Shuffled Iterative Receiver for LDPC-Coded MIMO Systems

Peiyao Zhao[1], Chen Qian[1], Zhaocheng Wang[1], Linglong Dai[1] and Sheng Chen[2,3]

[1]Tsinghua National Laboratory for Information Science and Technology (TNList),
Department of Electronic Engineering, Tsinghua University, Beijing 100084, China

[2]Electronics and Computer Science, University of Southampton, Southampton SO17 1BJ, U.K.

[3]Faculty of Engineering, King Abdulaziz University, Jeddah 21589, Saudi Arabia

***Abstract*—In this paper, we consider the low density parity check (LDPC) coded multi-input multi-output (MIMO) system with iterative detection and decoding (IDD). Since the traditional frame-by-frame receiver scheme suffers from a huge decoding delay, we propose an efficient scheme with a shuffled structure between the demapper and decoder, which adopts group vertical shuffled belief propagation (BP) algorithm. The proposed shuffled iterative receiver converges faster and significantly reduces the delay introduced by the IDD process. Simulation results demonstrate that our proposed shuffled iterative receiver exhibits several tenths dB of signal-to-noise ratio gain in comparison to the existing schemes, while imposing a much lower average number of iterations for the IDD process.**

***Index Terms*—Iterative detection and decoding, MIMO, shuffled iterative receiver, LDPC code, shuffled BP algorithm**

I. Introduction

Many receiver schemes have been designed to approach the channel capacity of multi-input multi-output (MIMO) systems. In particular, the receivers that adopt an iterative detection and decoding (IDD) structure [1], [2] are capable of closely approximating the optimal joint detection and decoding in an iterative fashion and, therefore, achieving excellent performance while maintaining tractable complexity. An IDD receiver consists of a soft detector/demapper and a soft decoder. The demapper estimates the log likelihood ratios (LLRs) of the encoded bits, which serve as the input of the decoder. Then the decoder generates *a posteriori* LLRs and feeds back the *extrinsic* information to the demapper. This iterative process is repeated until the procedure converges or the preset maximum number of iterations is reached.

Low density parity check (LDPC) code is a class of linear block code with near Shannon limit performance. It has been widely considered as a forward error correction (FEC) code in the IDD schemes for MIMO systems [3]–[7]. In [3], the decoder exchanges the *extrinsic* information with the demapper frame by frame per l_c decoding-loop iterations. In the process, the check node messages are either reset to zero or not reset after each demapper-decoder iteration, which are referred to as the resetting and non-resetting algorithms, respectively. The non-resetting algorithm with $l_c = 1$ is the traditional frame-by-frame scheme commonly used in LDPC-coded MIMO systems. However, such LDPC-coded MIMO systems suffer from the drawbacks of high computational complexity and severe iteration delay.

Shuffled decoding is first proposed in the turbo-decoding field to improve the convergence speed. The scheme proposed in [8] extends the shuffled decoding to reduce the delay of demapper-decoder iteration for bit-interleaved coded modulation with iterative demapping (BICM-ID) in single-input single-output systems. But the number of demapping units required remain large, which equals to the parallel order of the decoder. This may lead to a prohibitive computational complexity for high-order modulation, and thus it is unsuitable for MIMO systems.

In this paper, we develop an efficient IDD scheme with a shuffled structure between the demapper and decoder for LDPC-coded MIMO systems. The proposed shuffled iterative receiver as usual consists of a soft demapper, a bit-wise interleaver and an LDPC decoder. However, our decoder adopts a semi-parallel structure in which the *extrinsic* information generated in each decoding cycle is fed back to the demapper as the *a priori* information immediately, instead of waiting for the decoding completion of the whole code frame. The bit-wise interleaver is carefully designed to guarantee that the bits fed back by the decoder in each cycle are mapped onto several intact symbol vectors. The number of demapping units required by our shuffled iterative receiver equals to the number of symbol vectors, which is much smaller than that of the scheme proposed in [8]. We also propose a partial feedback of decoded bits which offers a flexible performance-complexity tradeoff. Based on a well-designed schedule, our scheme enjoys a low iteration delay as well as a relatively low complexity. Simulation results show that the proposed shuffled iterative receiver exhibits several tenths dB gains in the signal-to-noise ratio (SNR) in comparison to the non-iterative scheme and resetting algorithm given in [3], while imposing a much lower average number of iterations.

II. Background

A. System Model

The LDPC-coded MIMO system with N_t transmit antennas and N_r receive antennas is considered, in which the interleaver and de-interleaver are denoted by Π and Π^{-1}, respectively.

