Graph-Based Detection and LDPC Decoding over 2D Intersymbol Interference Channels

Yingying Li, Zhiliang Qin, Lianghui Zou, Yu Qin, and Qidong Lu
Weihai Beiyang Electric Group Co. Ltd., Weihai, Shandong, China
Email: {liyingying; qinzhiliang; zoulianghui; qinyu; luqidong}@beiyang.com

Abstract—In this paper, we propose a fully graph-based iterative detection and decoding scheme for Low-Density Parity-Check (LDPC) coded generalized two-dimensional (2D) intersymbol interference (ISI) channels. The 2D detector consists of a downtrack detector based on the symbol-level sum-product algorithm (SPA) and a bit-level SPA-based crosstrack detector. A LDPC decoder based on simplified check node operations is also proposed to provide soft information for the 2D channel detector. Numerical results show that the proposed receiver achieves better performance as compared with the trellis-based BCJR detector over 2×2 2D channels while at a significantly lower computational complexity.

Index Terms—LDPC decoding, symbol-level, sum-product algorithm, two-dimensional partial-response channels

I. Introduction

Detection over Two-Dimensional (2D) intersymbol interference (ISI) channels arises from the development of page-oriented memories [1] and next-generation ultra-high-density data storage systems, such as bit patterned media recording (BPMR) [2], Heat-Assisted Magnetic Recording (HAMR) [3], etc. For these systems, interference arising from adjacent tracks becomes non-negligible, which gives rise to a 2D intersymbol interference (ISI) model when considered together with the downtrack interference. The 2D ISI model is found in wireless communication applications, for instance, multiple-input multiple-output (MIMO) systems [4] and filter bank multi-carrier (FBMC) systems [5]. Moreover, it can be used to characterize the inter-cell interference (ICI) [6] that exists in solid-state non-volatile memories (NVM).

In [7], it is shown that the maximum-likelihood sequence detection (MLSD) for 2D ISI channels is a non-polynomial (NP)-hard problem. In [8], a maximum *a posterior* probability (MAP) detector was proposed to process three adjacent tracks to detect bits in the middle track. In [9], an iterative detection algorithm was proposed based on a *specific* assumption that the 2D channel matrix can be written as the product of two one-dimensional (1D) vectors. In [10], a graph-based detection scheme was proposed for a rate-1/2 binary Low-Density Parity-Check (LDPC) coded *generalized* 2D channel, which, however, suffers from serious performance degradation when high-rate LDPC codes are used due to the existence of numerous short cycles in the bit-level 2D channel graph. In [11], an iterative row/column 2D detector was developed based on the trellis-based BCJR algorithm. It is shown that the BCJR-based 2D detector performs better than other schemes including the row-column soft-decision-feedback algorithm (RCSDFA) [12] and the Markov chain Monte Carlo (MCMC)-based algorithm [13] when a rate-0.89 LDPC code is transmitted over 2D channels.

Of particular interest to data storage systems are high-rate LDPC codes with low-encoding complexities, such as Quasi-Cyclic (QC) codes whose encoders can be implemented based on linear feedback shift registers [14], or extended irregular accumulate (eIRA) codes [15] with encoding directly based on the parity-check matrix. The decoding complexity of high-rate LDPC codes, however, still remains an issue as they typically have a much larger check-node degree d_c as compared with low-rate codes. When decoded with the Sum-Product Algorithm (SPA) in the Log-Likelihood Ratio (LLR) domain [16], the conventional check-node operation based on forward/back recursions [17], [18] has a complexity growing linearly with d_c, which is computationally intensive for large values of d_c.

