SCIENCE CHINA
Information Sciences

• **Supplementary File** •

Low-Complexity Polar Code Construction for Higher Order Modulation

Yongrun YU[1], Zhiwen PAN[1*] & Nan LIU[1]

[1]*National Mobile Communications Research Laboratory, Southeast University, Nanjing, Jiangsu* 210096*, China*

Appendix A Preliminaries

Appendix A.1 Polar Codes

Polar codes are linear block code with generator matrix $\mathbf{G}_N = \mathbf{B}_N \mathbf{F}_2^{\otimes \log_2 N}$, where N is code length and $\mathbf{B}_N$ denotes bit-reverse permutation matrix [1]. $\mathbf{F}_2 = \left[\begin{smallmatrix} 1 & 0 \\ 1 & 1 \end{smallmatrix}\right]$. In this paper, $\mathbf{a}_1^N$ is used to denote row vector $(a_1, ..., a_N)$. With this notation, the encoding process can be expressed as $\mathbf{x}_1^N = \mathbf{u}_1^N \mathbf{G}_N$, where $\mathbf{x}_1^N = (x_1, ..., x_N)$ is codeword and $\mathbf{u}_1^N = (u_1, ..., u_N)$ is source bit sequence. $\mathbf{u}_1^N$ includes both information bits and frozen bits. Frozen bits are fixed to zero in this appendix. This work focuses on how to choose the index set of information bits in $\mathbf{u}_1^N$ under higher order modulation.

Denote W a binary-input memoryless symmetric channel with input alphabet $\mathcal{X} = \{0, 1\}$ and output alphabet $\mathcal{Y}$. $W(y|x)$ is the transition probability of W, where $x \in \mathcal{X}$ and $y \in \mathcal{Y}$. The Bhattacharyya parameter of W defined in [1] is $Z(W) = \sum_{y \in \mathcal{Y}} \sqrt{W(y|0)W(y|1)}$.

Denote $W_N^{(i)}$ the i-th synthesized channel in a length-N polar code with channel transition probability $W_N^{(i)}(\mathbf{y}_1^N, \mathbf{u}_1^{i-1}|u_i) = \frac{1}{2^{N-1}} \sum_{\mathbf{u}_{i+1}^N} \prod_{i=1}^N W(y_i|x_i)$. If the two underlying channels for $\mathbf{F}_2$ are different, the Bhattacharyya parameters of $W_2^{(1)}$ and $W_2^{(2)}$ are [10]:

$$Z(W_2^{(1)}) \leqslant Z(W_1) + Z(W_2) - Z(W_1)Z(W_2), Z(W_2^{(2)}) = Z(W_1)Z(W_2), \tag{A1}$$

where $Z(W_1)$ and $Z(W_2)$ are Bhattacharyya parameters of the two underlying channels with transition probability $W_1(y_1|x_1)$ and $W_2(y_2|x_2)$, respectively. The equality holds if W_1 and W_2 are both BECs.

Above Bhattacharyya parameters and their bounds can be used to construct polar code. Besides, as suggested in [1], Monte Carlo simulation can be used to construct polar codes via repeatedly running SC decoder to obtain the expectation of Bhattacharyya parameter for each polarized channel. However, Monte Carlo method requires numerous repeated SC decoding processes, which results in very high complexity $O(TN\log N)$ (T is usually of the order 10^6). Gaussian approximation [2] is also a popular algorithm to construct polar code, but it may involve numerical calculation to deal with $\phi(x)$ and $\phi^{-1}(x)$, where $\phi(x) = 1 - \frac{1}{\sqrt{4\pi x}} \int_{-\infty}^{+\infty} \tanh(\frac{\mu}{2}) \exp(-\frac{(\mu - x)^2}{4x}) d\mu$. Therefore, polar code construction using Bhattacharyya parameter has lower complexity, and in this appendix, we focus on polar code construction based on Bhattacharyya parameter.

Appendix A.2 Multi-level Coded Modulation

MLC is a coded modulation scheme that can achieve the coded modulation capacity under multi-stage decoding (MSD) [12]. MSD uses the decoding results of former component codes as feedback to demodulate and decode the remaining component codes.

Let $\mathcal{C}$ denote a 2^m-ary constellation. For any constellation point $c_i \in \mathcal{C}$, there exists a unique bit label corresponding to c_i, i.e., $c_i \leftrightarrow (b_1^{(i)}, ..., b_m^{(i)})$, where $b_j^{(i)} \in \{0, 1\}, j = 1, ..., m$ and the subscript $1, ..., m$ is called bit level. The channel that $b_j^{(i)}$ observes with transition probability $P(b_1^{(i)}, ..., b_{j-1}^{(i)}, y_i | b_j^{(i)})$ is called equivalent bit level channel j in MLC category, where y_i is channel output when c_i is the input. In the rest of this appendix, it is assumed that every $c_i \in \mathcal{C}$ has equiprobability and thus mutual information equals capacity. The channel capacity under MLC scheme is $C_{\text{MLC}} = I(\mathbf{B}_1^m; Y) = \sum_{i=1}^m I(B_i; Y|\mathbf{B}_1^{i-1})$, where $\mathbf{B}_1^m, B_i \in \{0, 1\}$ is the set of bit label of constellation points and Y is channel output. $I(B_i; Y|\mathbf{B}_1^{i-1})$ is the capacity of the i-th bit level and it can be calculated as [12]:

* Corresponding author (email: pzw@seu.edu.cn)

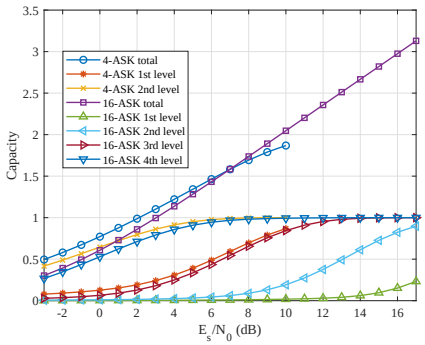

Figure A1 Bit level capacity under 4-ASK and 16-ASK.

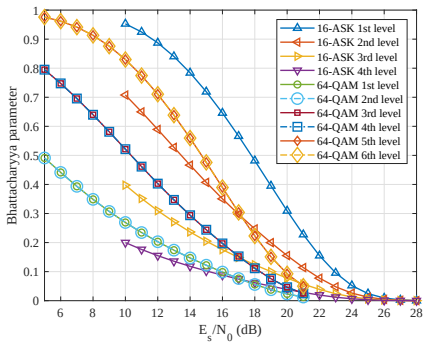

Figure A2 Bit level Bhattacharyya parameter under 16-ASK and 64-QAM.

$$I(B_i;Y|\mathbf{B}_1^{i-1}) = I(\mathbf{B}_i^m;Y|\mathbf{B}_1^{i-1}) - I(\mathbf{B}_{i+1}^m;Y|\mathbf{B}_1^i) = \sum_{\mathbf{b}_1^i,y} p(y,\mathbf{b}_1^i)\log_2\frac{p(y|\mathbf{b}_1^i)}{p(y|\mathbf{b}_1^{i-1})}, \tag{A2}$$

Examples of bit level capacity under 4-ASK and 16-ASK with set partitioning (SP) labeling are shown in Figure A1, where E_s/N_0 denotes the SNR in terms of average symbol energy.

Under MLC scheme, each bit level i has its own component code and the code rates among component codes are in general different. The code rate of a given component code in bit level i mainly relies on the capacity of this level, i.e., $I(B_i;Y|\mathbf{B}_1^{i-1})$. In order to achieve better performance under MLC scheme, the bit labeling rule should be SP labeling [4,5] because SP labeling ensures larger differences between bit level capacities, which results in a better modulation polarization.