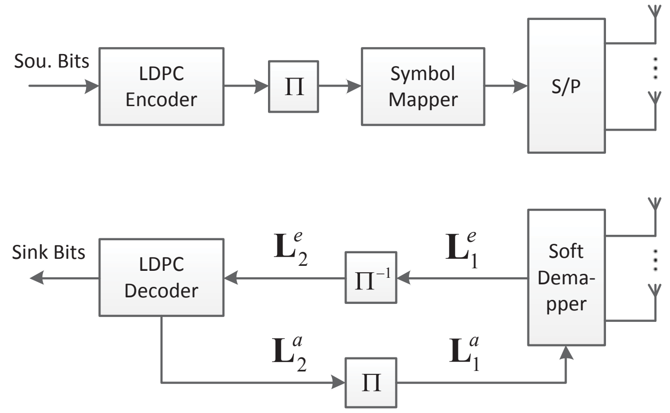

Fig. 1: LDPC-coded MIMO system with iterative detection and decoding.

As shown in Fig. 1, the source bits $\mathbf{u} = \left[u_1\ u_2 \cdots u_K\right]^{\mathrm{T}}$ are encoded by a rate-R_c LDPC code into $\mathbf{c} = \left[c_1\ c_2 \cdots c_N\right]^{\mathrm{T}}$, where $K = N \cdot R_c$. The coded bits after passing through the interleaver are grouped into vectors of length $K_b = m \cdot N_t$, and each bit vector is mapped onto a symbol vector $\mathbf{s} \in \mathbb{C}^{N_t \times 1}$ whose entries are chosen from a complex-valued constellation $\mathcal{A}$, where $|\mathcal{A}| = 2^m$ and m is the order of the constellation. The received signal $\mathbf{y}$ is given by

$$\mathbf{y} = \mathbf{H}\,\mathbf{s} + \mathbf{n}, \tag{1}$$

where $\mathbf{H} \in \mathbb{C}^{N_r \times N_t}$ is the MIMO channel matrix and $\mathbf{n} \in \mathbb{C}^{N_r \times 1}$ denotes a complex-valued additive white Gaussian noise (AWGN) vector with covariance matrix $\sigma^2 \mathbf{I}_{N_r}$. We assume a quasi-static Rayleigh flat fading environment, and the entries of $\mathbf{H}$ are independent and identically distributed (i.i.d.) complex-valued Gaussian variables with zero mean and a variance 0.5 per dimension. We further assume that the receiver has perfect knowledge of the channel matrix $\mathbf{H}$.

The receiver performs IDD as illustrated in Fig. 1. For each demapper-decoder iteration, the demapper calculates the *extrinsic* information $\mathbf{L}_1^e$ based on the channel observation $\mathbf{y}$ and the *a priori* information $\mathbf{L}_1^a$ provided by the decoder. Then $\mathbf{L}_1^e$ is forwarded to the decoder as the *a priori* information $\mathbf{L}_2^a$ after de-interleaving, based on which the decoder generates the *a posteriori* LLRs. The *extrinsic* information $\mathbf{L}_2^e$ of the decoder is in turn fed back to the demapper as the *a priori* information $\mathbf{L}_1^a$ after re-interleaving for next iteration. The iterative operations are repeated until all the checks are satisfied or the pre-determined maximum iteration number is reached.

B. Soft-Input Soft-Output Demapper

Both the optimal maximum likelihood (ML) demapper [1] and the suboptimal K-BEST sphere decoder demapper [9] are considered. The demapper computes the *extrinsic* information for each coded block of bits based on the received vector $\mathbf{y}$ and the *a priori* information $\mathbf{L}_1^a$. Let the *a priori* information $\mathbf{L}_1^a$ and the *extrinsic* information $\mathbf{L}_1^e$ associated with each coded block of bits b_i, $1 \leq i \leq K_b$, be denoted as $\left\{L_{1,1}^a, L_{1,2}^a, \cdots, L_{1,K_b}^a\right\}$ and $\left\{L_{1,1}^e, L_{1,2}^e, \cdots, L_{1,K_b}^e\right\}$, respectively. The *extrinsic* information $L_{1,i}^e$ of the i-th bit b_i, where $1 \leq i \leq K_b$, associated with the transmit vector $\mathbf{s}$ is given by [1]

$$\begin{aligned} L_{1,i}^e = & \log \sum_{\mathbf{s} \in \mathbb{B}_{i,0}} \exp\left(-\frac{||\mathbf{y} - \mathbf{H}\,\mathbf{s}||^2}{\sigma^2} + \sum_{j \neq i, b_j = 0} L_{1,j}^a\right) \\ & - \log \sum_{\mathbf{s} \in \mathbb{B}_{i,1}} \exp\left(-\frac{||\mathbf{y} - \mathbf{H}\,\mathbf{s}||^2}{\sigma^2} + \sum_{j \neq i, b_j = 0} L_{1,j}^a\right), \end{aligned} \tag{2}$$

