Iterative Detection and Decoding for MIMO Systems with Knowledge-Aided Belief Propagation Algorithms

Jingjing Liu, Peng Li and Rodrigo C. de Lamare
Department of Electronics, The University of York
Heslington, York, YO10 5DD, UK
Email: {jl622; pl534; rcdl500}@york.ac.uk

***Abstract*—In this paper, we consider the problem of iterative detection and decoding (IDD) for multi-antenna systems using low-density parity-check (LDPC) codes. The proposed IDD system consists of a soft-input soft-output parallel interference (PIC) cancellation scheme with linear minimum mean-square error (MMSE) receive filters and two novel belief propagation (BP) decoding algorithms. The proposed BP algorithms exploit the knowledge of short cycles in the graph structure and the reweighting factors derived from the hypergraph's expansion. Simulation results show that when used to perform IDD for multi-antenna systems both proposed BP decoding algorithms can consistently outperform existing BP techniques with a small number of decoding iterations.**

I. Introduction

Multi-input and multi-output (MIMO) systems can support several independent data streams, resulting in a significant increase of the system capacity [1]. In order to separate the data streams and mitigate the interference between them, a detection algorithm must be employed at the receiver. In the last decade or so, a great deal of effort has been devoted to the development of detection algorithms and their integration with channel decoding techniques [2]-[12]. In this context, MIMO systems with joint detection/decoding have been shown to produce excellent results, approaching the performance of an interference free scenario. In a system with joint detection/decoding an ideal receiver is comprised of two components: an efficient soft-input soft-output (SISO) MIMO signal detector and a SISO decoder with low delay. Specifically, the estimated log likelihood ratios associated with the encoded bits are computed by the detector and these estimates will serve as input to the decoder. Then in the second phase of the detection/decoding iteration, the decoder generates *a posteriori* probabilities for encoded bits of each data stream. As a result, the soft estimate of the transmitted symbol is obtained which can facilitate the detection in the first phase of the next outer iteration. The joint process of detection/decoding is then repeated in an iterative manner until the maximum number of iterations is reached. However, in practice there are many open issues for such an IDD scheme, e.g. severe detection/decoding delay especially for codes with short block lengths [4], [5], or prohibitively high computational complexity associated with IDD systems in general.

Low-density parity-check (LDPC) codes, invented by Gallager [13] are a class of linear block codes which can achieve near-Shannon capacity with linear-time encoding and parallelizable decoding algorithms. The standard BP algorithm is well-known as the most effective algorithm to decode LDPC codes [14], and has been widely employed as part of IDD schemes for MIMO systems [4], [7] and [15]. It can produce exact inference solutions only if the graphical model does not contain short cycles. With the existence of cycles, the standard BP algorithm has a number of shortcomings, such as convergence to a codeword is not guaranteed and convergence to a codeword can take many iterations, especially at low signal to noise ratios (SNR), which significantly deteriorate the decoding performance and cause unexpected transmission delay. Due to this fact, many applications of LDPC-coded MIMO systems have a performance degradation at some extent. In [16], the authors converted the problem of finding the fixed points of BP algorithms into that of solving a variational problem, and defined a set of reweighting factors. Recently, Wymeersch et al. [17] extended the use of reweighted BP algorithm from pairwise graphs to hypergraphs and reduced the set of reweighted parameters to a constant value, whereas Liu and de Lamare [18] considered two possible values.

In this paper, we develop an efficient IDD scheme for MIMO systems operating in a spatial multiplexing configuration with a reduced complexity and a low delay. The proposed scheme consists of a SISO parallel interference cancellation (PIC) scheme with linear minimum mean-square error (MMSE) receive filters and two novel knowledge-aided (KA) belief propagation (BP) decoding algorithms. The first KA decoding algorithm is termed cycles knowledge-aided reweighted BP (CKAR-BP) algorithm, whereas the second KA decoding techniques is called expansion knowledge-aided reweighted BP (EKAR-BP) algorithm. In the following, we present an IDD scheme for MIMO systems equipped with the proposed KA BP algorithms which can considerably improve the performance of existing schemes. The proposed CKAR-BP decoder takes advantage of the cycle distribution of the Tanner graph, while the proposed EKAR-BP decoder first expands the original graph into a number of subgraphs then locally optimizes the reweighting parameters. Incorporated with a SISO PIC-MMSE detector, both CKAR-BP and EKAR-BP algorithms are shown to outperform the standard BP and the uniformly reweighted BP (URW-BP) [17] algorithms when performing IDD for MIMO systems.

