GLDPC-PC Codes for MIMO Systems with Iterative Detection and Decoding

Binghui Shi, Yongpeng Wu, Yin Xu, Xiqi Gao, Xiaohu You and Wenjun Zhang

Abstract—**In this work, we propose the integration of GLDPC codes with short polar-like component codes, termed GLDPC codes with polar component codes (GLDPC-PC). This approach leverages the good distance properties of polar-like codes and mitigates their high decoding latency in long block lengths. A recently proposed soft-input soft-output decoder for polar-like codes enables effective iterative belief propagation decoding for GLDPC-PC, ensuring a low error floor under additive white Gaussian noise channels. Simulation results demonstrate that GLDPC-PC codes achieve significant performance improvements in multiple-input multiple-output systems with iterative detection and decoding (IDD). The proposed GLDPC-PC codes and the IDD scheme can be applied to various scenarios.**

Index Terms—**Generalized LDPC codes, soft-input soft-output decoding, polar-like codes, minimum mean-square error, iterative detection and decoding.**

I. Introduction

Generalized low-density parity-check (GLDPC) [2] code generalizes low-density parity-check (LDPC) code [1] by replacing the single parity-check (SPC) constraints with more general constraints based on stronger component codes. The belief propagation (BP) decoding of GLDPC codes iteratively pass extrinsic information between the variable nodes (VNs) and check nodes (CNs) in the Tanner graph where every CN represents a component code. The soft-input soft-output (SISO) decoder for the component code is necessary for BP decoding. One of the major obstacles in the development of GLDPC codes is the lack of low-complexity SISO decoders.

The optimal SISO decoding, called Bahl, Cocke, Jelinek and Raviv (BCJR) algorithm, is proposed in [3]. However, its computational complexity grows exponentially for linear block codes in the number of parity checks. [4] proposes an approximation based on list decoding but lacks precision and requires optimization of a weight factor and a saturation value. Whereas acquiring a SISO decoder with both satisfying complexity and precision for any linear block code is difficult, it can be achieved for specific code.

Polar-like codes represent a wide range of modifications of polar codes, whose codebook is obtained by performing a linear transformation on polar codebook [5]. For example, cyclic redundancy check (CRC) concatenated polar codes [6] exhibit excellent performance under successive cancellation list (SCL) decoding in short block length. Polarization-adjusted convolutional (PAC) codes [7] have been proven to possess better distance properties [8]. A recent work [9] introduces soft-output SCL (SO-SCL) to extract the bitwise soft information for polar-like codes based on the codebook probability.

Multiple-input multiple-output (MIMO) technology is widely used in current communication systems to achieve high data rates. A near-capacity receiver scheme is iterative detection and decoding (IDD) [10], which entails the SISO decoder for the channel coding. For the purpose of low complexity, minimum mean-square error (MMSE) parallel interference cancellation (PIC) [11] MIMO detection is proposed. Its sub-optimal performance further emphasizes the necessity of IDD scheme.

We propose a novel coding scheme, termed as GLDPC with polar component (GLDPC-PC) codes, by integrating the concept of GLDPC and SISO decoding with polar-like codes. This approach utilizes the superior distance properties of polar-like codes and avoids their high decoding complexity at longer block lengths, as the successive cancellation (SC)-based decoding for these codes requires a super-linear complexity. In the BP decoding for GLDPC-PC codes, the CNs are updated by SO-SCL. Furthermore, we analyze the performance of GLDPC-PC code in additive white Gaussian noise (AWGN) channel and MIMO IDD systems with this BP decoder. Although we focus on the point-to-point MIMO systems, the proposed GLDPC-PC codes and the IDD approach can be applied to various scenarios, e.g., multiuser MIMO and masssive MIMO systems.

This paper is organized as follows. Section II introduces the polar-like codes, their decoding and the basic concept of GLDPC. Section III introduces GLDPC-PC code and its BP decoding. The model of MIMO IDD system is given in Section IV. Section V shows the simulation results and Section VI concludes the paper.

II. Preliminaries

A. Notation

Vectors of length N are denoted as $\boldsymbol{x}$ or $x^N = (x_1, x_2, \cdots, x_N)$, where x_i is its i-th entry. For natural numbers, we write $[a] = \{i \in \mathbb{Z} \mid 1 \le i \le a\}$. For x^N, x^a where $a \in [N]$ denotes the sub-vector $(x_1, x_2, \cdots, x_a)$. Uppercase letters, e.g., X, denote random variables while lowercase letters, e.g., x, denote their realizations. The probability distribution of X evaluated at x is written as $P_X(x)$. Uppercase

B. Shi, Y. Wu, Y. Xu, and W. Zhang are with the Department of Electronic Engineering, Shanghai Jiao Tong University, Shanghai 200240, China. (e-mail: zeppoe@sjtu.edu.cn; yongpeng.wu@sjtu.edu.cn; xuyin@sjtu.edu.cn; zhangwenjun@sjtu.edu.cn). *(Corresponding author: Yongpeng Wu.)*

X. Gao and X. You are with the National Mobile Communications Research Laboratory, Southeast University, Nanjing 210096, China. (e-mail: xqgao@seu.edu.cn; xhyu@seu.edu.cn).

bold letters, e.g., $\mathbf{X}$, denote matrices. $\log$ denotes natural logarithm without special statement.