where $\mathbb{B}_{i,0}$ and $\mathbb{B}_{i,1}$ denote the sets of the candidate symbol vectors with $b_i = 0$ and $b_i = 1$, respectively. All the possible transmit vectors are considered by the ML demapper, leading to a computational complexity that increases exponentially with $K_b = m \cdot N_t$. By contrast, for the K-BEST demapper, a small set of the candidate vectors is generated by a breath-first tree search keeping only the best K candidate at each level, and consequently the complexity is reduced. It should be noted that if either $\mathbb{B}_{i,0}$ or $\mathbb{B}_{i,1}$ is null, no information is obtained regarding one of the two hypothesises of this bit. In such a case, then the output LLR is clipped to a constant value, denoted by $\pm l_{clip}$, respectively.

For both the ML demapper and the K-BEST demapper, the candidate transmit vectors and the corresponding Euclidean distances $||\mathbf{y} - \mathbf{H}\,\mathbf{s}||$ are stored for the iterative operation.

C. LDPC Decoder

A group vertical shuffled belief propagation (BP) algorithm [10] is adopted at the decoder to speed up the convergence of decoding. We group all the bit nodes into G layers uniformly and perform the vertical shuffled BP algorithm layers by layers in each iteration. The decoding process is summarized in

Algorithm 1 Group Vertical Shuffled BP Algorithm

For iteration t ($t = 1, 2, \cdots, t_{max}$) and layer g ($g = 1, 2, \cdots, G$), perform the following operations on each bit node b that belongs to layer g.

Horizontal process

$$R_{l,b}^{(g,t)} = 2 \tanh^{-1}\left(\frac{M_l^{(g-1,t)}}{\tanh\left(Q_{l,b}^{(t-1)}/2\right)}\right), \text{ where } l \in \mathcal{M}(b) \tag{3}$$

Vertical process

$$Q_b^{(t)} = F_b^{(t-1)} + \sum_{l \in \mathcal{M}(b)} R_{l,b}^{(g,t)} \tag{4}$$

$$Q_{l,b}^{(t)} = Q_b^{(t)} - R_{l,b}^{(g,t)} \tag{5}$$

Updating process

$$M_l^{(g,t)} = \prod_{\substack{b' \in \mathcal{N}(l) \\ G(b') < g}} \tanh\left(\frac{Q_{l,b'}^{(t)}}{2}\right) \prod_{\substack{b' \in \mathcal{N}(l) \\ G(b') \geq g}} \tanh\left(\frac{Q_{l,b'}^{(t-1)}}{2}\right) \tag{6}$$

Algorithm 1, where $R_{l,b}$ denotes the message passing from check node l to bit node b and $Q_{l,b}$ as the reverse operation, while F_b and Q_b are the *a priori* and *a posteriori* LLRs of bit b, respectively. The superscript pair (g, t) of a symbol represent the corresponding value at the g-th layer and t-th iteration. Since the values of $Q_{l,b}$,Q_b and F_b do not change with layer number g, we omit the superscript g on them. In Algorithm 1, $\mathcal{M}(b)$ denotes the set of the check nodes connected to bit node b, and $\mathcal{N}(l)$ denotes the set of bit nodes that participate in check l, while M_l is an intermediate variable defined in Eq. (6), where $G(b')$ denotes the layer number of bit node b'. Notice that in Eq. (6), $G(b') < g$ means that the bit node b' has been decoded in the former layers, and thus we use the updated value $Q_{l,b'}^{(t)}$. The decoding process of a layer is referred to as a decoding *cycle*. In each decoding cycle, the decoder generates the *a posteriori* LLRs of $P = N/G$ bits. The initialization, stopping criterion test and output steps remain the same as those of the standard BP algorithm [11].

The group shuffled decoding is suitable for quasi-cyclic LDPC (QC-LDPC) code whose check matrix is comprised of circulant matrices and null matrices of the same size $q \times q$. We can simply set $G = q$, and the g-th layer contains the g-th bit of each sub-matrix, where $g = 1, 2, \cdots, G$. In this paper, we use QC-LDPC code as an example, and we point out that other kinds of LDPC codes can also be supported.