The organization of this paper is as follows: Section II introduces the system model. In Section III, the proposed EKAR-BP and CKAR-BP algorithms are explained in detail. Section IV shows the simulation results along with discussions. Finally, Section V concludes the paper.

II. SYSTEM MODEL

Let us consider a narrowband MIMO system with N_T transmit antennas and N_R receive antennas ($N_R \geq N_T$). The MIMO system operates in a spatial multiplexing configuration and transmits data over flat fading channels. The received data after demodulation, matched filtering and sampling is collected in a vector $\boldsymbol{r} \in \mathbb{C}^{N_R \times 1}$ with sufficient statistics for detection and given by

$$\boldsymbol{r} = \boldsymbol{C}\boldsymbol{s} + \boldsymbol{n}, \tag{1}$$

where $\boldsymbol{C} \in \mathbb{C}^{N_R \times N_T}$ is the channel matrix, $\boldsymbol{s} \in \mathbb{C}^{N_T \times 1}$ is the encoded data vector and $\boldsymbol{n} \in \mathbb{C}^{N_R \times 1}$ is the noise vector with zero mean and power σ_n^2 elements. In what follows, we assume that the receiver has perfect knowledge of the channel matrix $\boldsymbol{C}$. In practice, an estimation algorithm must be employed to compute the parameters of $\boldsymbol{C}$ [10], [12].

A. PIC-MMSE Detection Algorithm

In a SISO PIC-MMSE detection algorithm, the estimates of the transmitted symbols are obtained based on the *a priori* log-likelihood ratios (LLRs) obtained from the LDPC channel decoder. These "soft" estimates are extracted from the received vector to perform interference cancellation for a MIMO system. The remaining noise-plus remaining interference terms are then equalized by a linear MMSE receive filter which is followed by the computation of the a posteriori LLRs of the individual constituent bits. The SISO PIC-MMSE algorithm used as an outer component is detailed in the following.

According to the SISO model in [2], when processing the kth stream, a PIC detector cancels the interference of all other streams ($q \neq k$) such that

$$\hat{\boldsymbol{r}}_k = \boldsymbol{r} - \sum_{q \neq k} \boldsymbol{c}_q \hat{y}_q = \boldsymbol{c}_k s_k + \tilde{\boldsymbol{n}}, \quad \forall k \tag{2}$$

where $y_q, q \neq k$ are the estimates of the transmitted co-channel symbols obtained from the channel decoder which are computed according to $\hat{y}_q = E[y_q] = \sum_{a \in \mathcal{O}} P[y_q = a]a$ where $P[y_q = a]$ corresponds to the a priori probability of the symbol a on the constellation map $\mathcal{O}$. The term $\boldsymbol{c}_k$ is the kth column of the channel matrix $\boldsymbol{C}$ and $\tilde{\boldsymbol{n}}$ is the noise-plus-remaining-interference vector to be equalized by linear MMSE receive filters as

$$\hat{y}_k = \tilde{\boldsymbol{w}}_k^H \hat{\boldsymbol{r}}_k = \tilde{\boldsymbol{w}}_k^H \boldsymbol{c}_k s_k + \tilde{\boldsymbol{w}}_k^H \tilde{\boldsymbol{n}}, \tag{3}$$

in which '$(\cdot)^H$' denotes the Hermitian transpose and the MMSE receive filter is given by $\tilde{\boldsymbol{w}}_k^H = E_s \boldsymbol{c}_k^H \big(\boldsymbol{C}\tilde{\boldsymbol{\Lambda}}_k \boldsymbol{C}^H + N_0 \boldsymbol{I}_{N_R}\big)$, where E_s is the transmission energy and $\tilde{\boldsymbol{\Lambda}}_k \in \mathbb{C}^{N_T \times N_R}$ is a diagonal matrix whose entries are the variances of the estimation errors.