B. Polar-like Codes

A binary polar-like code of block length N and K information bits is defined by a set $\mathcal{A} \subseteq [N]$ with $|\mathcal{A}| = K$ and a transfer matrix $\mathbf{T}$ of size $N \times N$ over GF(2), where N is a positive-integer power of 2 and $\mathcal{F} \triangleq [N] \backslash \mathcal{A}$. The encoding of polar-like codes is implemented by two transforms

$$c^N = u^N \mathbf{T} \mathbf{P}_n, \tag{1}$$

where c^N is the codeword, u^N is the input of the two transforms and $\mathbf{P}_n = \mathbb{F}^{\otimes n}$ is the n-th Kronecker power of binary Hadamard matrix $\mathbb{F}$ with $n = \log_2 N$. The K information bits s^K are placed in the subvector $u_{\mathcal{A}}$ of u^N and $u_{\mathcal{F}}$ is set to 0. We demand $\mathbf{T} = \mathbf{I}_N + \mathbf{R}$, where $\mathbf{R}$ is a strictly upper triangular matrix and $\mathbf{R}_{j,i} = 1$ only if $j \in \mathcal{A}$ and $i \in \mathcal{F}$. Let $\mathcal{A}_i, i \in \mathcal{F}$ denote the indices of 1 in i-th column of $\mathbf{R}$. Obviously, if $j \in \mathcal{A}_i$, $j < i$ and $\mathcal{A}_i \subseteq \mathcal{A}$.

The type of frozen constraints where the frozen bits are set to 0 statically is called static frozen bits, i.e., $\mathbf{T} = \mathbf{I}_N$. The other type of frozen constraints where the frozen bits u_i are linear functions of u^{i-1} is called dynamic frozen bits, for instance [7],

$$\mathcal{A}_i = \{i-2, i-3, i-5, i-6\} \cap \mathcal{A}, i \in \mathcal{F}. \tag{2}$$

This representation includes various modifications of polar codes, e.g., CRC-concatenated polar codes [6], and dynamic Reed-Muller (RM) codes [12] and PAC codes [7]. After u^N is determined, the codeword c_N is obtained by polar transform and then transmitted through binary-input discrete memoryless channel (B-DMC) with channel output y^N.

C. SC-based Decoding for Polar-like Codes

A common decoding method for polar codes is SC decoding, which can be introduced through the concept of binary decision trees. Every node of stage i has two edges to the next stage and these two edges corresponds to the decision for $u_i = 0$ or $u_i = 1$. At a node of stage i, the SC decoding performs a greedy decision based on the estimation $\hat{u}^{i-1}$ and obtains a decision of u_i as

$$\hat{u}_i = \begin{cases} \bigoplus\limits_{j \in \mathcal{A}_i} u_j, & i \in \mathcal{F} \\ \underset{u \in \{0,1\}}{\arg\max} Q_{U_i|Y^N U^{i-1}}\left(u \mid y^N \hat{u}^{i-1}\right). & i \in \mathcal{A} \end{cases} \tag{3}$$

where $Q_{U_i|Y^N U^{i-1}}\left(u \mid y^N \hat{u}^{i-1}\right)$ denotes an auxiliary conditional probability mass function (PMF) induced by assuming uniform U^N over $\{0,1\}^N$ [9]. SC decoding successively adopts this greedy decision from the root node to a leaf node. A path from the root node to a leaf node is called a decoding path and all the decoding paths form a set $\mathcal{U}$, specifically,

$$\mathcal{U} \triangleq \left\{u^N \in \{0,1\}^N \mid u_i = f_i\left(u^{i-1}\right), \forall i \in \mathcal{F}\right\}. \tag{4}$$

SC decoding only searches one path of the binary decision tree. However, SC decoding has the ability to provide a path metric (PM) $Q_{U^i|Y^N}\left(\hat{u}^i \mid y^N\right)$ to weight different partial decoding paths $\hat{u}^i$. Therefore, it is possible to track more than one paths and avoid the greedy decision. SCL decoding implemented this idea via tracking L decoding paths with largest PM.

At the nodes of stage i, SCL decoding calculate the PM of the $2L$ new paths from the existing L paths and store L paths with largest PM in these $2L$ new paths. By the Bayes' rule,

$$\begin{aligned} & Q_{U^i|Y^N}\left(\hat{u}^i \mid y^N\right) \\ & = Q_{U^{i-1}|Y^N}\left(\hat{u}^{i-1} \mid y^N\right) Q_{U_i|Y^N U^{i-1}}\left(\hat{u}_i \mid y^N \hat{u}^{i-1}\right), \end{aligned} \tag{5}$$

where the term $Q_{U_i|Y^N U^{i-1}}\left(\hat{u}_i \mid y^N \hat{u}^{i-1}\right)$ can be computed efficiently by the standard SC decoding and $Q_{U^0|Y^N}\left(\varnothing \mid y^N\right) \triangleq 1$ by definition. Let $\mathcal{L}$ denote the set of L decoding paths when SCL decoding terminates and obviously $\mathcal{L} \subseteq \mathcal{U}$. [14] modifies the PM to $-\log Q_{U^i|Y^N}\left(\hat{u}^i \mid y^N\right)$ to change SCL decoding into log-likelihood ratio (LLR) domain and adopts several approximations to reduce the computational complexity of SCL.