D. Iterative Operation Between Demapper and Decoder

The traditional MIMO IDD receiver performs demapper-decoder iteration in a frame-by-frame schedule, which means that the *extrinsic* information generated by the decoder can only be fed back to the demapper after the entire code frame has been decoded. In such a frame-by-frame schedule, the decoder and demapper work in turn, and each waits the other to complete its operations on an entire code frame, which leads to a huge iteration delay. The long delay of traditional frame-by-frame schemes severely limits the effective throughput of the system. In order to reduce the IDD delay, a large number of demapping units are required which however increases the complexity considerably.

For example, the message passing algorithm proposed in [8] extends the idea of shuffled decoding to exchange information between the demapper and decoder efficiently which utilizes the parallelism of the LDPC decoder and a partial update strategy of the demapper. In each sub-iteration, the LLRs of the bits involved are calculated by the demapper employing the existing *a priori* information, and then the decoder generates the *extrinsic* information of these bits which are fed back to the demapper immediately. This scheme reduces the delay introduced by demapper-decoder iteration considerably, but the number of demapping units required, which equals to the parallel order of the LDPC decoder, is large. Therefore, its complexity is relatively high, especially for systems with high-order modulation. Consequently, for a large MIMO system with high-order modulation, the computational complexity of the shuffled decoding scheme proposed in [8] may become prohibitively high.

III. Efficient Shuffled Iterative Receiver

As discussed in the previous section, a critical problem of the conventional frame-by-frame receiver scheme is the severe iteration delay induced. The shuffled receiver scheme of [8] may effectively reduce this IDD delay at the cost of high complexity. We propose an efficient shuffled iterative receiver which enjoys a low IDD delay and converges faster, while only imposing a relatively low complexity.

A. The Proposed Shuffled Iterative Receiver

Consider the group vertical shuffled BP algorithm adopted by the decoder. In the g-th decoding cycle, the *extrinsic* information $\mathbf{L}_2^e$ of P bits, denoted as $\mathbf{L}_{2,g}^e$, are generated by the decoder employing the *a priori* information $\mathbf{L}_2^a$ of these bits, denoted as $\mathbf{L}_{2,g}^a$. Note that $\mathbf{L}_{2,g}^a$ will only be used in the g-th decoding cycle of the next iteration, and thus the updating of $\mathbf{L}_{2,g}^a$ by the demapper does not interfere with the decoding operation of other layers, which means the demapper and decoder can work in parallel with a well designed schedule. In our proposed scheme, the *extrinsic* information $\mathbf{L}_{2,g}^e$ are fed back to the demapper immediately, after they are generated by the decoder. Then the demapper updates the *a priori* information $\mathbf{L}_{2,g}^a$, which will be forwarded to the decoder for next iteration. At the same time, the decoder moves to the next decoding cycle without waiting for the completion of demapping operation. The demapper and decoder form a pipeline structure which reduces the iteration delay significantly, compared with the traditional frame-by-frame scheme.

We also propose a partial feedback strategy, which only feeds back the *extrinsic* information of P_f bits in each decoding cycle, where $P_f \le P$. With a smaller number of bits participating in the feedback, the computational complexity of demapping is reduced. Besides, only the candidate transmit vectors and the corresponding Euclidean distances associated with the bits that participate in the feedback need to be stored, which leads to a reduction of RAM resources. Hence the partial feedback strategy offers a flexible trade-off between the performance and complexity. In particular, $P_f = 0$ indicates that no *extrinsic* information is exchanged between the decoder and demapper, which is equivalent to the non-iterative scheme, while $P_f = P$ means that all the *extrinsic* information are fed back and, therefore, the hardware complexity required is the highest and the BER performance attainable is the best.

The interleaver and de-interleaver are carefully designed to guarantee that the feedback bits are mapped onto several intact symbol vectors. Thus we have $P_f/K_b \in \mathbb{N}$. This minimizes the number of symbol vectors related to these P_f bits, and consequently it reduces the number of demapping units required, which equals to P_f/K_b. By contrast, the scheme proposed in [8] requires P demapping units, which is much larger than P_f/K_b. Thus our proposed shuffled iterative receiver enjoys a much lower complexity. The interleaving process is actually reading/writing the *extrinsic* information at appropriate address and, therefore, it can simply be realized as a look-up table (LUT) that memorises the reading/writing addresses at each decoding cycle.