B. Iterative Detection and Decoding

A block diagram of the IDD system employed in this work is depicted in Fig. 1. With the PIC-MMSE processing, we set $y_k = s_k + n_{\text{eff}}$ at the output of the detector, where n_{eff} is the effective noise factor after the MMSE filtering. By assuming that the output of the k-th layer y_k is statistically independent from the other layers [2], this leads to the approximation of the log-likelihood ratio (LLR) of bit $x_{k,j}$

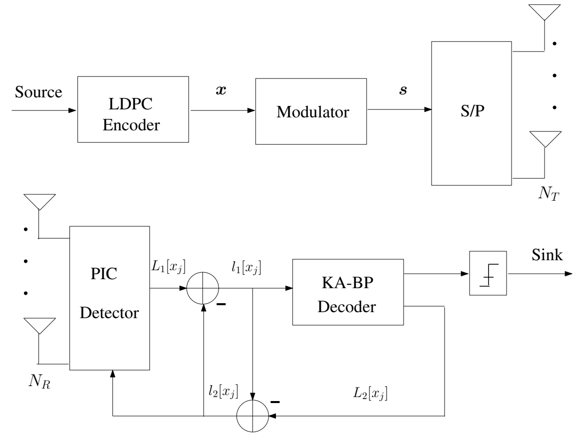

Fig. 1. Iterative LDPC-coded MIMO spatial multiplexing system with a SISO PIC-MMSE detector and the proposed KA-BP decoders.

$$\Lambda_1[x_{k,j}] \approx \log \frac{P(x_{k,j} = +1 | y_k)}{P(x_{k,j} = -1 | y_k)} = \lambda_1[x_{k,j}] + \lambda_2^p[x_{k,j}], \tag{4}$$

where the last term represents the *a priori* information for the coded bits $x_{k,j}$, which is obtained by the LDPC decoder. The first term λ_1 denotes the *extrinsic* information which is computed based on $\boldsymbol{r}$ and the *a priori* information λ_2^p. For the detector, by relaxing the stream index k, the coded bit *extrinsic* LLR is obtained as

$$\lambda_1[x_j] = \log \frac{\sum_{a_c \in \mathcal{A}_j^+} P(y|s = a_c) \exp(L_a(a_c))}{\sum_{a_c \in \mathcal{A}_j^-} P(y|s = a_c) \exp(L_a(a_c))} \tag{5}$$

where $\mathcal{A}_j^+$ and $\mathcal{A}_j^-$ denotes the subsets of constellation $\mathcal{A}$ where the bit x_j takes the values 1 and 0, respectively. The value $L_a(a_c)$ denotes the *a priori* symbol probability for symbol a_c and

$$P(y|s = a_c) = \frac{1}{\pi \sigma_{\text{eff}}^2} \exp\Big(\frac{-|y - s|^2}{\sigma_{\text{eff}}^2}\Big) \tag{6}$$

For an IDD scheme, the computed λ_1 is fed to the LDPC decoder as the *a priori* information. The LDPC decoder calculates the *a posteriori* LLR of each code bit as will be detailed later.

III. KNOWLEDGE-AIDED DECODING ALGORITHMS FOR IDD SCHEMES

The proposed CKAR-BP and EKAR-BP algorithms are designed to improve the convergence behaviour of the standard BP algorithm by reweighting part of the hypergraph. These algorithms take the short cycles into account, such that the decoder can generate more accurate marginal distributions corresponding to coded data. The reweighting strategy was first employed in the tree-reweighted BP (TRW-BP) algorithm reported in [16], where the authors reformulated the BP decoding problem into a tractable convex optimization problem that iteratively computes beliefs and factor appearance probabilities (FAPs). Later with the same concept but additional constraints, the uniformly reweighted BP (URW-BP) algorithm [17] was introduced for which the FAPs were constrained to be a constant. A disadvantage of URW-BP is that it can only be applied to regular LDPC codes. Compared to those two methods, CKAR-BP and EKAR-BP algorithms optimize the FAPs off-line by relaxing the constraints from [16] and [17]. Additionally, neither of them impose extra computational complexity to online decoding. Next, we present general message passing rules for reweighted BP algorithms, then elaborate both CKAR-BP and EKAR-BP decoders.

A. Message Passing Rules for Knowledge-Aided Decoders

The message passing rules of reweighted BP algorithms are briefly reviewed here, the derivation of which can be found in [16] with pairwise interactions and in [17] with higher-order interactions. Given a hypergraph having N variable nodes and M check nodes and the reweighting vector $\boldsymbol{\rho} = [\rho_1, \rho_2, \ldots, \rho_M]$, the message from the j-th variable node s_j to the i-th check node c_i is given by

$$\Psi_{ji} = \lambda_{\mathrm{In},j} + \sum_{i' \in \mathcal{N}(j) \backslash i} \rho_{i'} \Lambda_{i'j} - (1-\rho_i)\Lambda_{ij}, \qquad (7)$$

