of single polar codes. This paper proposes a new puncturing algorithm for product codes constructed by two identical polar codes. Puncturing is conducted on both constituent codes, to ensure that the new code outperforms the previously punctured product polar and single polar codes. This is evident in both waterfall and error floor regions.

Keywords—polar codes, product codes, punctured codes, iterative decoding.

1. Introduction

Polar codes are recently invented promising forward error correction (FEC) codes [1], which are well known for their low complexity in encoding and decoding. However, the lengths of codes are not flexible as they are in order of 2^n ($n = 1, 2, \dots$). This limits their application in video services and control-based systems, which usually require unequal error protection in their structure. Similar to other linear block codes, obtaining an arbitrary length for polar codes is possible by puncturing a number of bits from its original codeword.

Several puncturing methods were proposed, which are mainly applicable for single polar codes (SPC). In one of the methods, an exhaustive search is performed to delete rows and columns that maximize the exponent of the reduced generator matrix \mathbf{G} [2]. This technique offers better performance compared to random puncturing. However, implementation of an exhaustive search algorithm for determining a good polarized matrix still remains a time-consuming process. Another class of punctured polar codes was proposed in [3], where the first η_p unreliable bits in the bit-reversal order are punctured. An improvement is presented by Wang *et al.* [4], where a puncturing pattern is obtained based on the column weights of the generation matrix. In [5], puncturing and shortening sets are formed based on bit-reversal permutation of the most unreliable bits. Similarly, a technique was proposed to ensure short polar codes, which do not select consecutive coded bits

as punctured ones [6]. Finally, limitations of the above-mentioned techniques are overcome by puncturing bits at different levels of the code [7].

In general, performance of SPCs is limited due to the existence of a low minimum weight, or a relatively high minimum weight with high multiplicity. As a solution, product polar codes (PPCs) were proposed, which are constructed by the serial combination of two half-rate polar codes [8]. These codes generate the same minimum weight but with a lower multiplicity compared to SPCs with rate 1. It is also shown that with the same code rates and lengths, punctured PPCs provide better performance than punctured SPCs. The improvement is mainly evident in the medium to high signal-to-noise ratios. In the proposed method, puncturing of the product code is only conducted at one of its constituent codes. Low performance of PPCs in the waterfall region is mainly related to the weak performance of their punctured constituent codes [8].

This paper presents an algorithm, which improves the performance of punctured product polar codes, where their two constituent codes are punctured. In this way, bits having the highest error probability are punctured from the constituent codes. This ensures that the most unreliable bits are punctured for the product polar code and allows its constituent decoders to efficiently recover the original information.

The rest of the paper is organized as follows. A brief introduction of polar codes is given in Section 2. In Section 3, structure of product polar codes and a new puncturing technique applied for both constituent codes are presented. The effects that different puncturing techniques exert on iterative decoding are analyzed in Section 4. Numerical results are given in Section 5. Finally, Section 6 concludes the paper.

2. Polar Codes

The generator matrix, of a (N, K, d_{min}) polar code with length N , dimension K and minimum weight (d_{min}) is formed by taking n -th, $n = \log_2(N)$ Kronecker power of a binary matrix $B = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$, which is given by $G_N = B^{\otimes n}$. Codewords of these codes are formed by multiplying a message vector of length K with a sub-matrix of G_N , which includes only K rows of the generator matrix (G_K).

These rows are selected based on the reliability of N -th bit-channel. For any binary discrete memoryless channel, the reliability of N -th bit-channels can be recursively constructed by [1]:

$$Z(w_{2i}^j) = \begin{cases} 2Z(w_i^j) - Z(w_i^j)^2 & \text{if } 1 \leq j \leq i \\ Z(w_i^{j-i})^2 & \text{if } i+1 \leq j \leq 2i \end{cases}, \quad (1)$$

where i is a positive integer and in order of 2 ($1 \leq i \leq 2^{n-1}$) and $Z(w_{2i}^j)$ denotes the Bhattacharyya parameter of j -th bit-channel at stage $2i$ ($2i \leq N$). The code construction method shown in Eq. (1) was initially designed for the binary erasure channel (BEC), when the equation is initialized by $Z(w_1^1) = \frac{1}{2}$. For the binary additive white Gaussian noise (AWGN) channel, with zero mean and variance $\frac{N_0}{2}$, $Z(w_1^1)$ can be given by [9], [10]:

$$Z(w_1^1) = e^{-\frac{E_c}{N_0}}, \quad (2)$$

where $E_c = RE_b$, R is the rate of the code and E_b is the energy spent per bits.

The smallest $Z(w)$ value obtained from Eq. (1) refers to the most reliable bit-channel, known as a good bit-channel. Let A_I represents a set of good bit-channels. The remaining bits are known as bad bit-channels and are denoted as elements of A_I^c . Good channels are suitable for carrying information, and the rest are recognized as frozen bits.