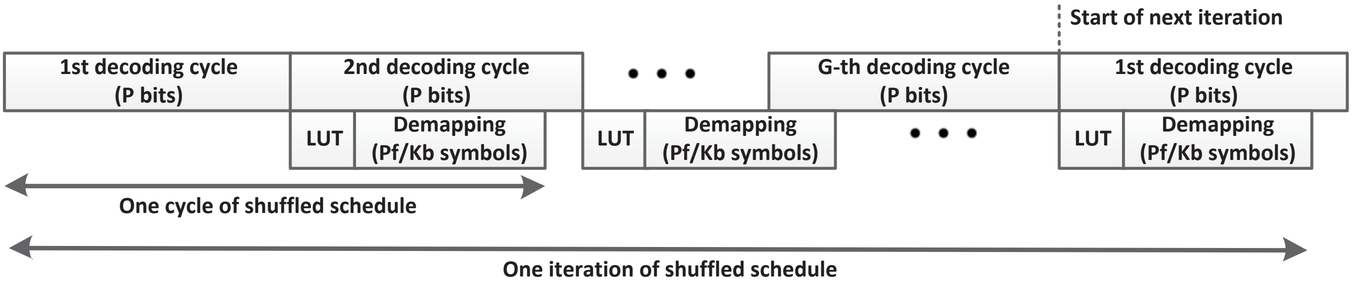
Start of next iteration
1st decoding cycle (P bits)
2nd decoding cycle (P bits)
G-th decoding cycle (P bits)
1st decoding cycle (P bits)
LUT
Demapping (Pf/Kb symbols)
One cycle of shuffled schedule
One iteration of shuffled schedule

Fig. 2: Schedule of the proposed shuffled iterative receiver.

The proposed shuffled iterative receiver is given in Algorithm 2, where it is seen that in each decoding cycle g, P bits are decoded, while for the P_f bits among these P bits, which are to be fed back, their *extrinsic* information are calculated by the P_f/K_b demapping units and thereafter the *a priori* information of these P_f bits are updated. Note that for MIMO systems with a large number of antennas and/or high-order modulation, K_b may become larger than P, which indicates that the bits decoded in one cycle cannot be mapped onto an intact symbol. Therefore some modifications are made on the schedule for such systems. We perform a decoder-demapper iteration per l_c decoding cycles and the decoded $l_c \cdot P_f$ bits are mapped onto several intact symbol vectors. Thus the number of demapping units required becomes $l_c \cdot P_f/K_b$. The receiver still enjoys a pipeline structure with low IDD delay.

Algorithm 2 Algorithm of Shuffled Iterative Receiver

For iteration t ($t = 1, 2, \cdots, t_{max}$) and cycle g ($g = 1, 2, \cdots, G$), denote n ($n = 1, 2, \cdots, P$) as the index of the bits processed in this cycle, $\tilde{n}$ ($\tilde{n} = 1, 2, \cdots, P_f$) as the index of the bits to be fed back to the demapper, and $\hat{n}$ ($\hat{n} = \Pi(1), \Pi(2), \cdots, \Pi(P_f)$) as the index of the interleaved feedback bits associated with index $\tilde{n}$. The interleaved bits are mapped onto symbol vector k ($k = 1, 2, \cdots, P_f/K_b$).

Decoding Process

1. Calculate the *a posteriori* LLRs of all the bits $Q_n^{(t)}$ using Algorithm 1.

2. Calculate the *extrinsic* information of the bits to be fed back as

$$L_2^e(\tilde{n}) = Q_{\tilde{n}}^{(t)} - F_{\tilde{n}}^{(0)} \tag{7}$$

Interleaving Process

$$L_1^a(\hat{n}) = L_2^e(\tilde{n}) \tag{8}$$

Demapping Process

Calculate the *extrinsic* information $L_1^e(\hat{n})$ by demapping symbol vector k using Eq. (2).

De-interleaving Process

Update the *a priori* information $F_{\tilde{n}}^{(t)}$ according to

$$F_{\tilde{n}}^{(t)} = L_1^e(\hat{n}) \tag{9}$$

The *a priori* information of the bits that do not participate in the iterative process remain unchanged.

B. *Analysis of The Proposed Scheme*

The proposed shuffled iterative receiver has several advantages. Firstly, the delay induced by the IDD process is greatly reduced compared with the traditional frame-by-frame scheme. Secondly, the number of demapping units required is much less than that of the existing shuffled receiver given in [8], leading to a low complexity. Furthermore, the proposed partial feedback strategy offers a flexible trade-off between the performance and complexity.

Fig. 2 illustrates the schedule of our proposed shuffled iterative receiver, where it can be observed that this shuffled iterative receiver employs a parallel schedule, namely, the decoder and demapper form a pipeline structure and they work simultaneously. As long as the sum of the clock cycles required for the LUT and demapping operations is guaranteed to be no larger than that of a decoding cycle, the decoder will never be idle to wait for the completion of demapping operation.