In this paper, we first propose a fully graph-based detector and decoder for LDPC coded generalized 2D channels. The 2D detector consists of a symbol-level SPA-based downtrack detector and a bit-level SPA-based crosstrack detector. The channel graph can be combined with the LDPC code graph to form an overall graph, thus making it possible to propagate soft messages through these three components in a flooding schedule. To reduce the decoding complexity of high-rate LDPC codes, we thereby restrict the check-node operation to a subset corresponding to the Q smallest reliabilities among all d_c bit-to-check messages, $2 \le Q \le d_c$. An attenuation factor is introduced to compensate for the over-estimation of LLR reliabilities. Simulation results show that with properly adjusted parameters, the proposed receiver achieves a noticeable performance gain over the BCJR-based scheme [11] for a 2×2 2D channel; while the LDPC

Manuscript received August 15, 2020; revised February 2, 2021.
Corresponding author email: qinzhiliang@beiyang.com
doi:10.12720/jcm.16.3.91-98

decoding complexity becomes comparable to that of the Min-Sum algorithm.

II. System Model

The system model of a binary LDPC coded 2D ISI channel is shown in Fig. 1.

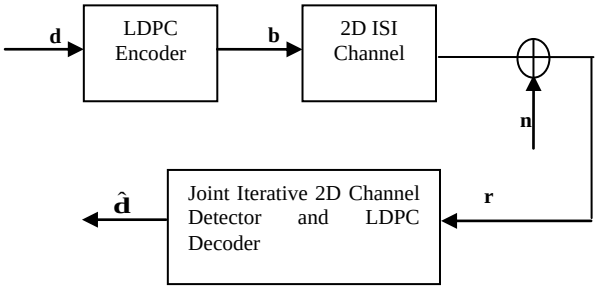

Fig. 1. System model of LDPC coded 2D ISI channels

The data sequence of K bits is passed to a high-rate LDPC encoder to generate a block of N code bits. The code bits are mapped to bipolar modulated bits **b** and scanned into an $l \times l$ matrix, where l is the square root of the LDPC codeword length. The channel output is an $l \times l$ matrix **R** with elements given by [9].

$$r(i,\, j) = \sum_{k_1=0}^{m} \sum_{k_2=0}^{m} h_{k_1,k_2} b(i-k_1,\, j-k_2) + n_{i,j} \tag{1}$$

where the 2D ISI is generated based on the 2D convolution of the channel response matrix **h** and the transmitted bit matrix **b**, $n_{i,j}$ denotes additive white Gaussian noise (AWGN) with zero mean and variance $\sigma^2=N_o/2$, N_o is the noise one-side power spectral density level, and m+1 denotes the number of elements over which the ISI extends in each dimension. The channel response matrix **h** in our simulations is a 2×2 matrix given by,

$$\mathbf{h} = \begin{bmatrix} 1 & 0.7 \\ 0.5 & 0.2 \end{bmatrix} \tag{2}$$

where the upper-left entry 1 is the coefficient corresponding to the currently transmitted bit. The user signal-to-noise ratio (SNR) is defined as

$$\mathrm{SNR} = 10 \log_{10} \left(\|\mathbf{h}\|^2 E_b / 2R\sigma^2 \right) \text{ dB} \tag{3}$$

where $\|\mathbf{h}\|$ is the 2-norm of the 2D channel matrix and E_b is the energy per data bit. Prior to transmission, a width-m guard band of −1 bits is added around the bit matrix for the purpose of isolating sectors and providing termination for detection algorithms.

III. Graph-based Detection and LDPC Decoding

The proposed receiver consists of a graph-based 2D channel detector and a low-complexity LDPC decoder, where the former is composed of two components, i.e., a symbol-level SPA-based downtrack detector and a bit-level SPA-based crosstrack detector. In the proposed scheme, symbol-level soft information is exchanged between the downtrack and the crosstrack detector, while the soft information of binary LDPC code bits is exchanged between the crosstrack detector and the LDPC decoder at each iteration.

A. Symbol-Level SPA-based Downtrack Detector

Let us define an (m+1)-tuple binary vector in the crosstrack direction as a downtrack transmitted symbol $x_{i,j}=[b_{i,j},b_{i-1,j},\ldots,b_{i-m,j}]$. The 2D channel can be viewed as a symbol-input 1D downtrack channel with memory length equal to m, which can be represented by a symbol-level trellis with p^m states per trellis stage and p branches departing from each state, where the downtrack symbol size is $p=2^{(m+1)}$.