Appendix A.3 Bit-interleaved Coded Modulation

Under 2^m-ary BICM scheme, each bit level i is treated independently when calculating the channel reliability metric, i.e., log-likelihood ratios (LLRs), at the demodulator [13]. Therefore, the channel with transition probability $W(y|c), c \in \mathcal{C}$ can be considered as m parallel bit channels as follows:

$$W(y|c) \to \{W(y|b_1), ..., W(y|b_m)\}. \tag{A3}$$

Under BICM, channel with transition probability $W(y|b_i)$ is called equivalent bit level channel i. Let $l^i(c)$ denote the i-th bit in a constellation point c and $\mathcal{C}_b^i$ represents all points in the constellation such that $l^i(c) = b$, i.e., $\mathcal{C}_b^i = \{c|l^i(c) = b\}$, where $b \in \{0,1\}$. The transition probability of $W(y|b_i)$ is:

$$W(y|b_i = b) = \frac{1}{2^{m-1}} \sum_{c\in\mathcal{C}_b^i} W(y|c). \tag{A4}$$

The Bhattacharyya parameter of $W(y|b_i)$ is:

$$Z(W(y|b_i)) = \frac{1}{2^{m-1}} \sum_y \sqrt{\sum_{c_0\in\mathcal{C}_0^i} W(y|c_0) \sum_{c_1\in\mathcal{C}_1^i} W(y|c_1)}. \tag{A5}$$

Examples of bit level Bhattacharyya parameter under 16-ASK and 64-QAM with Gray labeling are shown in Figure A2. For 64-QAM, $Z(W(y|b_i)) = Z(W(y|b_{i+1}))$, $i = 1, 3, 5$.

Since each bit level channel is treated independently when demodulating, the bit labeling rule is usually Gray labeling that makes the bits in bit label uncorrelated [4].

Appendix B Polar code construction for MLC scheme

For MLC scheme, by considering the differences between bit level capacities as modulation polarization and regarding the bit level channels as equivalent BECs, a low complexity polar code construction based on Bhattacharyya parameter is proposed. Next, an adaptive list decoding scheme is proposed to efficiently decode polar codes under MLC.

Appendix B.1 Relationship between Code and Modulation Polarization

The relationship between code and modulation polarization is shown in Figure B1 and Figure B2.

Figure B1 illustrates the common code polarization assuming that the underlying channels are BECs with capacity 0.5. The red line represents the first half polar code based on synthesized BECs with capacity 0.25 after the first polarization stage, while the black line denotes the second half polar code based on synthesized BECs with capacity 0.75.

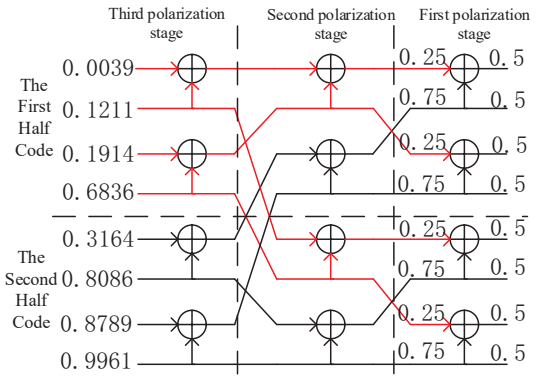

Figure B1 Polar code polarization based on BEC with capacity 0.5.

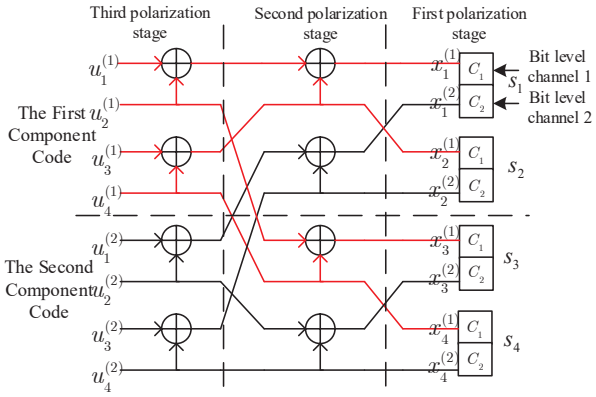

Figure B2 Combine code polarization with modulation polarization.

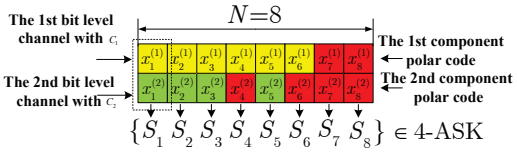

Figure B3 Polar coded 4-ASK MLC modulation.

In Figure B2, the first polarization stage is replaced with modulation polarization, where the blocks in the right-most column represent symbols in 4-ary modulation. C_1 and C_2 denote the first and second bit level capacity respectively under MLC scheme, i.e., $C_1 = I(Y;B_1)$, $C_2 = I(Y;B_2|B_1)$. $s_1, ..., s_4$ represent the modulated symbols. $\mathbf{x}^{(1)} = (x_1^{(1)}, ..., x_4^{(1)})$ and $\mathbf{x}^{(2)} = (x_1^{(2)}, ..., x_4^{(2)})$ denote polar encoded bits of $\mathbf{u}^{(1)} = (u_1^{(1)}, ..., u_4^{(1)})$ and $\mathbf{u}^{(2)} = (u_1^{(2)}, ..., u_4^{(2)})$, respectively. The red line corresponds to the first component polar code $\mathbf{x}^{(1)}$ that will be mapped to the first bit level while the black line corresponds to the second component polar code $\mathbf{x}^{(2)}$ that will be mapped to the second bit level.

It can be observed that Figure B1 and Figure B2 are identical except the first stage of polarization. In Figure B2, the first stage of polarization is caused by the 4-ary modulation. The two resulting MLC bit level channels have capacity C_1 and C_2, while in Figure B1 the first polarization stage produces two channels with capacity 0.25 and 0.75. This implies that under MLC scheme, after the first modulation polarization stage, the construction of polar codes can be implemented just as in binary modulation scheme, i.e., in Figure B2 bit level channel with capacity C_1 is used to construct the first length-4 polar code (red line) and bit level channel with capacity C_2 is used to construct the second length-4 polar code (black line).

In Figure B2, even if the underlying channel that transmits 4-ary symbols is AWGN channel, the two resulting bit level channels whose capacities are C_1 and C_2 are in general not AWGN channel, which makes polar code construction difficult. However, if we consider the two resulting bit level channels as BECs with Bhattacharyya parameters Z_1 and Z_2, i.e., $Z_1 = 1 - C_1$ and $Z_2 = 1 - C_2$, then (A1) can be used to calculate the Bhattacharyya parameter of the synthesized bit channels, thus completing the polar code construction.

Appendix B.2 Proposed Polar Code Construction for MLC Scheme

The proposed polar code construction scheme under 2^m-ary MLC modulation includes three steps.

(I) Calculate the bit level capacities, i.e., $C_i = I(Y; B_i|\mathbf{B}_1^{i-1})$, $1 \leqslant i \leqslant m$, which is based on (A2).

(II) Virtualize the m bit level channels as equivalent BECs with $Z_i = 1 - C_i$, $1 \leqslant i \leqslant m$. Then recursively calculate the Bhattacharyya parameter of synthesized bit channels according to (A1).

(III) Select K synthesized bit channels with K smallest Bhattacharyya parameters to transmit information bits while the remaining $mN - K$ bit channels are used to carry frozen bits, where N is the length of one component polar code and m is the modulation order.

An example of the proposed code construction approach is shown in Figure B3, where 4-ASK and polar codes with length 8 are considered. In Figure B3, the first row represents the first component polar coded bits $\mathbf{x}_1 = (x_1^{(1)}, ..., x_8^{(1)})$ and the second row represents the second polar coded bits $\mathbf{x}_2 = (x_1^{(2)}, ..., x_8^{(2)})$. The first component polar code is constructed by virtualizing the first bit level channel as a BEC with capacity C_1. The second component polar code is constructed by virtualizing the second bit level channel as a BEC with capacity C_2. The bit pair $(x_i^{(1)}, x_i^{(2)})$ is mapped to a 4-ASK symbol. The red blocks denote the synthesized bit channels with lowest Bhattacharyya parameters and thus they are used to carry information bits. The remaining blocks carry frozen bits. The modulated symbols are transmitted and at the receiver side, MSD [12] is used to sequentially demodulate and decode each component code. This example has rate $R = 6/16$.