Let $\mathbf{u} = (u_1 u_2 \dots u_K)$ be a message vector of length K and $\mathbf{x} = (x_1 x_2 \dots x_N)$ be the corresponding coded sequence, which is generated by $\mathbf{x} = \mathbf{u}\mathbf{G}_K$, $K \in \{A_I\}$. This paper follows the systematic encoding and decoding proposed by Arikan in [13]. The d_{min} of a systematic polar code is determined based on weights of its rows, whose indices are elements of A_I . This is given by [14], [15]:

$$d_{min}(C^{SPC}) = \min_{b \in A_I} \{2^{\omega(b)}\}, \quad (3)$$

where $\omega(b)$ is the number of ones in the binary expansion of b .

Based on an exhaustive search algorithm, it is proven that some polar codes have a low minimum weight compared to their length. On the other hand, some codes have high multiplicity for their relatively high minimum weights. These specifications deteriorate performance of codes at

medium to high signal-to-noise ratios, which is mainly dominated by their minimum weight specifications. By contrast, with the same code rate and length, it is possible to construct product codes whose minimum weights have lower multiplicity than those obtained from single polar codes [8].

3. Product Polar Codes (PPCs)

In (N, K) product polar codes with rate R , every message with length $K = K_1 \times K_2$ is converted to K_2 blocks with length K_1 and encoded by the half rate ($R_1 = \frac{K_1}{N_1} = \frac{1}{2}$) outer (N_1, K_1, d_{min_1}) polar code. These codewords ($\mathbf{x}_{K_2 \times N_1}^1$) are then encoded by another half rate inner (N_2, K_2, d_{min_2}) polar code to form a two-dimensional codeword matrix ($\mathbf{x}_{N_1 \times N_2}$) of (N, K, d_{min}) product polar codes with $R = \frac{1}{4}$, where $N = N_1 \times N_2$ and $d_{min} = d_{min_1} \times d_{min_2}$.

Interactive iterative decoding is used to decode the information received from the channel. The basic structure of iterative decoding of PPCs is shown in Fig. 1. Here, $\mathbf{y} = [\mathbf{y}_1 \mathbf{y}_2 \dots \mathbf{y}_{N_1}]^T$ denotes the noisy received sequence, where $\mathbf{y}_l = [y_{l,1} y_{l,2}, \dots, y_{l,N_2}]$ is the received block at time l . In the figure, $L_{em}(\hat{\mathbf{x}}_l)$ s is the extrinsic information of the estimated codeword ($\hat{\mathbf{x}}_l$) delivered by the m -th decoder ($m = 1, 2$). At t -th iteration, log-likelihood ratios (LLRs) of the decoders are determined by:

$$L_1^{(t)}(\hat{\mathbf{x}}_\alpha) = Le_2^{\pi(t-1)}(\hat{\mathbf{x}}_\alpha) + \mathbf{y}_\alpha \quad (4)$$

$$L_1^{(t)}(\hat{\mathbf{x}}_\beta) = Le_1^{(t-1)}(\hat{\mathbf{x}}_\beta)$$

$$L_2^{(t)}(\hat{\mathbf{x}}_\alpha) = Le_1^{\pi^{-1}(t)}(\hat{\mathbf{x}}_\alpha) + \mathbf{y}_\alpha^{\pi^{-1}},$$

where $\alpha \in \{A_I\}$, $\beta \in \{A_I^c\}$ and $Le_2^{(0)}(\hat{\mathbf{x}}_\alpha) = \mathbf{0}$. Here, $(o)^\pi$ and $(o)^{\pi^{-1}}$ are the interleaved and de-interleaved information of (o) . Note that a conventional concatenation technique is followed in this paper, where the inner code cannot be smaller than the outer code. This means $N_2 \geq N_1$ and $K_2 \geq K_1$.

Although product polar codes reduce the multiplicity of d_{min} , they are represented as low rate codes. In order to construct product codes with a rate greater than $\frac{1}{4}$, a proper puncturing technique is required to maintain a high level of error correcting capability.

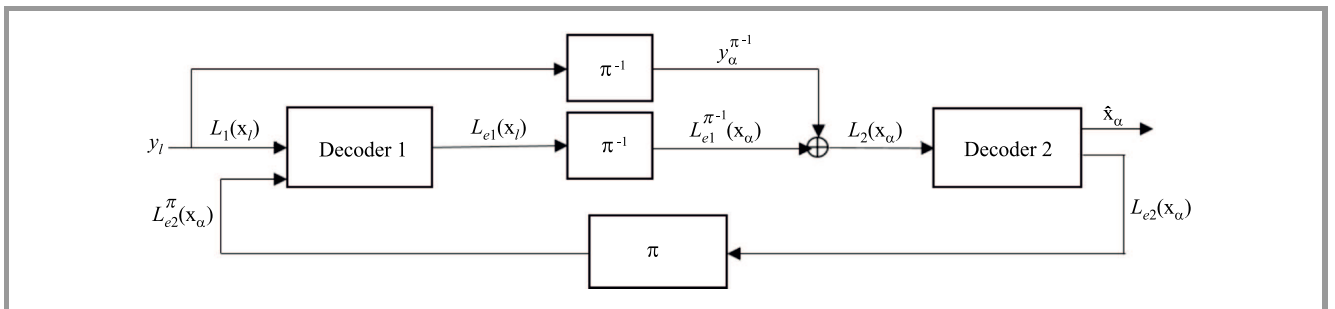


Fig. 1. Decoder for a two-dimensional product polar code.