III. GLDPC-PC and BP Decoding

A. Code Structure

GLDPC code extends LDPC code through replacing the SPC of every row in parity check matrix by several simply component codes [15]. GLDPC code consists of an adjacency matrix $\mathbf{\Gamma}$ of size $m \times N$ and m component codes with parity check matrix $\mathbf{H}_k, k \in [m]$. These m component codes can be identical or different. The parity check matrix $\mathbf{H}$ of GLDPC is obtained via replacing the j-th 1 in k-th row of $\mathbf{\Gamma}$ by the j-th column of $\mathbf{H}_k$. Therefore, the number of column of $\mathbf{H}_k$ must be equal to the number of 1 in k-th row of $\mathbf{\Gamma}$.

GLDPC-PC code refers to GLDPC code adopting polar-like codes as component code. The consideration of GLDPC-PC is to utilize the excellent performance and low-complexity SISO decoder of polar-like codes in short block length and avoid the high computational complexity of polar-like codes in long block length. The parity check matrix $\mathbf{H}_k$ of polar-like component codes can be conveniently obtained from $\mathbf{P}_n$ and $\mathbf{T}$. After replacement, the overall parity check matrix $\mathbf{H}$ is obtained. Gauss elimination is performed to $\mathbf{H}$ to eliminate the linearly dependent rows and obtain the generator matrix for systematic encoding.

[16] presents a GLDPC-PC code which has significantly lower error floors. The GLDPC-PC code adopts the (32,26) RM code as component code and adopts adjacency matrix

$$\mathbf{\Gamma}_0 = \begin{bmatrix} \gamma^0 & \gamma^0 & \gamma^0 & \cdots & \gamma^0 & \gamma^0 \\ \gamma^0 & \gamma^1 & \gamma^2 & \cdots & \gamma^{\tilde{n}-2} & \gamma^{\tilde{n}-1} \end{bmatrix}, \tag{6}$$

where γ is obtained by right rotation for 1 position of $\mathbf{I}_{\tilde{n}}$ and $\gamma^0 = \mathbf{I}_{\tilde{n}}$. $\mathbf{\Gamma}_0$ has quasi-cyclic (QC) structure and $\tilde{n} = 32$ since the number of 1 in every row of $\mathbf{\Gamma}_0$ must be 32. The component codes of first $\tilde{n}$ rows are the initial (32,26) RM code, and a random permutation is applied to the RM code for the component codes of last $\tilde{n}$ rows. For most cases, a (1024, 643) GLDPC-PC code is obtained after Gauss elimination.

$$L_i^{\mathrm{APP,Y}} = \log \frac{\sum_{c_i=0,c^N\in\mathcal{L}_C} Q_{C^N|Y^N}\left(c^N \mid y^N\right) + \left(Q_{\mathcal{U}}^*\left(y^N\right) - \sum_{c^N\in\mathcal{L}_C} Q_{C^N|Y^N}\left(c^N \mid y^N\right)\right) P_{C_i|Y_i}\left(0 \mid y_i\right)}{\sum_{c_i=1,c^N\in\mathcal{L}_C} Q_{C^N|Y^N}\left(c^N \mid y^N\right) + \left(Q_{\mathcal{U}}^*\left(y^N\right) - \sum_{c^N\in\mathcal{L}_C} Q_{C^N|Y^N}\left(c^N \mid y^N\right)\right) P_{C_i|Y_i}\left(1 \mid y_i\right)}$$
$$\text{where } \mathcal{L}_C = \left\{c^N \in \{0,1\}^N \mid c^N = u^N \mathbb{F}^{\otimes \log_2 N}, \forall u^N \in \mathcal{L}\right\}$$
$$\text{and } Q_{C^N|Y^N}\left(c^N \mid y^N\right) = Q_{U^N|Y^N}\left(u^N \mid y^N\right), \text{ for } c^N = u^N \mathbb{F}^{\otimes \log_2 N} \tag{9}$$

B. SISO Decoding for Polar-like Codes

BP decoder for GLDPC requires the SISO decoder for its component codes. Therefore, the SISO for polar-like codes is necessary for GLDPC-PC.

SISO decoding is a kind of decoding methods intended to minimize the bit error rate for a linear block code. A SISO decoder treats the sum of channel observation in LLR form L_i and a-priori information in LLR form L_i^{A} as input, where

$$L_i \triangleq \log \frac{p_{Y|C}\left(y_i \mid 0\right)}{p_{Y|C}\left(y_i \mid 1\right)},\ L_i^{\mathrm{A}} \triangleq \log \frac{P_{C_i}(0)}{P_{C_i}(1)}, i \in [N]. \tag{7}$$

The output of it is a-posteriori probability (APP) in LLR form and extrinsic information in LLR form, representing as L_i^{APP} and L_i^{E} respectively. To be specific

$$\begin{aligned} L_i^{\mathrm{APP}} &\triangleq \log \frac{P_{C_i|Y^N}\left(0 \mid y^N\right)}{P_{C_i|Y^N}\left(1 \mid y^N\right)} \\ &= \log \frac{\sum_{c_i=0,c^N\in\mathcal{C}} P_{C^N|Y^N}\left(c^N \mid y^N\right)}{\sum_{c_i=1,c^N\in\mathcal{C}} P_{C^N|Y^N}\left(c^N \mid y^N\right)}, \\ L_i^{\mathrm{E}} &\triangleq L_i^{\mathrm{APP}} - L_i^{\mathrm{A}} - L_i, i \in [N], \end{aligned} \tag{8}$$

where $\mathcal{C}$ is the codebook of the linear block code.

BCJR algorithm [3] realizes optimal SISO decoding for any linear code, but its complexity grows exponentially in the number of $N-K$ for block codes. Pyndiah's algorithm [4] and SO-SCL [9] are two methods to realize SISO decoder for polar-like codes in polynomial complexity.