For the ith track, i=0,…,l−1, an alternative representation of the downtrack channel can be formulated based on the concept of the factor graph [19], [20]. In Fig. 2(a), the downtrack channel graph for a 2×2 2D matrix is presented, where circles are used to denote symbol nodes corresponding to the ith row of transmitted symbols and squares denote channel check nodes describing ISI in the downtrack direction. Note that the guard band is also represented in the downtrack channel graph, which corresponds to m degree-1 symbol nodes. Hence, the number of symbol nodes and check nodes in the downtrack graph per row is (l+2m) and (l+m), respectively; while the channel check-node degree is m+1. Contrary to the LDPC code graph, the channel observation **R** is directly input to check nodes instead of to symbol nodes; while the *a priori* information $\mathbf{L}_{ct\to dt}$ of symbols **x** fed back from the crosstrack detector is input to symbol nodes, where the subscript *ct* and *dt* denotes the crosstrack and downtrack detector, respectively.

From a local perspective, the symbol-level SPA produces the *a posteriori* probability (APP) LLR of symbols **x** connected to a check node c_k, k=0,…,(l+m)−1, as shown in Fig. 2(b),

$$\Lambda^{dt}\left(x_z^{(c_k)} = x \middle| R_i^{(c_k)}\right) = \log \frac{P\left(x_z^{(c_k)} = x \middle| R_i^{(c_k)}, \boldsymbol{\lambda}_z^{(c_k)}\right)}{P\left(x_z^{(c_k)} = 0 \middle| R_i^{(c_k)}, \boldsymbol{\lambda}_z^{(c_k)}\right)} \tag{4}$$

where the superscript (c_k) denotes variables associated with c_k, the superscript *dt* denotes the soft information produced by the downtrack detector, the subscript z denotes the zth symbol connected to the check node c_k, z=0,…,m, the scalar $R_i^{(c_k)}$ denotes the channel output in the ith row corresponding to c_k, and $\boldsymbol{\lambda}_z^{(c_k)}$ denotes the *a priori* information provided by symbol nodes that participate in check c_k. After some manipulations, we have derived the following expression,

$$\Lambda^{dt}\left(x_z^{(c_k)} = x \middle| R_i^{(c_k)}\right) = \log \frac{\sum_{\mathbf{x}^{(c_k)} \in \{0,1,\cdots,2^{(m+1)}-1\}^{(m+1)}:\, x_z^{(c_k)}=x} \exp(\Omega)}{\sum_{\mathbf{x}^{(c_k)} \in \{0,1,\cdots,2^{(m+1)}-1\}^{(m+1)}:\, x_z^{(c_k)}=0} \exp(\Omega)} \tag{5}$$

where the metric in the downtrack direction is given by,

$$\Omega = -\frac{1}{2\sigma^2}\exp\left(R_i^{(c_k)} - \sum_{z=0}^{m} x_z^{(c_k)}\mathbf{h}_z\right)^2 + \sum_{z=0}^{m}\lambda_z^{(c_k)}\left(x_z^{(c_k)} = c\right) \quad (6)$$

, $\mathbf{h}_z$ is the zth column of the 2D matrix $\mathbf{h}$, z=0,…,m, and c is the symbol value taken by the connected symbol nodes. The summations in (5) are performed over all $p^{(m+1)}$ combinations of m+1 symbols. Hence, the computational complexity of the localized APP computation in (5) is $O\left(2^{(m+1)^2}\right)$, where the complexity is defined as the number of times that the metric (6) is evaluated per symbol node. Following (6), we proceed to obtain the extrinsic information $\mathbf{L}_{c\to s}^{dt}$ of downtrack transmitted symbols $\mathbf{x}$ by subtracting the *a priori* LLR from the APP LLR, which will be used as the input to symbol nodes as,

$$\mathbf{L}_{c\to s}^{dt} = \mathbf{\Lambda}^{dt} - \boldsymbol{\lambda} \quad (7)$$

where the subscript $c\to s$ denotes messages passed from check nodes to symbols nodes. Subsequently, the symbol nodes perform the summation operation over all incoming check-to-symbol messages except the message on the current edge in order to obtain the extrinsic symbol-to-check information $\mathbf{L}_{s\to c}^{dt}$ to the check node. After several message-passing iterations, the downtrack detector passes the extrinsic information $\mathbf{L}_{dt\to ct}$ of transmitted symbols $\mathbf{x}$ to the crosstrack detector as an *a priori* input for further detection.