3.1. Puncturing of Product Polar Codes

The arbitrary lengths of product polar codes can be obtained by puncturing a number of bits from codewords. Puncturing can only be performed at the inner code. In this case, codewords of the punctured product polar code are constructed by deleting n_ϕ rows and columns from the generator matrix of (N_2, K_2, d_{min_2}) code [8]. This concludes a $(N_1(N_2 - n_\phi), K_1.K_2)$ code, whose minimum weight (d'_{min}) is upper bounded with the minimum weight of $d_{min_1} \cdot (d_{min_2} - (n_1)_{n_\phi})$, where $(n_1)_{n_\phi}$ gives the number of punctured bits with the value of 1. Obviously, punctured bits are the most unreliable bits of (N_2, K_2, d_{min_2}) polar code, which are also elements of A_l^c .

Example 1: A (64, 16) product polar code can be formed by concatenating two (8, 4, 4) polar codes, $N_1 = N_2 = 8$ and $K_1 = K_2 = 4$.

Let

$$\mathbf{u} = \begin{pmatrix} u_{1,1} & u_{1,2} & u_{1,3} & u_{1,4} \\ u_{2,1} & u_{2,2} & u_{2,3} & u_{2,4} \\ u_{3,1} & u_{3,2} & u_{3,3} & u_{3,4} \\ u_{4,1} & u_{4,2} & u_{4,3} & u_{4,4} \end{pmatrix}$$

be the message with length K_1 , which is encoded by the outer (8, 4, 4) polar code. Its two-dimensional codeword matrix is given by:

$$\mathbf{x}^1 = \begin{pmatrix} p_{1,1} & p_{1,2} & p_{1,3} & u_{1,1} & p_{1,4} & u_{1,2} & u_{1,3} & u_{1,4} \\ p_{2,1} & p_{2,2} & p_{2,3} & u_{2,1} & p_{2,4} & u_{2,2} & u_{2,3} & u_{2,4} \\ p_{3,1} & p_{3,2} & p_{3,3} & u_{3,1} & p_{3,4} & u_{3,2} & u_{3,3} & u_{3,4} \\ p_{4,1} & p_{4,2} & p_{4,3} & u_{4,1} & p_{4,4} & u_{4,2} & u_{4,3} & u_{4,4} \end{pmatrix}, \quad (5)$$

where $p_{i,j}$ are the parity bits and $i, j \in \{A_l^c\}$. The \mathbf{x}^1 is interleaved and encoded by the inner code. The inner code is punctured by $\eta_\phi = 2$ bits. As the first and fifth bit-channels, according to Eq. (1), are the most unreliable bits amongst the eight bit-channels of (8, 4, 4) the polar code, every message of \mathbf{x}^1 is encoded by removing first and fifth rows and columns of \mathbf{G}_8 matrix. This concludes a two-dimensionally punctured codeword presented in below:

$$\mathbf{x} = \begin{pmatrix} \times & \times & \times & \times & \times & \times & \times & \times \\ p'_{2,1} & p'_{2,2} & p'_{2,3} & p'_{2,4} & p'_{2,5} & p'_{2,6} & p'_{2,7} & p'_{2,8} \\ p_{3,1} & p_{3,2} & p_{3,3} & p_{3,4} & p_{3,5} & p_{3,6} & p_{3,7} & p_{3,8} \\ p_{1,1} & p_{1,2} & p_{1,3} & u_{1,1} & p_{1,4} & u_{1,2} & u_{1,3} & u_{1,4} \\ \times & \times & \times & \times & \times & \times & \times & \times \\ p_{2,1} & p_{2,2} & p_{2,3} & u_{2,1} & p_{2,4} & u_{2,2} & u_{2,3} & u_{2,4} \\ p_{3,1} & p_{3,2} & p_{3,3} & u_{3,1} & p_{3,4} & u_{3,2} & u_{3,3} & u_{3,4} \\ p_{4,1} & p_{4,2} & p_{4,3} & u_{4,1} & p_{4,4} & u_{4,2} & u_{4,3} & u_{4,4} \end{pmatrix}, \quad (6)$$

where $p'_{i,j}$, $i, j \in \{A_l^c\}$ are the parity bits produced by the inner code. Indeed, all 8 codewords expressing columns of the matrix are punctured by two bits. This sums up

a (48, 16) product polar code formed by combining (8, 4, 4) and (6, 4, 4) polar codes.