where $i' \in \mathcal{N}(j) \backslash i$ is the neighboring set of check nodes of s_j except c_i. Since all messages are represented in LLRs, $\lambda_{\mathrm{In},j}$ is equal to $l_1[x_j$ in the first decoding iteration. We use Λ_{ij} to denote messages sent from c_i to s_j in previous decoding iterations, then for check nodes c_i Λ_{mn} is updated as

$$\Lambda_{ij} = 2\tanh^{-1}\Big(\prod_{j' \in \mathcal{N}(i) \backslash j} \tanh \frac{\Psi_{j'i}}{2} \Big), \qquad (8)$$

where '$\tanh(\cdot)$' denotes the hyperbolic tangent function as in the standard BP message passing rule to compute an LLR message from check node c_i to variable node s_j. Finally, we have the belief $b(x_j)$ with respect to x_j given by

$$b(x_j) = \lambda_{\mathrm{In},j} + \sum_{i \in \mathcal{N}(j)} \rho_i \Lambda_{ij}. \qquad (9)$$

The proposed KA-BP decoders iteratively employ (7)-(9) to update the message regarding each node. At the end of decoding, $\lambda_{\mathrm{Belief},j}$ serves as the soft output for deciding the value of $\hat{x}_j$ or for generating the extrinsic information $l_2[x_j]$ in the next IDD iteration. Notice that $\rho_i = 1, \forall i$ corresponds to the standard BP algorithm so that no additional complexity is introduced due to the presence of $\boldsymbol{\rho}$ in real-time decoding.

B. Cycles Knowledge-Aided Reweighted BP (CKAR-BP)

Given the knowledge of the distribution of cycles in the graph, the CKAR-BP algorithm selects the reweighting parameters in order to mitigate the effect of short cycles, i.e. the statistical dependency among the incoming messages being exchanged by nodes, leading to a situation in which the outgoing messages inaccurately have a high reliability or equivalently a low quality. The algorithm [19], used for counting short cycles, is a matrix multiplication technique which can find the girth g implicitly and calculate the number of cycles with length of g, $g{+}2$ and $g{+}4$, explicitly. As shown in Table I, after running the algorithm for counting cycles and calculating μ_g the average number of length-g cycles passing a check node, we determine the reweighting parameters $\rho_i (i = 0, 1, \ldots, M-1)$ under a simple criterion:

TABLE I

Algorithm Flow of CKAR-BP Decoder

Offline Stage 1: counting short cycles

1: Run the algorithm [19] to count the number of cycles with length-g passing the check node $c_i, \forall i$;

Offline stage 2: determination of ρ_i for the hypergraph

2: Determine variable FAPs for each check node:
if $g_{C_i} < \mu_g$ $\rho_i = 1$, otherwise $\rho_i = \rho_v$ where $\rho_v = 2/\bar{n_D}$;

Online Stage: real-time decoding

3: Update the belief $b(x_j)$ iteratively using reweighted message passing rules (7)–(9) with optimized $\boldsymbol{\rho} = [\rho_1, \rho_2, \ldots, \rho_M]$. Decoding stops if $\boldsymbol{H}\hat{\boldsymbol{x}}^{\boldsymbol{T}} = \boldsymbol{0}$ or the maximum number of decoding iterations is reached.

$$\rho_i = \begin{cases} 1 & \text{if} \quad g_{C_i} < \mu_g, \\ \rho_v & \text{otherwise}, \end{cases} \qquad (10)$$

where g_{C_i} is the number of length-g cycles passing a check node C_i, $\rho_v = 2/\bar{n_D}$ and $\bar{n_D}$ is the average connectivity for N variable nodes, which is computed by:

$$\bar{n_D} = \frac{1}{\int_0^1 \upsilon(x)dx} = \frac{M}{N\int_0^1 \nu(x)dx}, \qquad (11)$$

where $\upsilon(x)$ and $\nu(x)$ are distributions of the variable nodes and the check nodes, respectively. As an improvement to the URW-BP algorithm [17], the proposed CKAR-BP requires additional complexity due to the cycle counting algorithm [19]. Most importantly, CKAR-BP algorithm can improve the performance of the BP algorithm when decoding LDPC codes with both uniform structures (regular codes) and with non-uniform structures (irregular codes). More details of CKAR-BP and its applications can be found in [18].