Pyndiah's algorithm can apply to any linear block code and approximates L_i^{APP} through a candidate list $\mathcal{L}_C$ in (9) from SCL decoding,

$$L_i^{\mathrm{APP,P}} = \begin{cases} +\beta, \text{ if } \left\{c_i = 1, c^N \in \mathcal{L}_C\right\} = \varnothing \\ -\beta, \text{ if } \left\{c_i = 0, c^N \in \mathcal{L}_C\right\} = \varnothing \\ \log \frac{\max_{c_i=0,c^N\in\mathcal{L}_C} P_{C^N|Y^N}\left(c^N|y^N\right)}{\max_{c_i=1,c^N\in\mathcal{L}_C} P_{C^N|Y^N}\left(c^N|y^N\right)}, \text{ otherwise.} \end{cases} \tag{10}$$

In practical applications, a weighting factor α is multiplied to the extrinsic LLR $L_i^{\mathrm{E,P}}$ to reduce the influence of approximation. The factor α and the saturation value $\beta > 0$ require optimization via practical trials.

SO-SCL is designed for polar-like code. It fully utilizes the information of SCL decoder to provide a more precise approximation. Firstly, SO-SCL decoder approximates the sum of probabilities of all decoding paths in $\mathcal{U}$ and represents it as $Q_{\mathcal{U}}^*\left(y^N\right)$. Then $Q_{\mathcal{U}}^*\left(y^N\right)$ is adopted in soft output calculation (9) to obtain a more accurate approximation. Similar to Pyndiah's algorithm, a weighting factor α is adopted to the extrinsic LLR in practical use. The computational complexity of SO-SCL decoding is $\mathcal{O}\left(NL\log_2 N\right)$ since it dose not conduct any operation with complexity higher than SCL decoding.

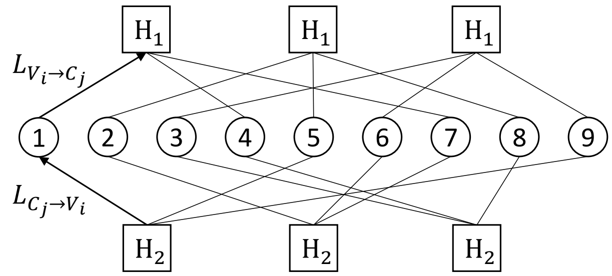

Fig. 1. The Tanner graph of $\mathbf{\Gamma}_0$ with $\tilde{n} = 3$. This graph has 9 VNs and 6 CNs. Every VN is connected to two codes with parity check matrix $\mathbf{H}_1$ and $\mathbf{H}_2$.

C. BP Decoding for GLDPC-PC

The BP decoding of GLDPC codes is performed through iteratively passing extrinsic information between the VNs and CNs in the Tanner graph of $\mathbf{\Gamma}$ [16]. The Tanner graph of $\mathbf{\Gamma}_0$ with $\tilde{n} = 3$ is shown in Fig. 1, where $\mathbf{H}_1$ and $\mathbf{H}_2$ are the parity check matrix of two different linear block codes with length 3. Let $L_{V_i \to C_j}, i \in [N], j \in \mathcal{N}\left(V_i\right)$ denote the extrinsic LLRs from VNs to CNs and $L_{C_j \to V_i}, j \in [m], i \in \mathcal{N}\left(C_j\right)$ denote the extrinsic LLRs from CNs to VNs, where $\mathcal{N}\left(V_i\right) = \left\{j \mid \mathbf{\Gamma}_{i,j} = 1\right\}, \mathcal{N}\left(C_j\right) = \left\{i \mid \mathbf{\Gamma}_{i,j} = 1\right\}$.

The input of BP decoder is $L_i^{\mathrm{in}} = L_i + L_i^A$ and the maximum number of iterations is set to R in advance. In every iteration, each VN $V_i, i \in [N]$ firstly sends its extrinsic information to its neighbor CNs via

$$L_{V_i \to C_j} = L_i^{\mathrm{in}} + \sum_{k\in\mathcal{N}(V_i), k\neq j} L_{C_k \to V_i}, \quad \forall j \in \mathcal{N}\left(V_i\right). \tag{11}$$

Afterwards, CNs possess SISO decoding of their component codes and send the output extrinsic LLRs to corresponding VNs, i.e., $L_{C_j \to V_i} = L_k^{\mathrm{E}}$ if V_i is the k-th VN of C_j. Finally, the APP LLRs is calculated by

$$L_i^{\mathrm{APP}} \approx L_i^{\mathrm{in}} + \sum_{k\in\mathcal{N}(V_i)} L_{C_k \to V_i}, \quad \forall i \in [N]. \tag{12}$$

The APP LLRs are only approximate values due to the cycles in $\mathbf{\Gamma}$. The decoder obtains $\hat{c}^N$ by hard decision based on L_i^{APP} and checks whether $\hat{c}^N$ is a codeword via whether the parity check of every component code is satisfied. The decoding terminates if $\hat{c}^N$ is a codeword or the number of iterations reaches R and the L_i^{APP} at this time is the final soft output.

For BP decoding, the SISO decoder of polar-like component codes can adopt Pyndiah's algorithm or SO-SCL. As described in [4], the weighting factor α and the saturation value $\beta > 0$ need optimization for every iteration for Pyndiah's algorithm. By SO-SCL, better performance is achieved even if α is constant, which is to be shown in simulations.