Let us take the QC-LDPC code in IEEE 802.11n with the code length $N = 1944$, the code rate $R_c = 2/3$ and the sub-matrix size of $q = 81$ as an example. The decoding process consists $G = 81$ cycles and in each cycle $P = 24$ bits of a layer are decoded. For simplicity, consider that the *extrinsic* information of all the $P_f = P$ bits are fed back. Further assume that a decoding cycle occupies T_c clock cycles, a demapping unit which handles a symbol vector needs T_d clock cycles, and the LUT operation needs δ clock cycles. As can be inferred from Fig. 2, for the proposed shuffled iterative receiver, a total of $81T_c + T_d + \delta$ clock cycles are required for an iteration. By contrast, for the traditional frame-by-frame scheme, if we use the same number of demapping units as the shuffled one, a decoder-demapper iteration requires $81T_c + 81T_d + \Delta$ clock cycles, where Δ is the delay of the interleaver which is typically very long. It can be seen that our shuffled iterative receiver significantly reduces the delay induced by the IDD process, compared with the traditional frame-by-frame scheme. Even compared with the non-iterative scheme, for which $81T_c$ clock cycles are needed for one iteration, our proposed shuffled scheme is competitive in terms of process delay.

IV. Simulation Results

We now present the simulation results to compare the proposed shuffled iterative receiver with the non-iterative scheme and the resetting algorithm given in [3]. The QC-LDPC code

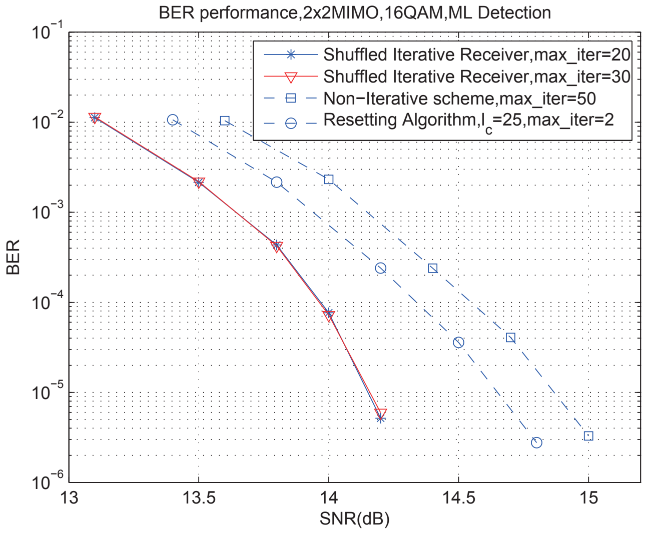

Fig. 3: BER performance comparison of the proposed shuffled iterative receiver, the non-iteration scheme and the resetting algorithm over the 2×2 MIMO channel with 16-QAM modulation and ML detection.

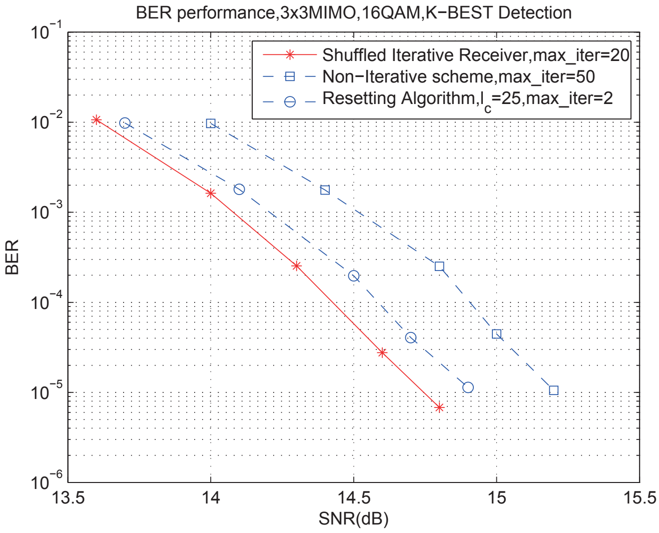

Fig. 5: BER performance comparison of the proposed shuffled iterative receiver, the non-iteration scheme and the resetting algorithm over the 3×3 MIMO channel with 16-QAM modulation and K-BEST detection.

in IEEE 802.11n with code length 1944 and code rate $2/3$ was employed. A quasi-static Rayleigh flat fading MIMO channel was assumed. And the SNR was defined as SNR $= \frac{E_s}{\sigma^2}$, where E_s denoted the average symbol energy.