C. Expansion Knowledge-Aided Reweighted BP (EKAR-BP)

The proposed EKAR-BP algorithm transforms the original hypergraph $\mathcal{G}$ into a set of $T \geq 1$ subgraphs and then locally optimizes the reweighting parameter vector $\boldsymbol{\rho}_t, t = 1, 2, \ldots, T$ with respect to each subgraph, where the size of the t-th subgraph determines the dimension of $\boldsymbol{\rho}_t$. It should be noted that $T = 1$ corresponds to the original TRW-BP algorithm [16] which has a computational complexity of $\mathcal{O}(M^2N)$ and the convergence of $\boldsymbol{\rho}$ is very slow for large graphs. Nevertheless, the optimization of $\boldsymbol{\rho}$ could be significantly less complex when more subgraphs are considered ($\boldsymbol{\rho}$). Thus, there is a need for a flexible method to decompose the original hypergraph into subgraphs. Inspired by [20], we apply a modified progressive-edge growth (PEG) approach to achieve the hypergraph expansion. Generally, the number of subgraphs T depends on a pre-set maximum expansion level $d_{\max}$, as a large $d_{\max}$ results in a small T but a high probability of existence of very short cycles within subgraphs. Compared to the greedy version of

TABLE II

Algorithm Flow of EKAR-BP Decoder

Offline Stage 1: subgraphs formation

1: Given a hypergraph $\mathcal{G}$ and $d_{\max}$, apply the modified PEG expansion to generate $T \geq 1$ subgraphs;

Offline Stage 2: optimization of $\boldsymbol{\rho}_t$ for the t-th subgraph

2: Initialize $\boldsymbol{\rho}_t^{(0)}$ to a valid value;

3: For each subgraph, calculate the beliefs $b(\boldsymbol{x}_t)$ and the mutual information term $\boldsymbol{I}_t = [I_{t,1}, I_{t,2}, \ldots, I_{t,L_t}]$ by using reweighted message passing rule (7)–(9);

4: With $b(\boldsymbol{x}_t)$ and $\boldsymbol{I}_t$ obtained from step 3, update $\boldsymbol{\rho}_t^{(r)}$ to $\boldsymbol{\rho}_t^{(r+1)}$ using the conditional gradient method;

5: Repeat steps 3–4 until $\boldsymbol{\rho}_t$ converges for each subgraph;

Offline Stage 3: choice of $\boldsymbol{\rho} = [\rho_1, \rho_2, \ldots, \rho_M]$ for decoding

6: For all T subgraphs, collect $\boldsymbol{\rho}_1, \ldots, \boldsymbol{\rho}_i, \ldots, \boldsymbol{\rho}_T$. In case of multiple values ρ_i for the same i-th check node, choose the one offering the best performance;

Online Stage: real-time decoding

7: Update the belief $b(x_j)$ iteratively using reweighted message passing rules (7)–(9) with the optimized $\boldsymbol{\rho} = [\rho_1, \rho_2, \ldots, \rho_M]$. Decoding stops if $\boldsymbol{H}\hat{\boldsymbol{x}}^{\boldsymbol{T}} = \boldsymbol{0}$ or the maximum number of decoding iterations is reached.

PEG [20], our modified PEG expansion has two differences: (i) the expansion stops as soon as every member of the set of nodes V_t has been visited; (ii) the number of edges incident to the node s_j might be less than its degree since some short cycles are excluded in subgraphs to guarantee that the local girth of each subgraph g_t is always larger than the global girth of the original graph g.

As shown in Table. II, after obtaining T subgraphs, we introduce $\boldsymbol{L} = [L_1, L_2, \ldots, L_T]$ in which L_t is the number of check nodes in the t-th subgraph. Note that $\sum_t L_t > M$ due to duplicated nodes during hypergraph expansion. With the t-th subgraph, we optimize the associated FAPs $\boldsymbol{\rho}_t = [\rho_{t,1}, \rho_{t,2}, \ldots, \rho_{t,L_t}]$ using a recursive optimization method, similar to TRW-BP [16] but with higher-order interactions and related message passing rules (7)–(9). The optimization problem is solved recursively as follows: 1) for all T subgraphs in parallel and fixed $\boldsymbol{\rho}_t^{(r)}$, use message passing rules (7)–(9) to calculate the beliefs $b(\boldsymbol{x}_t)$ as well as the mutual information term $\boldsymbol{I}_t = [I_{t,1}, I_{t,2}, \ldots, I_{t,L_t}]$ provided with $L_t \leq M$ check nodes in the t-th subgraph; 2) for all T subgraphs in parallel, given $\{\boldsymbol{I}_t\}_{t=1}^T$, use the conditional gradient method to update, for all t, $\boldsymbol{\rho}_t^{(r)}$ to $\boldsymbol{\rho}_t^{(r+1)}$, then go back to step 1).