3.2. The Modified Approach

A new puncturing scheme for the product polar code is constructed by puncturing both constituent codes. Indeed, a $(N', K_1.K_2)$ punctured code is formed, where $N' = (N_1 - n'_{\phi_1})(N_2 - n'_{\phi_2})$. In this case, n'_{ϕ_1} and n'_{ϕ_2} are the numbers of unreliable bits punctured from (N_1, K_1, d_{min_1}) and (N_2, K_2, d_{min_2}) polar codes, respectively.

Figure 2 shows steps in the process of construction of (49, 16) product polar code formed by two punctured (8, 4, 4) polar codes. The bit-channel one is the most unreliable amongst the eight channels of (8, 4, 4) polar code. Therefore, encoding at the outer code is performed by removing the first row and column of its generator matrix. Then, every message of \mathbf{u} (as shown in Fig. 2) is encoded to form a punctured codeword of the outer code. As bits in the first column are already punctured, there are $l-1$ ($l = N_1$) bits remaining in each codeword.

Let \mathbf{x}^1 be the matrix formed by punctured codewords of the outer code. This matrix is interleaved and each of its columns is used as the message for the inner code. Hence, a two-dimensional codeword matrix (\mathbf{x}) at the output of inner encoder is obtained. One may observe that only one bit from $l-1 = 7$ codewords of inner (8, 4, 4) code is punctured. Again, for this code, the first bit of the codeword is the most unreliable one. Therefore, bits positioned at the first row of \mathbf{x} are punctured to conclude a (49, 16) code.

The minimum weights (d''_{min}) of these punctured codes are given by $(d_{min_1} - (n'_1)_{n'_{\phi_1}}) \cdot (d_{min_2} - (n'_2)_{n'_{\phi_2}})$, where $(n'_1)_{n'_{\phi_1}}$ and $(n'_2)_{n'_{\phi_2}}$ are numbers of bits (with the value of 1) punctured from (N_1, K_1, d_{min_1}) and (N_2, K_2, d_{min_2}) polar codes, respectively. If $(n)_{n_\phi} = (n'_1)_{n'_{\phi_1}} + (n'_2)_{n'_{\phi_2}}$ and $(n'_1)_{n'_{\phi_1}} = (n'_2)_{n'_{\phi_2}}$, $d''_{min} = d'_{min} + (n'_1)_{n'_{\phi_1}} \cdot (n'_2)_{n'_{\phi_2}}$. Obviously, $d''_{min} > d'_{min}$. This shows that the product code punctured on the basis of two constituent codes has better performance than codes whose puncturing is only conducted on one of their constituent codes. Based on this analysis, improvement is primarily expected at the medium to high signal-to-noise ratios, where the codes' performance is dominated by their minimum weight specifications.

4. Puncturing Effect on the Iterative Decoding of PPCs

The effect of puncturing is verified through the iterative decoding process. In each iteration, LLRs of the information are updated and fed back to the input of the decoder until a maximum number of iterations (ite_{max}) is reached. At t -th iteration ($t \leq ite_{max}$), the iterative decoding process can be terminated if the following assumptions on the extrinsic information at the output of two decoders are satisfied [11]:

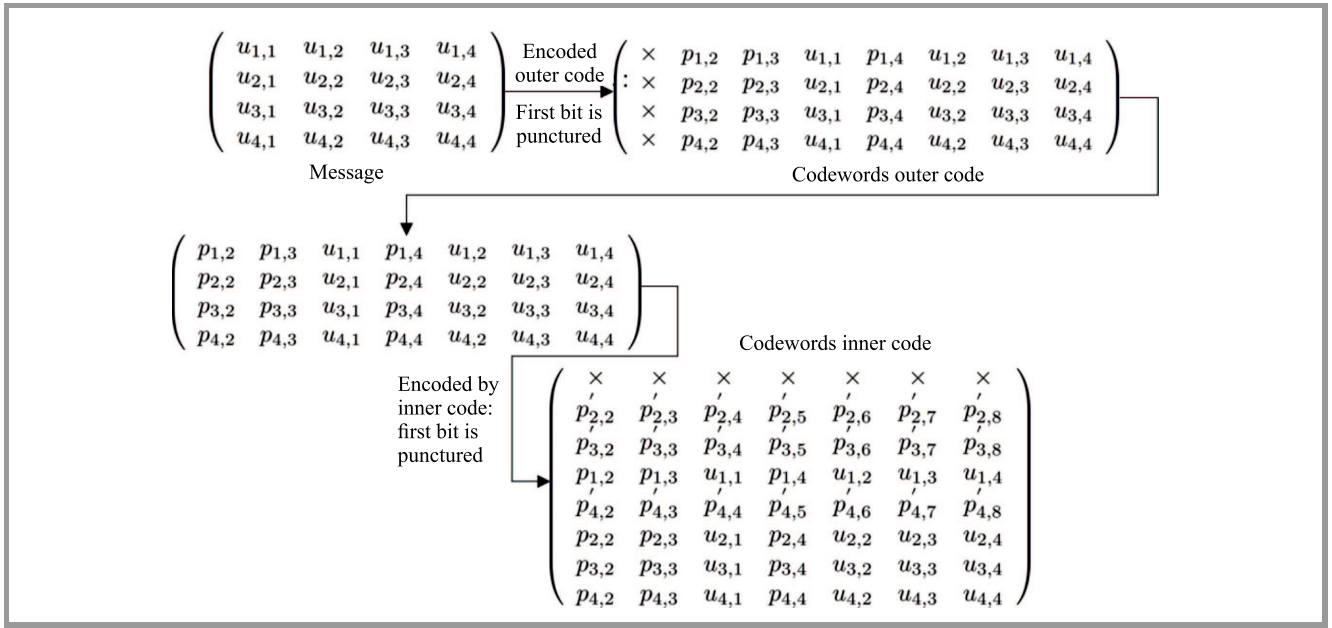


Fig. 2. Puncturing of two (8, 4, 4) polar codes for constructing a (49, 16) PPC.

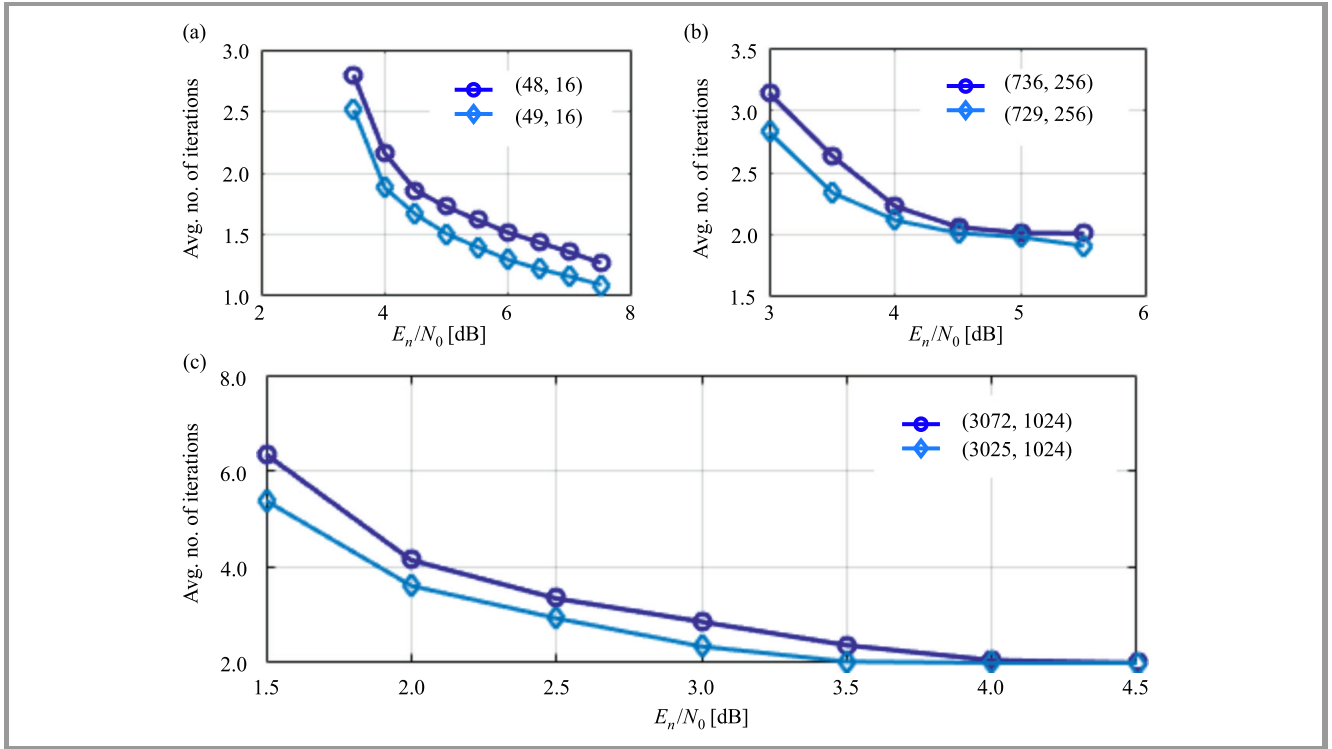


Fig. 3. Average number of iterations for punctured product polar codes.

- The polarities of soft-output information obtained from both decoders are not changed at two consecutive iterations. This is done by checking the sign of LLRs of two decoders, which is given by:

$$\begin{aligned} \text{sign} \left(L_{e1}^{(t)}(\hat{x}_{i,j}) \right) &= \text{sign} \left(L_{e2}^{(t)}(\hat{x}_{i,j}) \right) \\ &= \hat{x}_{i,j}^{(t)} = [\pm 1], \end{aligned} \quad (7)$$

where $\hat{x}_{i,j}$ denotes the estimate value of $x_{i,j}$ and $i, j \in \{A_I\}$.