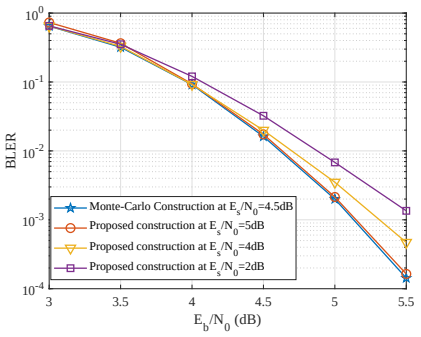

Figure B4 BLER of the proposed 4-ASK MLC codes.

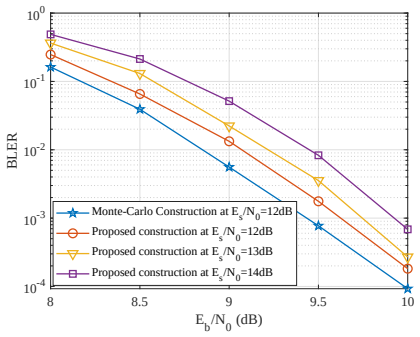

Figure B5 BLER of the proposed 16-ASK MLC codes.

Note that there is no need to design rate allocation rules for each component polar code under the proposed method. Given some E_s/N_0, m bit level capacities $(C_1, ..., C_m)$ are obtained via (A2), and $(Z_1, ..., Z_m) = (1 - C_1, ..., 1 - C_m)$ is also obtained. Then Z_i is used to calculate Bhattacharyya parameters $\mathbf{Z}_i = (Z_{1,i}, ..., Z_{N,i})$ for polarized channels in the i-th component code according to (A1). Next, a length-mN vector $\mathbf{Z}_{\text{total}} = (\mathbf{Z}_1, ..., \mathbf{Z}_m)$ is obtained by sequentially placing $\mathbf{Z}_i, 1 \leqslant i \leqslant m$. Once $\mathbf{Z}_{\text{total}}$ is obtained, K information bits are automatically allocated to each component codes by selecting K smallest elements in $\mathbf{Z}_{\text{total}}$, which is exactly the step (III) of the proposed scheme. Therefore, once E_s/N_0 is given, $\mathbf{Z}_{\text{total}}$ and subsequently polar code construction are obtained.

The implementation procedure of the proposed MLC polar code construction algorithm is similar to polar code construction procedure based on Bhattacharyya parameters in binary modulation case [10], except that under MLC scheme there are totally m component codes that need to be constructed. Polar code construction based on Bhattacharyya parameters in binary modulation case has complexity $O(N\log N)$ [10], where N is the codeword length. Therefore, the time complexity of the proposed code construction scheme is $O(mN\log N)$, where m is the modulation order. Note that Gaussian approximation (GA) [2] has the same asymptotic complexity, but the unit complexity of GA is much higher than that of the proposed method because GA has to deal with $\phi(x)$ and $\phi^{-1}(x)$, either through numerical calculations or table look-up. If the channel up/degrading method in [3] is used under MLC, the complexity is $O(mN\mu^2 \log \mu)$, where $\mu = (\log N)^2$ is suggested in Theorem 1 of [17] to guarantee good approximations. For Monte Carlo construction, the complexity is obviously $O(TmN\log N)$, where T is the number of repeated simulations for code construction, usually with the order 10^6. A brief complexity comparison between the proposed method and existing methods under MLC scheme is shown in Table B1.

Appendix B.3 Effectiveness of the Proposed Method

In this section, we show the effectiveness of the proposed code construction method under SC decoding by comparisons with simulation-based Monte Carlo approach.

The BLER performance comparison with Monte Carlo method is shown in Figures B4 and B5 for 4-ASK and 16-ASK, respectively. For 4-ASK, there are 2 component polar codes with $N = 512$, while for 16-ASK, there are 4 component codes with $N = 256$. The code rate is 0.5 and labeling rule is SP for both schemes. Modulation symbols are transmitted through AWGN channel and SC decoder is used. It can be seen that by adjusting the E_s/N_0 at which polar codes are constructed, the BLER of proposed method can be very close to that of Monte Carlo method under 4-ASK. The BLER gap between the proposed and Monte Carlo method is around 0.2dB under 16-ASK. Above results indicate the effectiveness of the proposed construction method. More detailed information is as follows. For 4-ASK, codes constructed at $E_s/N_0 = 5$dB with rate allocation 0.1797/0.8203 yield best BLER. The Monte Carlo construction for 4-ASK is training at $E_s/N_0 = 4.5$dB with rate allocation 0.1758/0.8242. For 16-ASK, codes constructed at $E_s/N_0 = 12$dB with rate allocation 0.0039063/0.18359/0.83984/0.97266 yield best BLER. The Monte Carlo construction for 16-ASK is training at $E_s/N_0 = 12$dB with rate allocation 0.0039063/0.16796875/0.828125/1.

Appendix B.4 Adaptive CA-SCL decoder in Polar Coded MLC scheme

In this section, we point out that under MLC, polar code constructed for SC decoder may not be suitable for SCL decoder [18]. This phenomenon makes the design of decoder closely related to code construction. Therefore, we propose an adaptive list decoding scheme for polar coded MLC based on the observation of the following simulation results.

Table B1 Polar code construction complexity comparisons under MLC scheme

Construction methods	Asymptotic complexity
the proposed method	$O(mN\log N)$, where $O(1)$ represents one time real number multiplication
Gaussian approximation [2]	$O(mN\log N)$, where $O(1)$ represents one time computation of $\phi(x)$ and $\phi^{-1}(x)$
Channel up/degrading [3] [17]	$O(2mN(\log N)^4(\log\log N))$
Monte Carlo construction	$O(TmN\log N)$ with T of the order 10^6

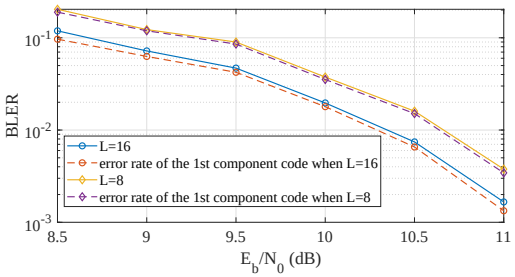

Figure B6 BLER of 16-ASK MLC polar codes under CA-SCL with $L = 8$ and 16. Each component code is concatenated with 8 bits CRC.

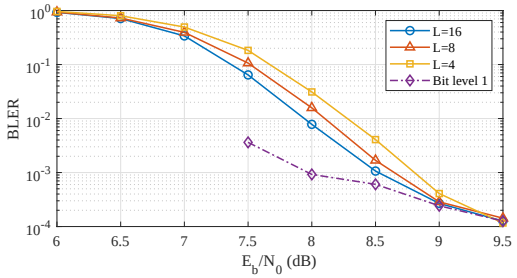

Figure B7 BLER of 16-ASK MLC polar codes under CA-SCL with different list size. The first component polar code is decoded by SC.

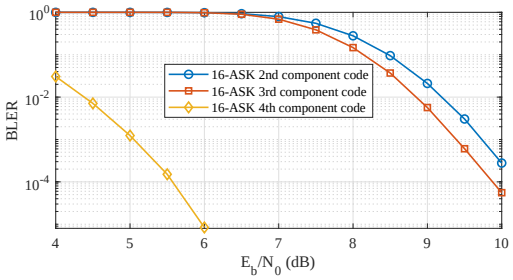

Figure B8 BLER of last three component codes in 16-ASK polar coded MLC.

In last subsection, under 16-ASK, code constructed at $E_s/N_0 = 12$dB has best BLER under SC. However, the first component code only has $0.0039 \times 256 = 1$ information bit (so does Monte Carlo construction), which makes it not suitable to concatenate with CRC bits. This can be verified by Figure B6, where each component polar code is concatenated with 8 bit CRC $g(x) = x^8 + x^6 + x^3 + x^2 + 1$ [16], and decoded by list size $L = 8$ and 16. Other simulation configurations in Figure B6 are the same as in Figure B5. It can be observed that although CA-SCL scheme is employed, the BLER is even worse than SC decoder shown in Figure B5. The reason is that the error rate of the whole code is dominated by the first component code because we use 8 bits CRC to protect only one information bit, which increases the rate of the first component polar code from $1/N$ to $9/N$, degrading the error correction performance.