The BP with SO-SCL decoder realizes the SISO decoding for GLDPC-PC codes. The computational complexity of BP with SO-SCL is at most $\mathcal{O}\left(RmNL\log_2 N\right)$, where N is the largest block length of all the component codes. Whereas the complexity of SO-SCL decoder for polar-like code is higher than the SISO decoder for SPC, the constraint of polar-like codes is more powerful than SPC. Therefore, less iterations are required for GLDPC-PC code to obtain similar performance with LDPC code and the complexity increase can be compensated by this.

IV. Iterative MIMO Detection and Decoding

This section presents a point-to-point MIMO IDD system coded by GLDPC-PC codes. The details of the system are as follows.

A. System Model

Consider a point-to-point MIMO system with N_t transmit antennas and N_r receive antennas. The K bits message s^K is encoded into N bits codeword c^N. The systematic encoding of GLDPC code is applied to obtain optimal performance in IDD.

Consider the use of 16-quadrature amplitude modulation (QAM) with Gray labeling so the input alphabet is $\mathcal{X} = \frac{1}{\sqrt{10}}\{\pm a \pm jb | a, b \in \{1,3\}\}$. A random interleaver permutes the encoded bits c^N after the encoder and obtain b^N. The permuted codeword $b^N \in \{0,1\}^N$ are mapped into $T = N/N_t/M_c$ channel inputs $\boldsymbol{x}_t \in \mathcal{X}^{N_t}, t \in [T]$, where M_c is the modulation order and $M_c = 4$ for 16-QAM. Any $G = N_t M_c$ bits are mapped into a input vector $\boldsymbol{x} \in \mathcal{X}^{N_t}$ in a fixed way and let $\boldsymbol{x} = \mathrm{map}\left(b^G\right)$ denote this process. The total transmitted power is then obtained by $E_s = \mathbb{E}\left[\boldsymbol{x}^H\boldsymbol{x}\right] = N_t$. The transmitter model is shown in the upper area of Fig 2.

We consider the fading channel where the fading coefficients $\mathbf{H}_t \in \mathbb{C}^{N_r \times N_t}$ are different in every channel use. $\mathbf{H}_t$ is known perfectly to the receiver. For every channel use, the channel output is

$$\boldsymbol{y}_t = \mathbf{H}_t \boldsymbol{x}_t + \boldsymbol{n}_t, \quad t \in [T], \tag{13}$$

where $\boldsymbol{y}_t \in \mathbb{C}^{N_r}$ is the received signal vector and $\boldsymbol{n}_t \in \mathbb{C}^{N_r}$ is AWGN vector whose entries follow independent and identically distributed (i.i.d.) complex Gaussian distribution with zero-mean and variance $2\sigma^2$, i.e., $\mathcal{CN}\left(0, 2\sigma^2\right)$. The noise variance $2\sigma^2$ is known to the receiver by long term measurement.

B. Iterative Detection and Decoding

The IDD is implemented via exchanging extrinsic LLRs between MIMO detector and SISO decoder of channel coding [10]. The receiver model is shown in Fig. 2.

The exchange starts from the MIMO detector. The MIMO detector treats a-priori LLRs L_i^{A1} and received vector $\boldsymbol{y}_t$ as input and outputs APP LLRs L_i^{APP1}. L_i^{A1} is all zero in the beginning. If b_i is mapped into input vector $\boldsymbol{x}_t$ and is the i_G-th bit in $\boldsymbol{x}_t$, from [10] we have

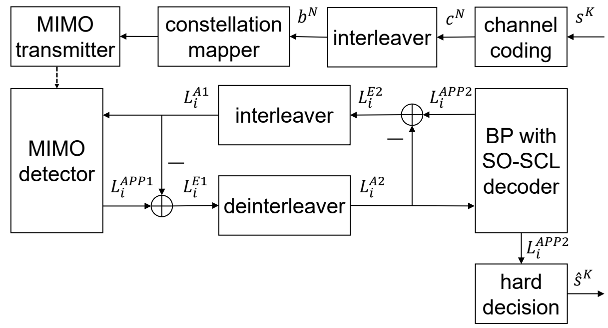

Fig. 2. Transmitter and receiver scheme with iterative detection and decoding. In the receiver, the extrinsic LLRs are exchanged between MIMO detector and BP decoder.

$$\begin{aligned} L_i^{\mathrm{A1}} &\triangleq \log \frac{P_{B_i}(0)}{P_{B_i}(1)}, \\ L_i^{\mathrm{APP1}} &\triangleq \log \frac{P_{B_i|Y^N}\left(0 \mid y^N\right)}{P_{B_i|Y^N}\left(1 \mid y^N\right)} \\ &= \log \frac{\sum_{\boldsymbol{x} \in \mathcal{X}_i^0} p_{\mathbf{Y}|\mathbf{X}}\left(\boldsymbol{y}_t \mid \boldsymbol{x}\right) \exp\left(\sum_{j \in \mathcal{J}_{\boldsymbol{x}}} L_j^{\mathrm{A1}}\right)}{\sum_{\boldsymbol{x} \in \mathcal{X}_i^1} p_{\mathbf{Y}|\mathbf{X}}\left(\boldsymbol{y}_t \mid \boldsymbol{x}\right) \exp\left(\sum_{j \in \mathcal{J}_{\boldsymbol{x}}} L_j^{\mathrm{A1}}\right)}, \end{aligned} \tag{14}$$