For the $(N_r = 2) \times (N_t = 2)$ MIMO system with 16-QAM modulation and the ML detection, Figs. 3 and 4 compare the bit error rate (BER) performance and the average iteration numbers of the three receivers, respectively. For the non-iterative scheme, the maximum iteration number was set to 50. For the resetting algorithm, the decoder and demapper exchanged the *extrinsic* information once per $l_c = 25$ decoding-loop iterations, and the maximum iteration number of the decoder-demapper loop was set to 2. For our proposed scheme, the *extrinsic* information of $P_f = P = 24$ bits were fed back in each cycle, while the maximum iteration numbers of 20 and 30 were considered. For each scenario, the iterative process was repeated until the LDPC decoder converged or the preset maximum iteration number was reached.

It can be observed from Fig. 3 that our proposed shuffled iterative receiver provides approximately $0.7\,\text{dB}$ and $0.5\,\text{dB}$ gains in the SNR over the non-iterative scheme and the resetting algorithm, respectively, at the BER level of 10^{-5}. Furthermore, the average iteration number of our shuffled iterative algorithm is much less than those of the other two schemes at the same BER level, as can be seen from Fig. 4. Additionally, we also notice that the performance gain of our

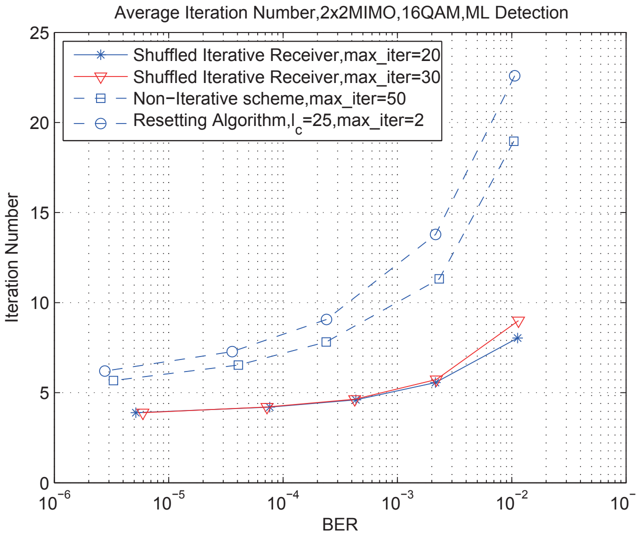

Fig. 4: Average iteration number comparison of the proposed shuffled iterative receiver, the non-iteration scheme and the resetting algorithm over the 2×2 MIMO channel with 16-QAM modulation and ML detection.

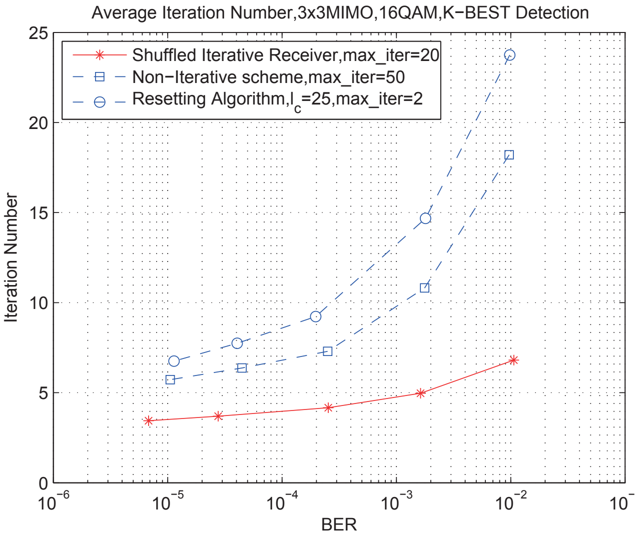

Fig. 6: Average iteration number comparison of the proposed shuffled iterative receiver, the non-iteration scheme and the resetting algorithm over the 3×3 MIMO channel with 16-QAM modulation and K-BEST detection.

proposed shuffled iterative receiver attained by increasing its maximum iteration number from 20 to 30 is limited. This demonstrates that our shuffled iterative receiver is capable of obtaining a good performance even with a relatively small maximum iteration number.

Next we present the simulation results for the $(N_r = 3) \times (N_t = 3)$ MIMO system with 16-QAM modulation and the K-BEST detection where K was set as 64. For our shuffled iterative receiver, only the maximum iteration number of 20 was considered, while in each cycle, all the decoded $P_f = P = 24$ bits were fed back. The parameters of the other two schemes remained the same as the previous example. As can be seen from Fig. 5, the proposed shuffled iterative receiver exhibits approximately $0.5\,$dB and $0.2\,$dB gains in the SNR at the BER level of 10^{-5} over the non-iterative scheme and the resetting algorithm, respectively. Due to the suboptimal demapping algorithm adopted, the gains are not as large as in the case of adopting the ML demapper, but they are still substantial. Fig. 6 shows that the average iteration number is also greatly reduced by our shuffled iterative receiver, compared with the other two schemes.