One may argue that since there is only one information bit in the first component code, SC decoder is exactly the maximum likelihood decoder. Thus SC can be employed to decode the first component code, and the remaining three component codes are still decoded by CA-SCL. However, we will show such scheme yields the same phenomenon as in Figure B6. In Figure B7, the first component code is decoded by SC and the remaining three component codes are decoded by CA-SCL with list size $L = 4, 8$ and 16. The BLER performances of the whole code and only the first code are both given. It can be seen that although we can increase the list size for the last three sub-codes, the BLER is dominated by the error rate of the first component code, i.e., once the decoding of the first component code is wrong, the decoding of the whole code will fail. Figures B6 and B7 imply that polar codes constructed for SC decoder may be inappropriate for CA-SCL decoder at least under 16-ASK MLC scheme.

To overcome this, for 16-ASK, we suggest to select an E_s/N_0 that results in a totally frozen first component code, i.e., the rate of the first component code is 0. There are two advantages to do so. On the one hand, since the first component code consists of frozen bits, the decoder knows the codeword and there is no need to decode the first component code. On the other hand, the BLER of the whole code will not be limited by the first component code (this code observes bit channel with lowest capacity). Set $E_s/N_0 = 13$dB and the corresponding rate allocation is 0/0.19922/0.83984/0.96094. We then observe the BLER for the last three component codes in Figure B8, where simulation configuration is the same as in Figure B7 except that: no error propagation exists in MSD/the decoder is SC. Obviously, it can be concluded that the second and third component codes have much higher BLER, which indicates that once the decoding of whole code fails, it is very likely that the error occurs at the second or the third code. Therefore, these two sub-codes require larger list size to protect them. Based on this observation, an adaptive CA-SCL scheme for polar coded MLC system is proposed.

With some modification, adaptive CA-SCL decoder in [14] can be extended to polar coded MLC. [14] focuses on binary modulation (only one code in one frame) so maximum allowed list size $L_{\max}$ is fixed. Here, maximum allowed list size $L_{\max,i}$ is assigned for the i-th component polar code, i.e., component code with higher error rate may require larger $L_{\max}$. When decoding the i-th code, at first, SC (with list size 1) is used. If decoding result passes CRC check, the decoding of i-th component code is successfully finished. If not, list size doubles and CA-SCL is employed. One advantage of such adaptive scheme is that the decoding automatically stops when CRC is satisfied (the minimum required list size is automatically found), which avoids complex analysis between list size and BLER for every component code. Another advantage is that

Algorithm B1 Adaptive CA-SCL for Polar coded MLC

Input: $\mathbf{y}_1^N$, $\{L_{\max,1},...,L_{\max,m}\}$
Output: $\hat{\mathbf{x}}_{1,1}^N,...,\hat{\mathbf{x}}_{1,m}^N$
1: **for** $i \leftarrow 1:m$ **do**
2: $\mathbf{llr}_1^N \leftarrow$ MLC demodulator($\mathbf{y}_1^N$, $\hat{\mathbf{x}}_{1,1}^N,...,\hat{\mathbf{x}}_{1,i-1}^N$), $L \leftarrow 1$, $\hat{\mathbf{x}}_{1,i}^N \leftarrow$ SC($\mathbf{llr}_1^N$)
3: **while** $\hat{\mathbf{x}}_{1,i}^N$ does not pass CRC **do**
4: $L \leftarrow 2L$
5: **if** $L > L_{\max,i}$ **then**
6: Decoding fails
7: **else**
8: $\hat{\mathbf{x}}_{1,i}^N \leftarrow$ CA-SCL($\mathbf{llr}_1^N$, L)
9: **end if**
10: **end while**
11: **end for**

when the list size for decoding i-th component code reaches $L_{\max,i}$ and the decoding result still does not pass CRC, then the decoding of the i-th component code fails. This brings early termination criterion for polar coded MLC scheme. Once the decoding of a certain component code fails, the decoder immediately reports decoding failure, which further reduces decoding latency. This adaptive scheme can be described by Algorithm 1.

In Algorithm 1, the inputs are received signal $\mathbf{y}_1^N$ and maximum decoding list size $\{L_{\max,1},...,L_{\max,m}\}$ for each component code. The outputs $\hat{\mathbf{x}}_{1,1}^N,...,\hat{\mathbf{x}}_{1,m}^N$ are the estimates for each component code. Line 1 means that there are totally m component polar codes to be demodulated and decoded. For each component code, line 2 shows that after demodulation, SC decoder is first employed to decode each component code. Once the decoding result of SC decoder does not satisfy CRC, lines 4-8 are activated. The list size L is doubled and CA-SCL decoder with L is used until there exists one decoding path passing CRC check (decoding successes) or a predetermined $L_{\max}$ is reached (decoding fails). The decoding complexity of Algorithm 1 is given in the following proposition.

Proposition 1. Algorithm 1 has worst case complexity $O([\sum_{i=1}^{m} 1 + P_{e,i}(2L_{\max,i}-2)]N\log N)$, where $P_{e,i}$ is the BLER of the i-th componet polar code. This worst case complexity is obtained without consideration of above stated early termination criterion.

Proof. $C_{\rm sc} = O(N\log N)$ represents the complexity of SC decoder and CA-SCL decoder with list size L approximately has complexity $LC_{\rm sc}$. Assume that μ polar coded MLC blocks are transmitted and consider the i-th, $1 \leqslant i \leqslant m$, component code. Denote ν_i the number of i-th component code that cannot be correctly decoded by SC decoder. L_i^k is the list size when adaptive CA-SCL decoder stops for the k-th, $1 \leqslant k \leqslant \nu_i$, received MLC block, where the i-th component polar code cannot be correctly decoded by SC decoder. Then the average complexity C_i for decoding i-th component code can be expressed as:

$$\begin{aligned} C_i &= \frac{\mu + \sum_{k=1}^{\nu_i}(2+4+...+L_i^k)}{\mu} C_{\rm sc} = \frac{\mu + \sum_{k=1}^{\nu_i}(2L_i^k-2)}{\mu} C_{\rm sc} \\ &\leqslant \frac{\mu + \nu_i(2L_{\max,i}-2)}{\mu} C_{\rm sc} = [1 + P_{e,i}(2L_{\max,i}-2)]C_{\rm sc}. \end{aligned} \tag{B1}$$

Since there are totally m component codes that need to be decoded, the total worst case complexity can be obtained by adding (B1).

Appendix C Polar code construction for BICM scheme

In this section, a novel channel mapping scheme is introduced, through which polar codes are constructed for BICM scheme. First, for a better understanding of the proposed channel mapping and code construction scheme, the system model of our BICM scheme is given. Next, a general channel mapping and construction algorithm is proposed. Then, we take 16-ASK and 64-QAM as examples to further illustrate the application of the general construction algorithm.

Appendix C.1 Polar Coded BICM System Model

In general, as stated in [6, 9], if we want to get optimized polar codes for BICM, there are totally $N!/((N/m)!)^m$ different channel mappings that match coded bits to a certain modulation bit level, where N is code length and m denotes modulation order. $N!/((N/m)!)^m$ is too large to be studied. Hence, in this appendix we employ a suboptimal model. BICM model in [11] is adopted and a novel channel mapping method is proposed in this section. One may be interested in how far the suboptimal scheme is from the optimal one. This can be achieved by making comparisons with Monte Carlo-based polar code construction, which can be considered as optimal with sufficiently large times of repeated running. Such comparisons are done in Appendix D, where simulation results are given. The BICM system model [11] is shown in Figure C1.