where $\mathcal{X}_i^0 = \left\{\boldsymbol{x} = \mathrm{map}\left(d^G\right) \mid d^G \in \{0,1\}^G, d_{i_G} = 0\right\}$, $\mathcal{X}_i^1 = \left\{\boldsymbol{x} = \mathrm{map}\left(d^G\right) \mid d^G \in \{0,1\}^G, d_{i_G} = 1\right\}$ and $\mathcal{J}_{\boldsymbol{x}} = \left\{j \mid \mathrm{map}\left(d^G\right) = \boldsymbol{x}, d_j = 0\right\}$. From (13),

$$p_{\mathbf{Y}|\mathbf{X}}\left(\boldsymbol{y} \mid \boldsymbol{x}\right) = \frac{1}{\left(2\pi\sigma^2\right)^N} \exp\left(-\frac{1}{2\sigma^2}\|\boldsymbol{y} - \mathbf{H}_t\boldsymbol{x}\|^2\right). \tag{15}$$

The input of SISO decoder is the result after deinterleaving the extrinsic information of MIMO detector L_i^{E1}. The APP LLRs L_i^{APP2} is output and the extrinsic LLRs L_i^{E2} is interleaved and treated as one of the inputs of MIMO detector. L_i^{APP2} and L_i^{E2} are defined in (8) and computed by the BP with SO-SCL decoder.

As shown in Fig. 2, conventional extrinsic information exchange is achieved by

$$\begin{aligned} L_i^{\mathrm{E1}} &= L_i^{\mathrm{APP1}} - L_i^{\mathrm{A1}}, \\ L_i^{\mathrm{E2}} &= L_i^{\mathrm{APP2}} - L_i^{\mathrm{A2}}. \end{aligned} \tag{16}$$

However, setting $L_i^{\mathrm{E2}} = L_i^{\mathrm{APP2}}$ can improve the performance for the MMSE-PIC detector adopted in this system [17]. Furthermore, we set $L_i^{\mathrm{E2}} = \rho L_i^{\mathrm{APP2}}, \rho \in (0, 1]$ to decrease the impact of this action.

Let R denote the total number of decoding iterations of BP. Let D denote the times of MIMO detection. From the process of IDD system, we can know that the BP decoder performs $N_I = R/D$ rounds of iteration and exchanges extrinsic LLRs once.

C. SISO MMSE-PIC Detection

The computation of the LLRs in (14) entails enormous computational complexity, which grows exponentially in the number of $N_t M_c$. Therefore, several sub-optimal algorithms are proposed to approximate (14) in polynomial complexity. The low-complexity SISO MMSE-PIC algorithm proposed in [18] is considered in this system.

For every input vector $\boldsymbol{x}_t$, received vector $\boldsymbol{y}_t$, a-priori LLRs $L_{i,j}^{\mathrm{A2}}, i \in [N_t], j \in [M_c]$ of the bits in $\boldsymbol{x}_t$ and channel $\mathbf{H}_t$, MMSE-PIC detection outputs the LLRs of the G bits in $\boldsymbol{x}_t$. It is summarized in the following five steps.

a) Computation of Soft-Symbols: Every input vector $\boldsymbol{x}_t$ possesses N_t symbols $x_i \in \mathcal{X}, i \in [N_t]$ and every symbol possesses M_c bits $b_{i,j} \in \{0,1\}, j \in [M_c]$. $L_{i,j}^{\mathrm{A2}}$ denotes the a-priori information of the j-th bit in symbol x_i. Based on the a-priori information, the algorithm firstly computes the expectation and variance of every symbol x_i by

$$\begin{aligned} \hat{x}_i &= \mathbb{E}[x_i] = \sum_{x \in \mathcal{X}} x P(x_i = x) \\ E_i &= \mathrm{Var}[x_i] = \mathbb{E}\left[|e_i|^2\right] \end{aligned} \tag{17}$$

where $e_i = x_i - \hat{x}_i$ and $P(x_i = x) = \prod_{j=1}^{M_c} P(b_{i,j} = d_j)$ if the j-th bit in x is d_j. LLR can be converted into probability by

$$P(b_{i,j} = d) = \frac{1}{1 + \exp\left((2d-1)L_{i,j}^{A}\right)}. \tag{18}$$

b) Parallel Interference Cancellation: The algorithm then cancels the interference for every symbol x_i induced by other symbols $x_j, j \neq i$ via

$$\hat{\boldsymbol{y}}_i = \boldsymbol{y}_t - \sum_{j, j \neq i} \mathbf{h}_j \hat{x}_j = \mathbf{h}_i x_i + \tilde{\mathbf{n}}_i \tag{19}$$

where $\mathbf{h}_j$ is the j-th column of $\mathbf{H}_t$ and $\tilde{\mathbf{n}}_i = \sum_{j,j\neq i} \mathbf{h}_j e_j + \mathbf{n}$ corresponds to the noise-plus-interference (NPI).

c) MMSE Filter-Vector Computation: All the MMSE filter vectors are computed by

$$\mathbf{W}^H = \mathbf{A}^{-1}\mathbf{H}_t^H, \tag{20}$$

where $\mathbf{A} = \mathbf{H}_t^H \mathbf{H}_t \mathbf{\Lambda} + N_0 \mathbf{I}_{N_t}$ with $\mathbf{\Lambda}$ denoting a $N_t \times N_t$ diagonal matrix having $\Lambda_{i,i} = E_i, i \in [N_t]$ and $N_0 = 2\sigma^2$. Every row $\mathbf{w}_i^H$ of $\mathbf{W}^H$ corresponds to the MMSE filter vectors for symbol x_i.