The optimization problem is given by

$$\begin{aligned} \text{minimize} \quad & -\boldsymbol{\rho}_t^{\dagger}\boldsymbol{I}_t \\ \text{s.t.} \quad & \boldsymbol{\rho}_t \in \mathbb{T}(\mathcal{G}_t), \end{aligned}$$

where $(\cdot)^{\dagger}$ denotes matrix transpose, $\mathbb{T}(\mathcal{G}_t)$ is the set of all valid FAPs over the subgraph $\mathcal{G}_t$ and $I_{t,l}$ is a mutual information term depending on $\boldsymbol{\rho}_t^{(r)}$, the previous value of $\boldsymbol{\rho}_t$. By denoting the objective function as $f(\boldsymbol{\rho}_t) = -\boldsymbol{\rho}_t^{\dagger}\boldsymbol{I}_t$, we first linearize the objective around the current value $\boldsymbol{\rho}_t^{(r)}$:

$$f_{\text{lin}}(\boldsymbol{\rho}_t) = f(\boldsymbol{\rho}_t^{(r)}) + \nabla_{\boldsymbol{\rho}_t}^{\dagger} f(\boldsymbol{\rho}_t^{(r)})(\boldsymbol{\rho}_t - \boldsymbol{\rho}_t^{(r)}), \qquad (12)$$

where$\nabla_{\boldsymbol{\rho}_t} f(\boldsymbol{\rho}_t^{(r)}) = -\boldsymbol{I}_t$. Secondly, we minimize $f_{\text{lin}}(\boldsymbol{\rho}_t)$ with respect to $\boldsymbol{\rho}_t$, denoting the minimizer by $\boldsymbol{\rho}_t^*$ and $z_t^{(r+1)} = \max(f_{\text{lin}}(\boldsymbol{\rho}_t^*), z_t^{(r)})$, where $z_t^0 = -\infty$. Finally, $\boldsymbol{\rho}_t^{(r)}$ is updated to $\boldsymbol{\rho}_t^{(r+1)}$ as

$$\boldsymbol{\rho}_t^{(r+1)} = \boldsymbol{\rho}_t^{(r)} + \alpha(\boldsymbol{\rho}_t^* - \boldsymbol{\rho}_t^{(r)}), \qquad (13)$$

in which α is chosen as

$$\arg\min_{\alpha \in [0,1]} f(\boldsymbol{\rho}_t^{(r)} + \alpha(\boldsymbol{\rho}_t^* - \boldsymbol{\rho}_t^{(r)})). \qquad (14)$$

At every recursion, $f(\boldsymbol{\rho}_t^{(r)})$ is an upper bound on the optimized objective, while $z_t^{(r+1)}$ is a lower bound. Note that the proposed EKAR-BP algorithm is straightforward to use if the LDPC code was designed by PEG, or its variations [21], [22], but is not limited to such designs.

IV. Simulation Results

In this section, we present the simulation results of the proposed IDD scheme with the CKAR-BP and EKAR-BP algorithms for a 4×4 LDPC-coded MIMO system with PIC-MMSE detection. The LDPC code is a regular code designed by the PEG algorithm [20] whose block length N is 1000, the rate R is 0.5, the girth (g) is 6, and the degree distributions are $3(\upsilon(x) = x^4)$ and $5(\nu(x) = x^6)$ respectively. We consider uncorrelated Rayleigh flat fading channels and used 30 inner decoding iterations in this experiment. For the EKAR-BP decoder, $T = 20$ subgraphs have been generated, where check nodes are allowed to be re-visited, and 600 recursions were employed to obtain $\boldsymbol{\rho}$.

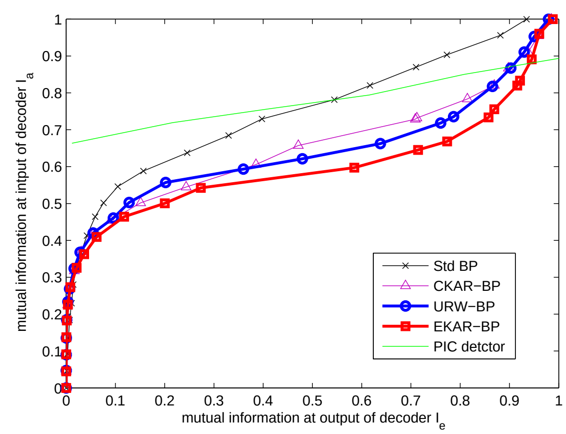

Fig. 2. EXIT charts of different decoders with a PIC detector. The proposed EKAR-BP decoder matches better with the PIC detector than other decoders. The EXIT chart of the PIC detector is obtained at $E_b/N_0 = 4$dB.