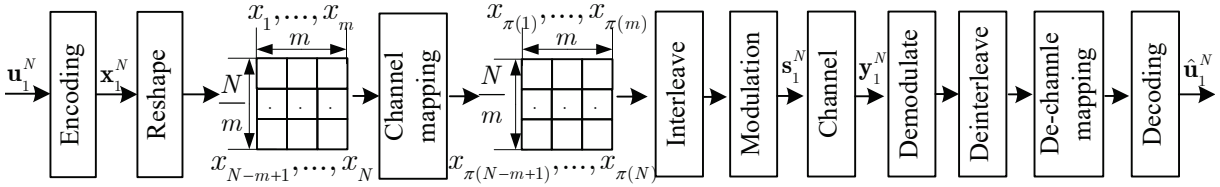

Figure C1 BICM system model.

$\mathbf{u}_1^N = (u_1, ..., u_N)$ denotes the source bits before encoding and $\mathbf{x}_1^N = (x_1, ..., x_N)$ represents the polar coded bits. The reshaping process of $\mathbf{x}_1^N$ puts $(x_1, ..., x_m)$ in the first row, $(x_{m+1}, ..., x_{2m})$ in the second row, and so on, where m is modulation order. The channel mapping procedure is a column permutation of the reshaped version of $\mathbf{x}_1^N$. The first row of the output of the channel mapping is denoted by $(x_{\pi(1)}, ..., x_{\pi(m)})$, the second row is denoted by $(x_{\pi(m+1)}, ..., x_{\pi(2m)})$, and so on. Note that in this model channel mapping is simplified to a column permutation such that $\pi(km + i) = \pi((k-1)m + i) + m$. The meaning of the channel mapping process is to better combine polar codes with the 2^m-ary modulation. After column-wise random interleaving, each row that consists of m bits is mapped to a constellation symbol. The receiver performs demodulating, deinterleaving, de-channel mapping, and decoding.

Appendix C.2 Channel Mapping and Code Construction Scheme

In this subsection, channel mapping and code construction scheme are proposed for BICM. Bhattacharyya parameters in (A1) and (A5) are used to analyze the polarization phenomenon caused by polar codes as well as modulation.

In Figure C1, the rows of the output of interleaver module will be mapped to constellation symbols. In order to better combine polar codes with the 2^m-ary modulation, we first design polar codes with length m (typically very short) to adapt to the modulation polarization caused by the 2^m-ary modulation (The reason why doing so will be explained in the following several paragraphs). This is achieved by solving the following problem:

$$\begin{aligned} \underset{\pi\{1,\ldots,m\}}{\text{minimize}} \quad & \textstyle\sum_{i=m/2+1}^{m} Z(W_m^{(i)}), \\ \text{s.t.} \quad & Z(W_i) = Z(W(y|b_{\pi(i)})), \\ & i = 1, ..., m. \end{aligned} \tag{C1}$$

where m is the modulation order. $W_m^{(i)}$ is the i-th synthesized bit channel of a length-m polar code. $W(y|b_{\pi(i)})$ is the $\pi(i)$-th equivalent bit channel and W_i transmits the i-th coded bit. The search space of the optimization problem is $\pi\{1, 2, ..., m\}$, which represents all permutations of integer set $\{1, 2, ..., m\}$. Thus, the search space contains $m!$ elements. π is called *channel mapping* in this appendix.

$Z(W_m^{(i)})$ in (C1) is obtained via (A1). The recursive calculation in (A1) starts from the following equation, i.e., using (C2) to initialize the recursive construction process:

$$Z(W_i) = Z(W(y|b_{\pi(i)})), 1 \leqslant i \leqslant m, \tag{C2}$$

where $Z(W(y|b_{\pi(i)}))$ is obtained by using π to permute the Bhattacharyya parameter $Z(W(y|b_i))$ of i-th modulation bit level. The value of $Z(W(y|b_i))$ can be found in Figure A2 when E_s/N_0 is given.

Equation (C2) means that the Bhattacharyya parameter of the underlying channel, i.e., $Z(W_i)$, is obtained by permuting the Bhattacharyya parameter of modulation bit level channels, i.e., $Z(W(y|b_i))$. Figure C2 is a simple example (solid line box part) to illustrate (C1) when $m = 4$. Note that there are two identical π, which represents the coded bits are partitioned into N/m blocks and the same permutation is implemented for each block. This finishes the channel mapping process for the whole N coded bits.

The summation over $m/2 + 1$ to m is inspired by the polarization figure in [1], which indicates that the second half bit indices guarantee higher reliability and a large portion of information bits tends to cluster in the second half bit index. For example, the capacity of synthesized bit channels of (1024, 512) polar codes that are constructed assuming that all underlying channels are BECs with capacity 0.5 is shown in Figure C3. Among 512 information indices, 385 indices (the percentage is $p = 75.2\%$) are larger than $N/2 = 512$. If the code rate is reduced to 0.125, i.e., there are 128 information bits, above percentage p will increase to 93.75%. This phenomenon inspires that we should minimize the Bhattacharyya parameter of the second half synthesized bit channels to obtain better BLER performance because a lager portion of information bit is located in these channels.

In Figure C2 we can see that Bhattacharyya parameters $Z_1 + Z_2$ and $Z_3 + Z_4$ are minimized through solving (C1). Z_1, Z_2, Z_3, and Z_4 are then used to construct the second half polar code. Since $Z_1 + Z_2$ and $Z_3 + Z_4$ are minimized, the reliability of four synthesized bit channels in the dashed line box (the second half synthesized bit channel) is guaranteed. It can be checked that when $N > 8$, the second half synthesized bit channel has relatively small Bhattacharyya parameter due to solving (C1), i.e., the property shown in Figure C2 can be extended to longer polar codes.

Since the modulation order m is relatively small and thus $m!$ is also small, e.g., for 16-ASK, $m = 4$ and $4! = 24$, brute-force search can be applied to solve (C1). Note that the solution to (C1) may not be unique. This phenomenon is due to the symmetry in (A1), i.e. even if $Z(W_1)$ and $Z(W_2)$ are swapped, $Z(W_2^{(1)})$ and $Z(W_2^{(2)})$ still remain the same value. This symmetry property is studied in [9]. According to Theorem 1 in [9], the number of permutations that will result in different

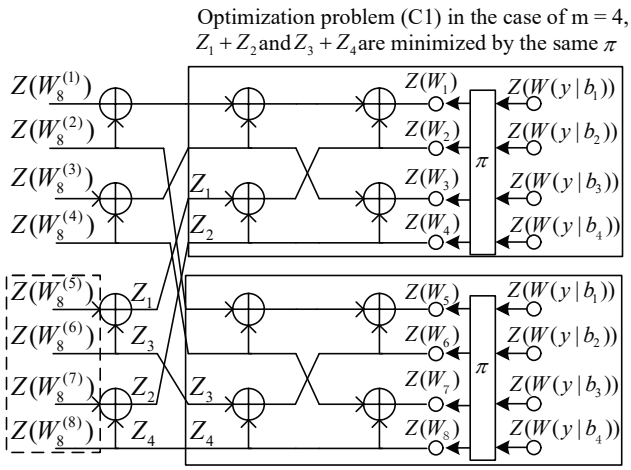

Figure C2 An example to illustrate (C1).

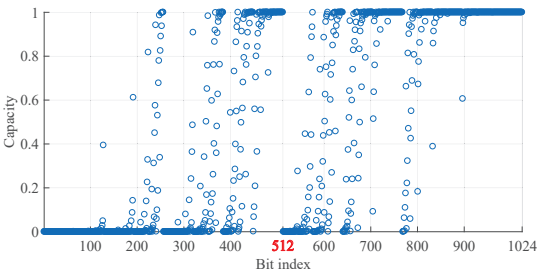

Figure C3 Capacity of synthesized bit channel in polar code with length 1024.