d) MMSE Filtering: The MMSE filter vector $\mathbf{w}_i^H$ are then applied to obtain the estimate of symbol x_i via

$$\tilde{x}_i = \frac{1}{\mu_i} \mathbf{w}_i^H \hat{\boldsymbol{y}}_i, \tag{21}$$

where $\mu_i = \mathbf{w}_i^H \mathbf{h}_i$ denotes the bias. The NPI variance ν_i can be computed as

$$\nu_i = \frac{1}{\mu_i} - E_i. \tag{22}$$

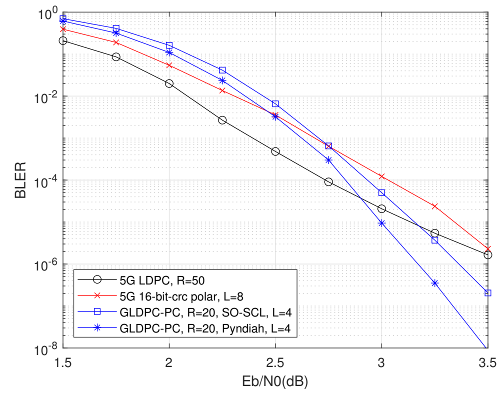

Fig. 3. BLER performance of GLDPC-PC code, 5G LDPC code and 5G polar code under AWGN channel. The block length N is 1024 and message bit length K is 643.

e) LLR Computation: With the estimates and NPI variances of all symbols, the extrinsic LLRs are calculated as

$$L_{i,j}^{\mathrm{E2}} = \frac{1}{\nu_i}\left(-\min_{a \in \mathcal{Z}_j^0} |\tilde{x}_i - a|^2 + \min_{a \in \mathcal{Z}_j^1} |\tilde{x}_i - a|^2\right). \tag{23}$$

where $\mathcal{Z}_j^b$ denotes the set of symbols in $\mathcal{X}$ which the j-th bit is equal to b.

V. Numerical Results

This section provides the Monte Carlo (MC) simulations for the GLDPC-PC under AWGN channel and MIMO IDD system. The complexity of BP decoding for GLDPC-PC and LDPC is balanced by reducing the number of iterations for GLDPC codes.

A. AWGN Channel

Fig 3 compares the (1024, 643) GLDPC-PC code in Section III-A under BP decoding with Pyndiah's algorithm and SO-SCL. The α for SO-SCL decoder is 0.6. The α and β for Pyndiah's algorithm are optimized for every iteration and E_b/N_0, so they are omitted here. The iteration round R is 20 and list size L for the both algorithms is 4. SO-SCL has better performance and is convenient to use, so in the following simulations we only consider SO-SCL.

Additionally, the (1024, 643) LDPC code and polar code in the 5-th generation wireless system (5G) are compared with the GLDPC-PC code. For LDPC code, BP decoding is adopted and maximum number R of iterations is set to 50. For polar code, a 16 bit CRC code is concatenated to the 5G polar code and SCL decoding is applied with list size $L = 8$.

Simulation results show that the GLDPC-PC code performs better than LDPC code below the block error rate (BLER) $\approx 3 \times 10^{-5}$ with SO-SCL. Because of the irregular property [19] of 5G LDPC code, the performance of it in low E_b/N_0 exceeds GLDPC-PC code. GLDPC-PC code with irregularity has the potential to perform better in this region.

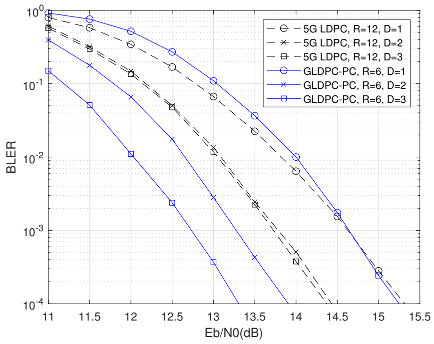

Fig. 4. BLER performance for (1024, 643) GLDPC-PC code and 5G LDPC code in 4×4 MIMO channel. $N_I = R/D$ rounds of BP is performed after once detection.

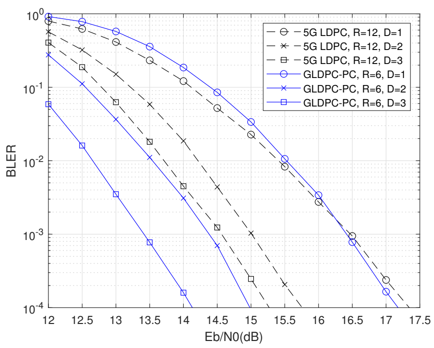

Fig. 5. BLER performance for (1024, 643) GLDPC-PC code and 5G LDPC code in 8×8 MIMO channel. $N_I = R/D$ rounds of BP is performed after once detection.