In comparison with the standard BP and URW-BP algorithms, we first draw an *extrinsic* information transfer (EXIT)

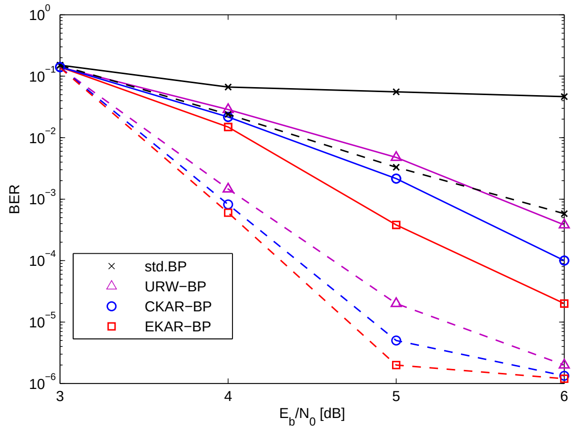

Fig. 3. Comparison of the standard BP, URW-BP, CKAR-BP, and EKAR-BP in terms of BER performances for a 4×4 system.

charts of different decoders with the SISO PIC detector in Fig. 2. Although the curve of the PIC-MMSE detector does not reach the top-right $(1, 1)$ point at the given SNR, it is obvious that the combination of PIC-MMSE detector and the proposed EKAR-BP decoder creates the widest detection and decoding tunnel. Additionally, only the tunnel between the PIC-MMSE detector and the standard BP decoder is closed at an early stage, which indicates that performance gain from the IDD process could be significantly diminished in this case. To verify the result of the EXIT chart, Fig. 3 depicts the performance in bit-error ratio (BER) of the MIMO system. We have used 30 inner decoding iterations and up to 3 outer detection and decoding iterations. The performance curves after 2 outer iterations are denoted by solid lines while the curves after 3 outer iterations are denoted by dashed lines. From Fig. 3, both CKAR-BP and EKAR-BP decoders outperform the standard BP and URW-BP decoder in the first detection and decoding iteration. In the third outer iteration, the proposed decoders are still able to generate relatively good performance when considering the low SNR range and the block length of code. Notice that there is an error floor effect at the BER of 10^6, which can be mitigated by using decision feedback techniques, [8], [11] and [12]. As mentioned in Section III, the key feature of the proposed KA-BP decoders lies in that no additional complexity is imposed in real-time decoding since the optimization of $\boldsymbol{\rho}$ is carried out offline. Moreover, by increasing the number of subgraphs T the EKAR-BP can accelerate the optimization process such that it can be employed for time-varying channels.

V. Conclusion

We have proposed an IDD scheme for MIMO systems with a conventional PIC-MMSE detector and two novel KA-BP decoders, which implement the reweighting strategy for decoding finite-length regular or irregular LDPC codes. The proposed CKAR-BP and EKAR-BP algorithms have different computational costs in the optimization phase, but neither of which requires extra complexity for online decoding. Furthermore, the EKAR-BP algorithm provides a trade-off between the number of expanded subgraphs and the convergence speed of the reweighting parameters. Numerical results show that the proposed IDD system is able to offer good performance while using a reduced number of inner and outer iterations.

References

[1] I. E. Telatar, "Capacity of Multi-Antenna Gaussian Channels", *Eur. Trans. Telecommun.*, vol. 10, no. 6, pp. 585-595, 1999.

[2] X. Wang and H. V. Poor, "Iterative (Turbo) soft interferece cancellation and decoding for coded CDMA," *IEEE Trans. Communications*, vol. 47, No. 7, pp. 1046-1061, 1999.

[3] B. Hochwald and S. T. Brink, "Achieving near-capacity on a multiple-antenna channel," *IEEE Transactions on Communications*, vol. 51, pp. 389-399, Mar. 2003.

[4] S. ten Brink, G. Kramer, A. Ashikhmin, "Design of low-density parity-check codes for modulation and detection," *IEEE Transactions on Communications*, vol. 52, no. 4, pp. 670- 678, April 2004.