Our simulation investigation therefore shows that the proposed shuffled iterative receiver attains several tenths dB gains in the SNR in comparison to the widely used non-iterative scheme and resetting algorithm, as well as imposes a smaller number of iterations at a give BER level, compared with the existing schemes. Furthermore, as demonstrated in the previous section, our shuffled iterative receiver exhibits a much lower IDD delay, compared with the traditional frame-by-frame scheme. Therefore, our proposed scheme offers a low-complexity and low-delay design to achieve a high MIMO system throughput.

V. Conclusions

In this contribution, we have proposed a shuffled iterative receiver for LDPC-coded MIMO systems. In our shuffled iterative receiver, the decoder adopts the vertical group shuffled BP algorithm, and the *extrinsic* information of the decoded bits generated in each cycle are fed back to the demapper immediately, rather than waiting for the completion of decoding the entire code frame. The decoder and demapper form a pipeline structure which leads to a significant reduction in the IDD delay. A partial feedback strategy has also been suggested to provide a flexible performance and complexity trade-off. Simulation results have demonstrated that the proposed shuffled iterative receiver outperforms the existing non-iterative scheme and resetting algorithm in terms of achievable BER performance, while imposing a smaller average number of iterations. Our work thus has shown that our proposed scheme offers a low-complexity and low-delay IDD design for high-throughput LDPC-coded MIMO systems.

Acknowledgment

This work was supported by National Nature Science Foundation of China (Grant No. 61271266), National High Technology Research and Development Program of China (Grant No. 2014AA01A704), Beijing Natural Science Foundation (Grant No. 4142027), Shenzhen Visible Light Communication System Key Laboratory (ZDSYS20140512114229398) and Shenzhen Peacock Plan (No. 1108170036003286).

References

[1] 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.

[2] Y. Liu, M. P. Fitz, and O. Y. Takeshita, “Full rate space-time turbo codes,” *IEEE J. Select. Areas Commun*, vol. 19, no. 5, pp. 969–980, May 2001.

[3] J. Hou, P. H. Siegel, and L. B. Milstein, “Design of multi-input multi-output systems based on low-density parity-check codes,” *IEEE Trans. Commun.*, vol. 53, no. 4, pp. 601–611, Apr. 2005.

[4] S. ten Brink, G. Kramer, and A. Ashikhmin, “Design of low-density parity-check codes for modulation and detection,” *IEEE Trans. Commun.*, vol. 52, no. 4, pp. 670–678, Apr. 2004.

[5] J. Nam, S. R. Kim, J. Ha, and J. Y. Ahn, “A new design of iterative detection and decoding with soft interference cancellation,” in *Proc. VTC 2008-Fall* (Calgary, BC, Canada), Sept. 21-24, 2008, pp. 1–6.

[6] J. Liu, P. Li, and R. C. de Lamare, “Iterative detection and decoding for MIMO systems with knowledge-aided belief propagation algorithms,” in *Proc. Asilomar Conf. Signals, Syst., Comp.* (Pacific Grove, CA), Nov. 4-7, 2012, pp. 1250–1254.

[7] B. Lu, G. Yue, and X. Wang, “Performance analysis and design optimization of LDPC-coded MIMO OFDM systems,” *IEEE Trans. Signal Processing*, vol. 52, no. 2, pp. 348–361, Feb. 2004.

[8] M. Li, C. A. Nour, C. Jego, J. Yang, and C. Douillard, “A shuffled iterative bit-interleaved coded modulation receiver for the DVB-T2 standard: Design, implementation and FPGA prototyping,” in *Proc. 2011 IEEE Workshop Signal Processing Systems* (Beirut, Lebanon), Oct. 4-7, 2011, pp. 55–60.

[9] Z. Guo and P. Nilsson, “Algorithm and implementation of the k-best sphere decoding for mimo detection,” *IEEE J. Select. Areas Commun.*, vol. 24, no. 3, pp. 491–503, Mar. 2006.

[10] J. Zhang and M. P. C. Fossorier, “Shuffled iterative decoding,” *IEEE Trans. Commun.*, vol. 53, no. 2, pp. 209–213, Feb. 2005.

[11] D. J. C. MacKay, “Good error-correcting codes based on very sparse matrices,” *IEEE Trans. Inf. Theory*, vol. 45, no. 2, pp. 399–431, Mar. 1999.