B. MIMO Channel

Rayleigh fading channel is considered in all the simulations, e.g., the entries of $\mathbf{H}_t$ are i.i.d. as $\mathcal{CN}(0,1)$ and $\mathbf{H}_t, t \in [T]$ are independent to each other. The relation of E_b/N_0 and E_s/N_0 follows [10]

$$\left.\frac{E_b}{N_0}\right|_{\text{dB}} = \left.\frac{E_s}{N_0}\right|_{\text{dB}} + 10\log_{10}\frac{NN_r}{KN_tM_c}. \tag{24}$$

We adopt the (1024, 643) GLDPC-PC code in Section III-A and compare it with the (1024, 643) 5G LDPC code. The number of antennas is set to $N_t = N_r = 4$ or $N_t = N_r = 8$ and 16-QAM is considered. R is set to 12 for LDPC code and 6 for GLDPC-PC code. R is consistent for all D for the fairness among different values of D. SO-SCL decoder is adopted with list size $L = 4$ and $\alpha = 0.6$ in BP decoding. The factor ρ is optimized for LDPC and is set to 0.6 for both LDPC and GLDPC.

As shown in Fig. 4 and Fig. 5, the GLDPC-PC code performs better than LDPC when $D = 1$ and BLER is low, which is consistent with the results in AWGN. Furthermore, GLDPC-PC outperforms LDPC when $D = 2$ and 3. At BLER $\approx 10^{-4}$ and $D = 3$, GLDPC-PC outperforms LDPC by about 1.1 dB for both 4×4 MIMO and 8×8 MIMO.

VI. Conclusion

In this work, we first introduce the GLDPC-PC codes. These codes exhibit significantly lower error floors, which brings performance gains in the low BLER region in AWGN channel. The BP decoder is introduced for GLDPC-PC codes, which adopts the low-complexity SO-SCL decoder for polar-like codes. We also adapt this BP with SO-SCL decoder in MIMO IDD systems coded by GLDPC-PC codes. Simulation results show the performance advantages of GLDPC-PC code with fewer rounds of decoding iteration. The GLDPC-PC code has the potential to perform better and exceed LDPC code with more elaborate construction.

References

[1] R. Gallager, "Low-density parity-check codes," *IRE Trans. Inf. Theory*, vol. 8, no. 1, pp. 21-28, Jan. 1962.

[2] R. Tanner, "A recursive approach to low complexity codes," *IEEE Trans. Inform. Theory*, vol. 27, pp. 533–547, Sep. 1981.

[3] L. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate (corresp.)," *IEEE Tran. Inf. Theory*, vol. 20, no. 2, pp. 284–287, Mar. 1974.

[4] R. Pyndiah, "Near-optimum decoding of product codes: block turbo codes," *IEEE Trans. Commun.*, vol. 46, no. 8, pp. 1003–1010, Aug. 1998.

[5] E. Arikan, "Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels," *IEEE Tran. Inf. Theory*, vol. 55, no. 7, pp. 3051-3073, Jul. 2009.

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

[7] E. Arıkan, "From sequential decoding to channel polarization and back again," *arXiv preprint arXiv:1908.09594*, 2019.

[8] Y. Li, H. Zhang, R. Li, J. Wang, G. Yan and Z. Ma, "On the weight spectrum of pre-transformed polar codes," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Melbourne, Australia, Jul. 2021, pp. 1224-1229.

[9] P. Yuan, K. R. Duffy, and M. Médard, "Near-optimal generalized decoding of Polar-like codes," *arXiv preprint arXiv:2402.05004*, 2024.

[10] B. M. Hochwald and S. ten Brink, "Achieving near-capacity on a multiple-antenna channel," *IEEE Trans. Commun.*, vol. 51, no. 3, pp. 389-399, Mar. 2003.

[11] X. Wang and H. V. Poor, "Iterative (turbo) soft interference cancellation and decoding for coded CDMA," *IEEE Trans. Commun.*, vol. 47, no. 7, pp. 1046–1061, Jul. 1999.

[12] M. C. Coşkun, J. Neu, and H. D. Pfister, "Successive cancellation inactivation decoding for modified reed-muller and ebch codes," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Los Angeles, CA, USA, Jun. 2020, pp. 437-442.

[13] E. Arikan, "Systematic polar coding," *IEEE Commun. Letters*, vol. 15, no. 8, pp. 860-862, Aug. 2011.

[14] A. Balatsoukas-Stimming, M. B. Parizi, and A. Burg, "LLR-based successive cancellation list decoding of Polar codes," *IEEE Trans. Signal Process.*, vol. 63, no. 19, pp. 5165–5179, Oct. 2015.

[15] G. Liva, W. E. Ryan and M. Chiani, "Quasi-cyclic generalized ldpc codes with low error floors," *IEEE Trans. Commun.*, vol. 56, no. 1, pp. 49-57, Jan. 2008.

[16] M. Lentmaier, G. Liva, E. Paolini, and G. Fettweis, "From product codes to structured generalized LDPC codes," in *Proc. 5th Int. ICST Conf. Commun. Netw. China*, Beijing, China, Aug. 2010, pp. 1–8.

[17] R. Wiesmayr, C. Dick, J. Hoydis and C. Studer, "DUIDD: Deep-unfolded interleaved detection and decoding for MIMO wireless systems," in *Proc. Conf. Rec. Asilomar Conf. Signals Syst. Comput. (ACSSC)*, Pacific Grove, CA, USA, 2022, pp. 181-188.

[18] C. Studer, S. Fateh and D. Seethaler, "ASIC implementation of soft-input soft-output MIMO detection using MMSE parallel interference cancellation," *IEEE J. Solid-State Circuits*, vol. 46, no. 7, pp. 1754-1765, Jul. 2011.

[19] T. J. Richardson, M. A. Shokrollahi and R. L. Urbanke, "Design of capacity-approaching irregular low-density parity-check codes," *IEEE Tran. Inf. Theory*, vol. 47, no. 2, pp. 619-637, Feb. 2001.