- The difference between the magnitudes of the extrinsic information of a decoder input and output is small.

Figure 3 shows number of iterations applied for iterative decoding of different polar codes. In simulations, AWGN

channel is modeled and the code is modulated by binary phase shift keying (BPSK). The pipeline decoder with the maximum of 60 iterations is constructed for the decoding of constituent polar codes. Five iterations are also applied in process of interactive iterative decoding between the two decoders.

For (48, 16) and (49, 16) PPCs, puncturing is conducted on one and two of their constituent codes, respectively. In this case, two bits from each of eight codewords are punctured for (48, 16) code, while only one bit of seven codewords from each of (8, 4, 4) codes is punctured for the (49, 16) code. As the (49, 16) product code applied a lower number of punctured bits in its codewords, a weaker impact of the punctured bits on the LLR calculation of the information bits is observed. While the most unreliable bits are punctured from the inner code of (48, 16) PPC, some bits recognized as reliable bits for the outer code also punctured. Due to the removal of these reliable bits, interactive iterative decoding does not provide a good error correcting capability. Hence, decoding of this code is conducted by a higher number of iterations than in the case of the (49, 16) code.

Similarly, (736, 256) and (729, 256) PPCs are formed by puncturing one and both of their constituent codes, respectively. In this case, $N_1 = N_2 = 32$, $n_\phi = 9$ and $n'_{\phi 1} = n'_{\phi 2} = 5$. This means that nine bits are punctured from each of 32 codewords, when one constituent code is only punctured. Alternatively, five bits are punctured from 27 codewords, when both of the constituent codes are punctured. Although a higher number of bits are punctured to obtain the (729, 256) code, its iterative decoder applies a lower number of iterations than that of the (736, 256) code. Moreover, 16 bits are punctured from each of 64 codewords of the (3072, 1024) code, while nine bits are punctured from 55 codewords of the (3025, 1024) code. As shown in the figure, a lower number of iterations for the (3025, 1024) code is applied compared to (3072, 1024) at $\frac{E_b}{N_0} \leq 3.5$ dB.

5. Numerical Results

In this section, the block error rate (BLER) and bit error rate (BER) performance of product codes under AWGN channel with BPSK modulation are evaluated. Codes are decoded by using the belief propagation decoding method. Again, for product codes, the iterative decoding of their constituent decoders is conducted with 60 iterations. Five iterations are considered for the interactive decoding between two constituent decoders. For SPCs, decoding is also accomplished by 60 iterations.

Figure 4 shows performance of (48, 16) and (49, 16) product codes, as well as the (49, 16) single polar code. As mentioned in previous sections, (48, 16) and (49, 16) product codes are formed by puncturing one and both of their constituent codes, respectively. It is observed that (49, 16) product code has a similar performance to that of the single punctured code, while it performs better than (48, 16) product code. In addition, at $\frac{E_b}{N_0} \geq 5.5$ dB,

(49, 16) product code shows a performance better than that of (49, 16) SPC, as its error floor occurs at $\text{BER} < 10^{-5}$. The confidence interval of (49, 16) product code at $\frac{E_b}{N_0} = 6.0$ dB is $[0.1276 \times 10^{-3}, 0.3915 \times 10^{-3}]$, which is $[0.1907 \times 10^{-3}, 0.4945 \times 10^{-3}]$ for the SPC.

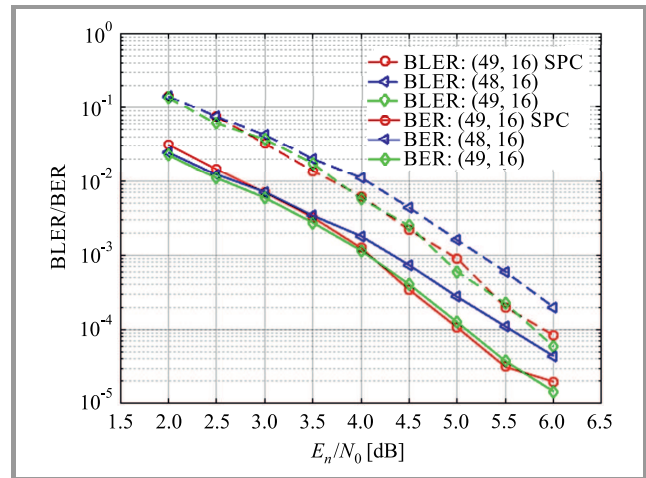


Fig. 4. BLER and BER performance comparison of (48, 16) and (49, 16) product polar codes and (49, 16) SPC.