Algorithm C1 BICM Polar Code Construction

Input: $\mathbf{Z} = \{Z(W(y|b_i))\}_{i=1}^{m}$, K, N

Output: information set $\mathcal{A}$

1: Solving optimization problem (C1) and get π, $\mathbf{Z}_\pi \leftarrow \pi(\mathbf{Z})$
2: **for** $i \leftarrow 1 : N$ **do**
3: $\quad Z(W_i) \leftarrow \mathbf{Z}_\pi(i \bmod m)$
4: **end for**
5: recursively calculate $Z(W_N^{(i)})$ using the upper bound in (A1)
6: sort $\left\{Z(W_N^{(1)}), ..., Z(W_N^{(N)})\right\}$
7: select the indices of K smallest $Z(W_N^{(i)})$ as information set $\mathcal{A}$

polarized Bhattacharyya parameters is reduced from $m!$ to $(m!)/2^{m-1}$. Next we will show how polar code construction complexity is reduced compared with [9].

Denote M the number of permutations that result in different polarized Bhattacharyya parameters, such as $M = (m!)/2^{m-1}$ in above discussion. In [9], each permutation among M candidates is employed to construct the corresponding polar code with length N. The best polar code that has minimum summation of Bhattacharyya parameter at information index is selected from M candidates. Since one time of Bhattacharyya parameter-based construction has complexity $O(N\log N)$, the total complexity of construction method in [9] is $O(MN\log N)$. However, in our proposed method, each permutation among M candidates is used to construct a sub-polar code with length m according to (C1). Therefore, the total complexity of the proposed method is $O(Mm\log m+N\log N)$, where the first term denotes that each permutation is employed to construct a length m code, and the second term represents that the permutation who satisfies (C1) is selected to run the construction of the entire length-N code. For example when $N = 1024$, $m = 4$, and $M = (m!)/2^{m-1} = 3$, the complexity of method in [9] is around $3 \times 1024 \times 10 = 30720$, while the proposed method has complexity $3 \times 4 \times 2 + 10240 = 10264$. If m is large when the order of modulation is higher, more complexity reduction can be achieved by the proposed method. When it comes to comparison with Mote Carlo construction, the complexity of the proposed scheme is still much lower since Monte Carlo construction has complexity $O(TN\log N)$, where T is the time of repeated simulations with the order of magnitude around 10^6. A brief complexity comparison between the proposed method and existing methods is given in Table C1.

After the solution to (C1) is obtained, polar codes under BICM scheme are constructed based on Algorithm 2. In Algorithm 2, line 1 means that problem (C1) is solved, and the resulting π is used to permute $\mathbf{Z} = \{Z(W(y|b_i))\}_{i=1}^{m}$. Lines 2-4 represent that the permuted version of $\mathbf{Z}$, i.e., $\mathbf{Z}_\pi$, is used as the Bhattacharyya parameters of the channels that coded

Table C1 Polar code construction complexity comparisons under BICM scheme

Construction methods	Asymptotic complexity
the propose method	$O(Mm\log m + N\log N)$, where $M = (m!)/2^{m-1}$
Search-space reduced method in [9]	$O(MN\log N)$, where $M = (m!)/2^{m-1}$
Monte Carlo construction	$O(TN\log N)$ with T of the order 10^6

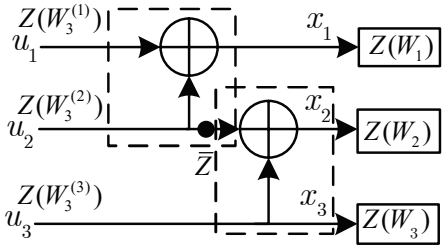

Figure C4 Bhattcharyya parameter under $\mathbf{F}_3$.

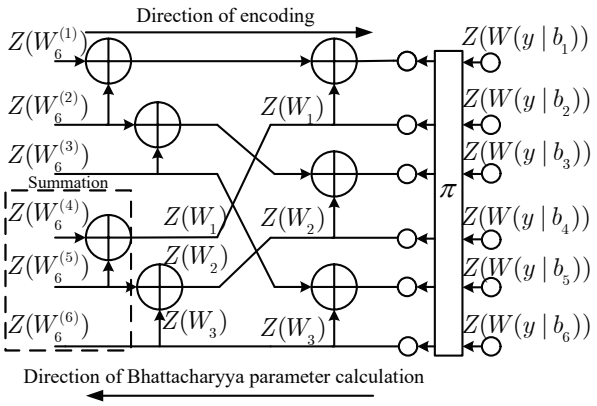

Figure C5 The process of calculating objective function in (C1) under 64-QAM.

bits $\mathbf{x}_1^N$ observe. Line 5 is the recursive code construction process using the upper bound in (A1). Lines 6-7 represent that the reliability of synthesized channels $W_N^{(i)}$, $1 \leqslant i \leqslant N$, is sorted and the K synthesized channels with best reliability are selected as information indices.

In the following subsection, 16-ASK and 64-QAM are taken as examples to illustrate the proposed channel mapping and code construction scheme.

Appendix C.3 Polar Coded 16-ASK and 64-QAM BICM

Under 16-ASK, $E_s/N_0 = 11$dB is selected for channel mapping and code construction. At $E_s/N_0 = 11$dB, according to Figure A2, the bit level Bhattacharyya parameter is $(Z(W(y|b_1)), ..., Z(W(y|b_4))) = (0.92, 0.65, 0.35, 0.18)$. By solving (C1), the optimal solution is $\pi = (\begin{smallmatrix}1&2&3&4\\4&1&3&2\end{smallmatrix})$. An example of polar coded 16-ASK BICM is shown in Figure C6 when code length $N = 8$.

Under 64-QAM, $E_s/N_0 = 12$dB is selected for channel mapping and code construction. At $E_s/N_0 = 12$dB, according to Figure A2, the bit level Bhattacharyya parameter is $(Z(W(y|b_1)), ..., Z(W(y|b_6))) = (0.20, 0.20, 0.40, 0.40, 0.71, 0.71)$. However, in 64-QAM, a single code word of polar codes based on only 2×2 kernel matrix cannot be transmitted by integer number of 64-QAM symbols. Therefore, in order to avoid puncturing as in [11], compound polar codes in [8] are adopted here by using both 2×2 and 3×3 kernel. The optimal 3×3 kernel is $\mathbf{F}_3 = [\begin{smallmatrix}1&0&0\\1&1&0\\0&1&1\end{smallmatrix}]$. $\mathbf{F}_3$ is optimal because it has maximum rate of polarization among all 3×3 matrices, which can be proved by Theorem 11 in [15].

The generator matrix of compound polar code using $\mathbf{F}_3$ is $\mathbf{G}_N = (\mathbf{I}_{N/3} \otimes \mathbf{F}_3)\mathbf{Q}(\mathbf{I}_3 \otimes (\mathbf{B}_{N/3}\mathbf{F}_2^{\otimes \log_2(N/3)})$, where $N = 3 \times 2^n$ and $\mathbf{Q}$ is a permutation matrix that acts on a length N vector $\mathbf{a}$. $\mathbf{Q}$ maps the element of $\mathbf{a}$ in the $3i + j$-th index to the $i + N/3 \times j$-th index. The decoding of compound polar codes with such generator matrix is given in [7, 8].

The recursive relationship of Bhattacharyya parameters under $\mathbf{F}_3$ is shown in the following lemma.

Lemma 1. $Z(W_3^{(1)})$, $Z(W_3^{(2)})$, and $Z(W_3^{(3)})$ are shorthand for $Z(W_3^{(1)}(\mathbf{y}_1^3|u_1))$, $Z(W_3^{(2)}(\mathbf{y}_1^3, u_1|u_2))$ and $Z(W_3^{(3)}(\mathbf{y}_1^3, \mathbf{u}_1^2|u_3))$, respectively. $Z(W_1)$, $Z(W_2)$, and $Z(W_3)$ are shorthand for $Z(W_1(y_1|x_1))$, $Z(W_2(y_2|x_2))$ and $Z(W_3(y_3|x_3))$, respectively. Then we have $Z(W_3^{(1)}) \leqslant \sum_{i=1}^3 Z(W_i) + \prod_{i=1}^3 Z(W_i) - \sum_{i=1}^3 \prod_{j \neq i}^3 Z(W_j)$, $Z(W_3^{(2)}) \leqslant Z(W_1)[Z(W_2) + Z(W_3) - Z(W_2)Z(W_3)]$, and $Z(W_3^{(3)}) = Z(W_2)Z(W_3)$. The equality in $Z(W_3^{(1)})$ and $Z(W_3^{(2)})$ holds when W_1, W_2 and W_3 are BECs.