[5] J. Hou, P. H. Siegel, L. B. Milstein, "Design of multi-input multi-output systems based on low-density Parity-check codes," *IEEE Transactions on Communications*, vol. 53, no. 4, pp. 601- 611, April 2005.

[6] H. Lee, B. Lee, and I. Lee, "Iterative detection and decoding with an improved V-BLAST for MIMO-OFDM Systems, *IEEE J. Sel. Areas Commun.*, vol. 24, pp. 504-513, Mar. 2006.

[7] Jianming Wu, Heung-No Lee, "Performance Analysis for LDPC-Coded Modulation in MIMO Multiple-Access Systems," *IEEE Transactions on Communications*, vol. 55, no. 7, pp. 1417-1426, July 2007.

[8] R.C. de Lamare, R. Sampaio-Neto, "Minimum Mean-Squared Error Iterative Successive Parallel Arbitrated Decision Feedback Detectors for DS-CDMA Systems", *IEEE Transactions on Communications*, vol. 56, no. 5, May 2008, pp. 778 - 789.

[9] J. W. Choi, A. C. Singer, J Lee, N. I. Cho, "improved linear soft-input soft-output detection via soft feedback successive interference cancellation, *IEEE Trans. Commun.*, vol. 58, no. 3, March 2010.

[10] R. C. de Lamare and R. Sampaio-Neto, "Adaptive reduced-rank equalization algorithms based on alternating optimization design techniques for MIMO systems," *IEEE Trans. Veh. Technol.*, vol. 60, no. 6, pp. 2482 2494, Jul. 2011.

[11] P. Li, R. C. de Lamare and R. Fa, "Multiple Feedback Successive Interference Cancellation Detection for Multiuser MIMO Systems," *IEEE Transactions on Wireless Communications*, vol. 10, no. 8, pp. 2434-2439, August 2011.

[12] Peng Li, R. C. de Lamare, "Adaptive Decision-Feedback Detection With Constellation Constraints for MIMO Systems," *IEEE Transactions on Vehicular Technology*, vol. 61, no. 2, pp. 853-859, Feb. 2012.

[13] R. G. Gallager, "Low-density parity check codes," *IRE Trans. Inf. Theory.*, vol. 39, no. 1, pp. 37-45, Jan. 1962.

[14] W. Ryan and S. Lin,"Channel Codes: Classical and Modern," Cambridge University Press, 1st edition, Oct. 30, 2009.

[15] F. Ding, T. Chen, "Gradient Based Iterative Algorithms for Solving a Class of Matrix Equations," *IEEE Transactions on Automatic Control*, vol. 50, no. 8, pp. 1216- 1221, Aug. 2005.

[16] M. J. Wainwright, T. S. Jaakkola, and A.S. Willsky, "A new class of upper bounds on the log partition function," *IEEE Trans. Information Theory*, vol. 51, no. 7, pp. 2313 - 2335, July 2005.

[17] H. Wymeersch, F. Penna and V. Savic, "Uniformly Reweighted Belief Propagation for Estimation and Detection in Wireless Networks," *IEEE Trans. Wireless Communications*, vol. PP, No. 99, pp. 1-9, Feb. 2012.

[18] J. Liu, R. C. de Lamare, "Low-Latency Reweighted Belief Propagation Decoding for LDPC Codes," *IEEE Communications Letters*, vol. 16, no. 10, pp. 1660-1663, October 2012.

[19] T. R. Halford, K. M. Chugg, "An algorithm for counting short cycles in bipartite graph," *IEEE Trans. on Infor. Theory*, vol. 52, no. 1, pp. 287-292, Jan. 2006.

[20] Y. Hu, E. Eleftheriou and D. M. Arnold, "Regular and irregular progressive edge-growth tanner graph," *IEEE Transaction on Information Theory*, vol. 51, no. 1, pp. 386-398, Jan. 2005.

[21] A. Uchoa, C. T. Healy, R. de Lamare and R. D. Souza, "Design of LDPC Codes Based on Progressive Edge Growth Techniques for Block Fading Channels," *IEEE Communications Letters*, vol. 15, no. 11, pp. 1221-1223, 2011.

[22] C. T. Healy, R. C. de Lamare, "Decoder-Optimised Progressive Edge Growth Algorithms for the Design of LDPC Codes with Low Error Floors", vol. 16, no. 6, pp. 889-892, 2012.