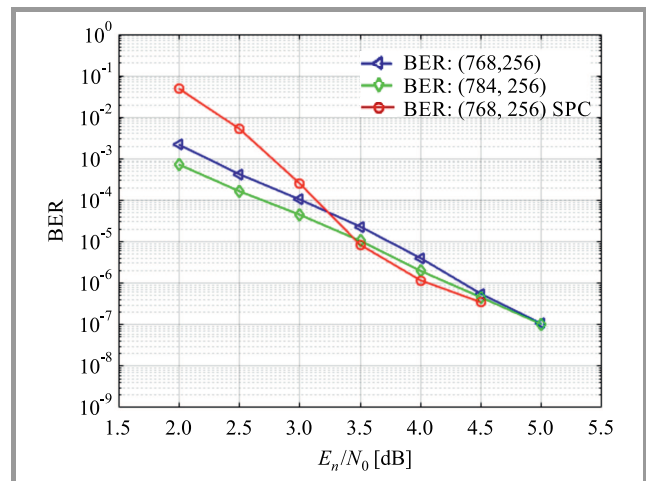


Fig. 5. BER performance comparison of (768, 256) and (784, 256) product polar codes and (768, 256) SPC.

Figure 5 shows the performance of (768, 256) and (784, 256) product codes, as well as of the (768, 256) single polar code. The (768, 256) product code is constructed by con-catenating (32, 16, 4) and (24, 16, 4) polar codes, whereas (784, 256) product code is formed by combining two (28, 16, 4) polar codes. In the (768, 256) product code, $\eta_\phi = 8$ bits from every codeword are punctured, whereas half of η_ϕ bits are punctured from 28 codewords of the (784, 256) product code. The effect of interactive decoding is observed at $\frac{E_b}{N_0} \geq 4.5$ dB for these codes, when the decoder can properly recover the punctured information from a block of information. As a lower number of bits is punctured from codewords of the (784, 256) product code, this code provides a better performance than the (768, 256)

product code for $2.0 \text{ dB} \leq \frac{E_b}{N_0} \leq 4.0 \text{ dB}$. The (784, 256) product code also provides better BER performance than (768, 256) single polar code for $\frac{E_b}{N_0} \geq 4.5 \text{ dB}$. Figure 3b shows also that a lower number of iterations is required for this product code. By contrast, for the single polar code, 256 bits are removed from the codeword, which means that its error floor is at $\frac{E_b}{N_0} \geq 4.5 \text{ dB}$. No error floor has been observed for the product code.

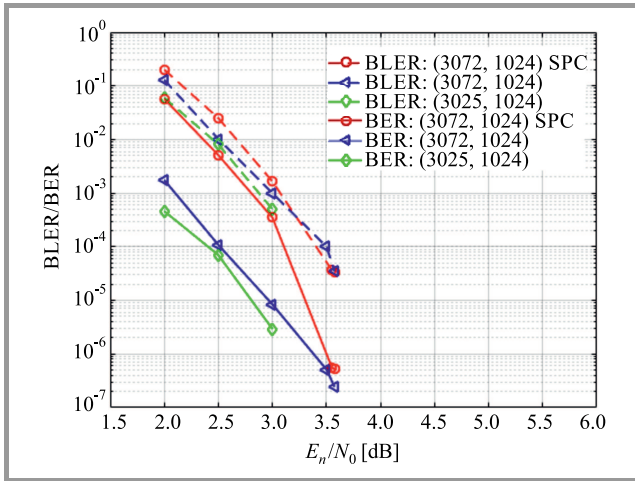


Fig. 6. BLER and BER performance comparison of (3072, 1024) and (3025, 1024) product polar codes, and (3072, 1024) SPC.

Figure 6 shows the BER and BLER performance of (3072, 1024) and (3025, 1024) codes. It is seen that both product codes have a better performance compared to that of the single code at the waterfall region. This is achieved for $\frac{E_b}{N_0} \leq 3.0 \text{ dB}$. For $\frac{E_b}{N_0} > 3.0 \text{ dB}$, (3072, 1024) SPC and PPC formed by puncturing of one of their constituent codes demonstrate a similar performance. At the same time, their confidence intervals are $[0.4404 \times 10^{-3}, 0.7497 \times 10^{-3}]$ and $[0.3970 \times 10^{-3}, 0.6932 \times 10^{-3}]$, respectively. However, the (3025, 1024) PPC constructed based on the puncturing of two constituent codes outperforms them.

6. Conclusions

The paper proposed a framework of product punctured polar codes, where puncturing is applied to both of the constituent codes. The analysis and simulations concluded that codes constructed by this method outperform the previously proposed ones, with a lower number of iterations applied in their iterative decoding structure.

References

[1] E. Arikan, "Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels", *IEEE Trans. on Inform. Theory*, vol. 55, no. 7, pp. 3051–3073, 2009 (doi: 10.1109/TIT.2009.2021379).