Proof. Direct calculations for $Z(W_3^{(1)})$, $Z(W_3^{(2)})$, and $Z(W_3^{(3)})$ are relatively complex and let us think it in a different way. In Figure C4, this 3×3 matrix is decomposed into two 2×2 matrices, where the black dot with $\overline{Z}$ is the bridge that connects two 2×2 matrices. By replacing $Z(W_1)$ and $Z(W_2)$ in (A1) with $Z(W_2)$ and $Z(W_3)$, $Z(W_3^{(3)})$ is immediately obtained and we have $\overline{Z} = Z(W_2) + Z(W_3) - Z(W_2)Z(W_3)$. Once more, by replacing $Z(W_2)$ in (A1) with $\overline{Z}$, $Z(W_3^{(1)})$ and $Z(W_3^{(2)})$ are obtained. In (A1) the condition for equality is that W_1 and W_2 are both BECs. Therefore, in this lemma the condition for equality is that W_1, W_2, and W_3 are all BECs. This completes the proof.

Since the polarization kernel has changed with $\mathbf{F}_3$, the complexity of solving (C1) should be re-analyzed. When $m = 6$, the process of calculating the objective function in (C1) is shown in Figure C5.

When calculating the summation of Bhattacharyya parameters in the dashed line box in Figure C5, the upper bounds in lemma 1 are used, which implies the underlying channels are considered as BECs. It can be checked with the help of lemma 1 that $\sum_{i=4}^6 Z(W_6^{(i)}(\mathbf{y}_1^6, \mathbf{u}_1^{i-1}|u_i)) = \sum_{i=1}^3 Z(W_i)$. Therefore in Figure C5, once $Z(W_1)$, $Z(W_2)$, and $Z(W_3)$ are obtained, the value of the objective function is obtained by adding them. The search space of (C1) is all permutation π that results in different value of $\sum_{i=1}^3 Z(W_i)$. In Figure A2 we can see that $Z(W(y|b_i)) = Z(W(y|b_{i+1}))$, $i = 1$, 3 or 5. This property will further reduce the number of permutation π that results in different $\sum_{i=1}^3 Z(W_i)$. The counting of π can be divided into 3 cases (the symmetry in (A1) is also considered).

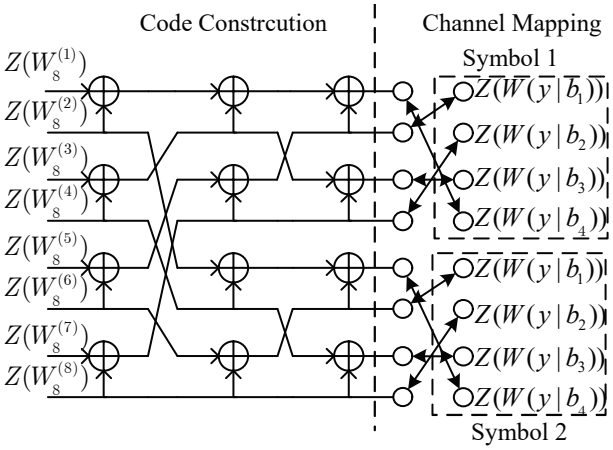

Figure C6 BICM 16-ASK with code length $N = 8$.

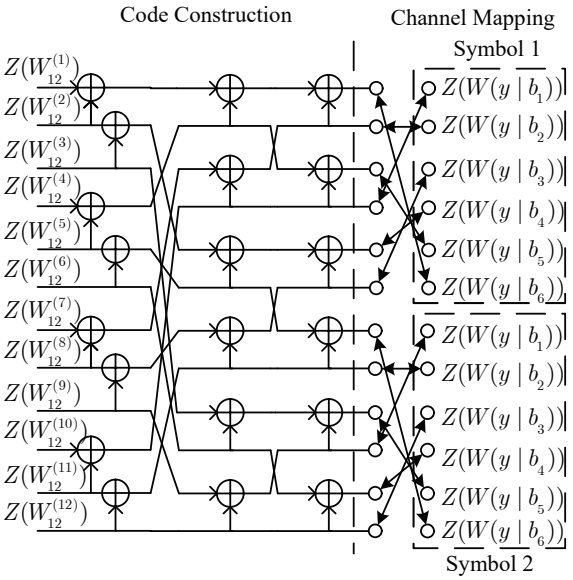

Figure C7 BICM 64-QAM with code length $N = 12$.

The first case is that every 2×2 polarization module in Figure C5 is assigned with $Z(W(y|b_i))$ and $Z(W(y|b_{i+1}))$, $i = 1$, 3 or 5. Any π that satisfies such condition will produce $\sum_{k=1}^{3} Z(W_k) = \sum_{i=1,3,5} Z(W(y|b_i))^2$. Hence only one permutation is needed among all such π.

The second case is that only one 2×2 polarization module is assigned with $Z(W(y|b_i))$ and $Z(W(y|b_{i+1}))$, $i = 1, 3$ or 5. In this case, assume that $Z(W(y|b_1))$ and $Z(W(y|b_2))$ are assigned to the topside 2×2 polarization module. Then in order to be different from circumstance 1, $Z(W(y|b_i))$ and $Z(W(y|b_{i+1}))$, $i = 3, 5$ cannot be assigned to one 2×2 polarization module. That is to say the remaining two 2×2 polarization modules must have Bhattacharyya parameter pair with different values. This circumstance produces three different values of $\sum_{k=1}^{3} Z(W_k) = Z(W(y|b_j))^2 + 2\prod_{l \neq j} Z(W(y|b_l))$, $l, j \in \{1, 3, 5\}$. Thus only three permutations are required.

The last case is that none of the three 2×2 polarization modules is assigned with two identical Bhattacharyya parameters. This will produce only one value of $\sum_{i=1}^{3} Z(W_i) = Z(W(y|b_1))Z(W(y|b_3)) + Z(W(y|b_1))Z(W(y|b_5)) + Z(W(y|b_3))Z(W(y|b_5))$ and hence only one permutation is needed. To summarize, only 5 permutations are required to solve (C1), which indicates the effectiveness of the proposed channel mapping algorithm.

With the help of lemma 1 and above analysis, (C1) can be solved. The solution is $\pi = \left(\begin{smallmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 6 & 2 & 5 & 1 & 4 & 3 \end{smallmatrix}\right)$. An example of polar coded 64-QAM modulation is shown in Figure C7 when code length $N = 12$.

Appendix D Simulation Results

In this section simulation results of the proposed polar code construction schemes are given. For both MLC and BICM scheme, the underlying channel that transmits modulation symbols is AWGN channel. The bit labeling rules are SP and Gray labeling for MLC and BICM, respectively.

Appendix D.1 BLER Performance under proposed MLC Scheme

In this subsection, BLER of the proposed polar code construction under MLC scheme is demonstrated. Polar coded 16-ASK modulation is considered and simulation results are shown in Figures D1 and D2. Figure D1 depicts BLER, while Figure D2 shows average decoding complexity for each component code under adaptive CA-SCL approach. Polar codes are constructed at $E_s/N_0 = 13$dB (corresponding Bhattacharyya parameter can be found in Figure A1 using $Z = 1 - C$) for 16-ASK. Each component code has length 256 and the total code rate is 0.5 with rate allocation 0/0.19922/0.83984/0.96094. According to this rate allocation, since the first component polar code with rate 0 is entirely frozen, there is no need to decode it, which reduces decoding latency. The 4-th component polar code has rate 0.96094. This rate is very high and not suitable to be further concatenated with CRC so it is just decoded by SC decoder. The second and third component polar codes are decoded by adaptive CA-SCL. Therefore, the maximum allowed list-size vector (LV) under such configuration is $LV = (1, L_{\max,2}, L_{\max,3}, 1)$, where LV_i is used for i-th component code. An 8-bit CRC $g(x) = x^8 + x^6 + x^3 + x^2 + 1$ [16] is used for CA-SCL. In Figure D1, it is assumed that $L_{\max,2} = L_{\max,3}$ because the 2-nd and 3-rd component codes have similar error rate according to Figure B8.