[2] D. M. Shin, S. C. Lim, and K. Yang, "Design of length-compatible polar codes based on the reduction of polarizing matrices", *IEEE Trans. on Commun.*, vol. 61, no. 7, pp. 2593–2599, 2013 (doi: 10.1109/TCOMM.2013.052013.120543).

[3] K. Niu, K. Chen, and J.-R. Lin, "Beyond turbo codes: Rate-compatible punctured polar codes", in *Proc. IEEE Int. Conf. on Commun. ICC 2013*, Budapest, Hungary, 2013, pp. 3423–3427 (doi: 10.1109/ICC.2013.6655078).

[4] R. Wang and R. Liu, "A novel puncturing scheme for polar codes", *IEEE Commun. Lett.*, vol. 18, no. 12, pp. 2081–2084, 2014 (doi: 10.1109/LCOMM.2014.2364845).

[5] I. L. V. Bioglio and F. Gabry, "Low-complexity puncturing and shortening of polar codes" [Online]. Available: <https://arXiv:1701.06458v1>.

[6] X. W. Q. Y. L. Zhang, Z. Zhang, and Y. Chen, "On the puncturing patterns for punctured polar codes", in *Proc. IEEE Int. Symp. on Inform. Theory ISIT 2014*, Honolulu, HI, USA, 2014, pp. 121–125 (doi: 10.1109/ISIT.2014.6874807).

[7] M. Hanif and S. Vafi, "An efficient puncturing method for the short and long length polar codes", in *Proc. 11th Int. Conf. on Sig. Process. and Commun. Syst. ICSPCS 2017*, Surfers Paradise, QLD, Australia, 2017 (doi: 10.1109/ICSPCS.2017.8270452).

[8] M. A. Hanif and S. Vafi, "A novel scheme of product polar codes with high efficiency and rate compatibility" (under review at *Wireless Communications and Mobile Computing*).

[9] H. Li and J. Yuan, "A practical construction method for polar codes in AWGN channels", in *Proc. IEEE TENCON Spring Conf.*, Sydney, NSW, Australia, pp. 223–226, 2013 (doi: 10.1109/TENCONSpring.2013.6584444).

[10] H. Vangala, E. Viterbo, and Y. Hong, "A comparative study of polar code constructions for the AWGN channel" [Online]. Available: <https://arxiv.org/abs/1501.02473>

[11] R. Y. Shao, S. Lin, and M. P. C. Fossorier, "Two simple stopping criteria for turbo decoding", *IEEE Trans. on Commun.*, vol. 47, no. 8, pp. 1117–1120, 1999 (doi: 10.1109/26.780444).

[12] E. Arikan, "Polar codes: A pipelined implementation", in *Proc. 4th Int. Symp. on Broadcast Commun. ISBC 2010*, Melaka, Malaysia, 2010, pp. 11–14 [Online]. Available: <https://pdfs.semanticscholar.org/a9ed/49203bffe7b124984a5874fa0940103d8ab.pdf>

[13] E. Arikan, "Systematic polar coding", *IEEE Commun. Lett.*, vol. 15, no. 8, pp. 860–862, 2011 (doi: 10.1109/LCOMM.2011.061611.110862).

[14] N. Hussami, S. B. Korada and R. Urbanke, "Performance of polar codes for channel and source coding", in *Proc. IEEE Int. Symp. on Inform. Theory ISIT 2009*, Seoul, South Korea, 2009 (doi: 10.1109/ISIT.2009.5205860).

[15] Q. Zhang, A. Liu, Y. Zhang, and X. Liang, "Practical design and decoding of parallel concatenated structure for systematic polar codes", *IEEE Trans. on Commun.*, vol. 64, no. 2, pp. 456–466, 2016 (doi: 10.1109/TCOMM.2015.2502246).



Mohammad Abu Hanif is currently pursuing his Ph.D. degree in Engineering and Information Technology at Charles Darwin University, Australia. He hold a M.Sc. degree in Information and Communications from South Korea and a B.Sc. degree in Computer Science and Engineering from Bangladesh. After his master course, he worked on

several projects as a Research Assistant in South Korea. His research interests are in information theory, error-control coding and wireless communications.

E-mail: mohammad.hanif@cdu.edu.au
 College of Engineering, IT and Environment
 Charles Darwin University
 Darwin, Australia



Sina Vafi received his Ph.D. degree from the University of Wollongong (UoW), Wollongong, Australia, in 2006. In 2007, he joined Charles Darwin University as a Lecturer of Electrical and Electronics Engineering. In 2008, he became a Senior Lecturer. From 2006 to 2007, he was involved in re-

search, design, and implementation of 2.4 GHz broadband wireless mesh networks at UoW. He currently leads research projects on error protection techniques and their application for broadband wireless video transmission systems. His main research interests include algebraic coding theory and video transmission systems.

E-mail: sina.vafi@cdu.edu.au

College of Engineering, IT and Environment
Charles Darwin University
Darwin, Australia