As contrary to Figure B7, it is observed that with the increase of $L_{\max,2}$ and $L_{\max,3}$, the BLER keeps improving. It can be seen through Figure D2 that the average decoding complexity drops quickly as E_b/N_0 increases. The complexity is defined by the total list sizes one component code uses, i.e., for i-th component code, if SC successes, then complexity is considered as 1. If adaptive CA-SCL successes with list size L, the complexity is considered as $1 + 2 + 4 + ... + L = 2L - 1$.

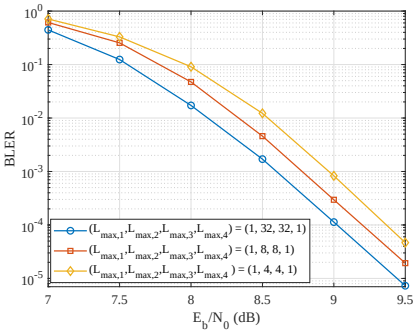

Figure D1 BLER of 16-ASK polar coded MLC under CA-SCL with different list size.

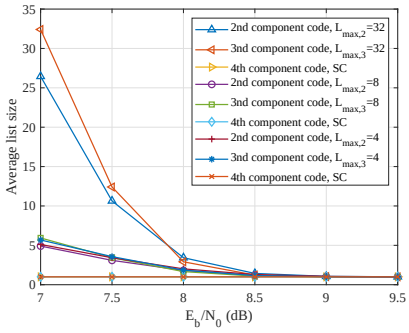

Figure D2 Average complexity of each component codes under 16-ASK polar coded MLC with different list size.

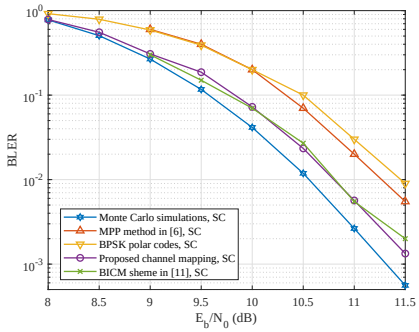

Figure D3 BLER of BICM 16-ASK polar coded modulation.

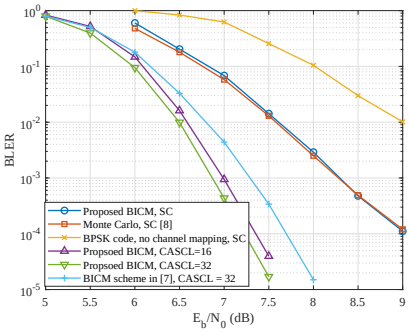

Figure D4 BLER of BICM 64-QAM polar coded modulation.

Appendix D.2 BLER Performance under BICM Scheme

In this subsection the BLER of the proposed polar code construction under BICM is given. In following context, BPSK polar code means that polar codes designed for BPSK are directly used in high order modulation without channel mapping. All BPSK polar codes are constructed using $Z(W(y|x)) = 0.3$.

We first focus on the BLER of BICM 16-ASK in Figure D3. For 16-ASK, we use polar codes with $N = 1024$ and $R = 0.5$ and all schemes in Figure D3 are decoded by SC. It can be observed that the BLER of the proposed polar code construction matches Monte Carlo scheme that is trained at $E_s/N_0 = 13$dB well in low SNR regime (8-9dB). When BLER $= 10^{-2}$, there is 0.25dB loss compared with Monte Carlo scheme. The proposed algorithm outperforms polar codes designed for BPSK modulation about 0.75dB and outperforms the MPP scheme in [6] around 0.5dB at BLER $= 10^{-2}$. Although the proposed method does not show obvious advantage over BICM scheme in [11], it should be noted that BICM scheme in [11] needs puncture when the required code length is not power of 2, while the proposed method does not need puncture by altering the polarization matrix.

In Figure D4, the proposed BICM 64-QAM algorithm is compared with Monte Carlo scheme in [8], BPSK polar codes, and polar coded BICM 64-QAM scheme in [7]. For 64-QAM, polar codes with length $N = 1536$ and rate $R = 0.5$ are simulated and the corresponding type of decoder is given in the legend. Compared with BPSK polar codes, the BLER performance is significantly improved by the proposed algorithm. In the simulation SNR range, the proposed algorithm matches the performance of compound polar codes that are constructed by Monte Carlo simulation in [8]. This confirms the effectiveness of the proposed polar code construction under SC. When it comes to CA-SCL decoder, 16-bit CRC $g(x) = (x + 1)(x^{15} + x + 1)$ [16] is used. It can be seen that when list size $L = 32$, the proposed scheme outperforms algorithms in [7], which confirms that the channel mapping for 64-QAM in this appendix is better than that in [7].

References

1 Arikan E. Channel polarization: a method for constructing capacity-achieving codes for symmetric binary-input memoryless channels. IEEE Trans Inf Theory, 2009, 55: 3051-3073
2 Wu D, Li Y, Sun Y. Construction and block error rate analysis of polar codes over awgn channel based on gaussian approximation. IEEE Commun Lett, 2014, 18: 1099-1102
3 Tal I, Vardy A. How to construct polar codes. IEEE Trans Inf Theory, 2013, 59:6562-6582

4 Seidl M, Schenk A. Stierstorfer C, et al. Polar-coded modulation. IEEE Trans Commun, 2013, 61: 4108-4119
5 Tavildar S. Bit-permuted coded modulation for polar codes. In: Proceedings of 2017 IEEE Wireless Communications and Networking Conference Workshops, San Francisco, 2017, 1-6
6 Shin D, Lim S, Yang K. Mapping selection and code construction for 2m-ary polar-coded modulation. IEEE Commun Lett, 2012, 16: 905-908
7 Chen P, Xu M, Bai B, et al. Design of polar coded 64-qam. In: Proceedings of International Symposium on Turbo Codes and Iterative Information Processing, Brest, 2016. 251-255
8 Mahdavifar H, El-Khamy M, Lee J, et al. Polar coding for bit-interleaved coded modulation. IEEE Trans Veh Technol, 2016, 65: 3115-3127
9 Lau F, Tam W. Reducing the bit-mapping search space of a bit-interleaved polar-coded modulation system. In: Proceedings of International Conference on Advanced Technologies for Communications, Quy Nhon, 2017. 198-203
10 Zhang Y, Liu A, Pan K, et al. A practical construction method for polar codes. IEEE Commun Lett, 2014, 18: 1871-1874.
11 Chen K, Niu K, Lin J R. An efficient design of bit-interleaved polar coded modulation. In: Proceedings of IEEE 24th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications, London, 2013. 693-697
12 Wachsmann U, Fischer R, Huber J. Multilevel codes: theoretical concepts and practical design rules. IEEE Trans Inf Theory, 1999, 45: 1361-1391
13 Caire G, Taricco G, Biglieri E. Bit-interleaved coded modulation. IEEE Trans Inf Theory, 1998, 44: 927-946
14 Li B, Shen H, Tse D. An adaptive successive cancellation list decoder for polar codes with cyclic redundancy check. IEEE Commun Lett, 2012, 16: 2044-2047
15 Korada S, Sasoglu E, Urbanke R. Polar codes: Characterization of exponent, bounds, and constructions. IEEE Trans Inf Theory, 2010, 56: 6253-6264
16 Koopman P, Chakravarty T. Cyclic redundancy code (CRC) polynomial selection for embedded networks. In: Proceedings of International Conference on Dependable Systems and Networks, Florence, 2004. 145-154
17 Pedarsani R, Hassani S, Tal I, et al. On the construction of polar codes. In: Proceedings of 2011 IEEE International Symposium on Information Theory Proceedings, St. Petersburg, 2011. 11-15
18 Tal I, Vardy A. List decoding of polar codes. IEEE Trans Inf Theory, 2015, 61: 2213-2226