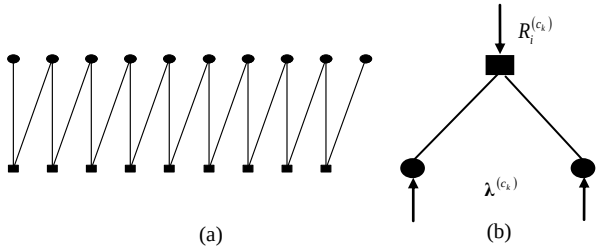

Fig. 2. Symbol-Level 1D downtrack channel graph for a 2×2 2D matrix. for illustration purposes, transmitted bits over 2D channels are arranged into an 8×8 matrix.

B. Bit-Level SPA-Based Crosstrack Detector

For the jth column, j=0,…,l−1, the crosstrack detector views the extrinsic information $\mathbf{L}_{dt\to ct}$ of downtrack symbols $\mathbf{x}$, which is produced by the downtrack detector, as an input over a (m+1)-tap 1D binary-input channel. The other input is the *a priori* information of LDPC code bits fed back from the LDPC decoder. Similarly, these two inputs are fed into the check nodes and bit nodes of the crosstrack graph, respectively.

Since the crosstrack graph represents ISI between adjacent bits, the bit-level SPA can be used to produce the extrinsic information of LDPC code bits, which will be passed into the LDPC decoder for processing. When local iterations are invoked between the downtrack detector and the crosstrack detector, we need to further modify the SPA to generate the extrinsic information $\mathbf{L}_{ct\to dt}$ of downtrack symbols $\mathbf{x}$, which will be fed back to the downtrack detector as the *a priori* information.

C. Simplified LDPC Decoding

For clarity purposes, we take the conventional notations of LDPC decoding used in the literature, e.g., [16], [17]. Let us denote the set of bits that participate in check m by $N(m)=\{n: H_{mn}=1\}$, and the set of checks in which bit n participates as $M(n)=\{m: H_{mn}=1\}$, m=0,…, M−1 , n=0,…, N−1. Let $N(m)\backslash n$ represent the exclusion of n from $N(m)$, while $M(n)\backslash m$ represents the exclusion of m from $M(n)$. Let ε_n denote the *a priori* information of bit n. Let $\varepsilon_{m\to n}$ and $\varepsilon_{n\to m}$ denote messages passed from check m to bit n and *vice versa* in the LDPC code graph, respectively. Let β_n denote the *a posteriori* LLR of LDPC code bit n. The LLR-domain SPA decoder for LDPC decoding involves a check-node update followed by a bit-node update, which are given by (8) and (9), respectively,

$$\varepsilon_{m\to n}^{(i)} = \prod_{n'\in N(m)\backslash n}\operatorname{sign}\left(\varepsilon_{n'\to m}^{(i-1)}\right) 2\tanh^{-1}\left(\prod_{n'\in N(m)\backslash n}\tanh\left(\left|\varepsilon_{n'\to m}^{(i-1)}\right|/2\right)\right) \quad (8)$$

where the superscript i denotes the ith decoding iteration, the bit-to-check input is given by $\varepsilon_{n'\to m}^{(i-1)} = \beta_{n'}^{(i-1)} - \varepsilon_{m\to n'}^{(i-1)}$, and

$$\beta_n^{(i)} = \varepsilon_n + \sum_{m'\in M(n)}\varepsilon_{m'\to n}^{(i)} \quad (9)$$

If the syndrome check is satisfied, the LDPC decoder is terminated earlier; otherwise, the decoder continues until the maximum number of iterations is reached. The computational complexity of the SPA per bit per iteration is dominated by the LDPC check-node degree d_c. From (8), it is observed that the dominant terms in the check node operation correspond to those with the smallest reliabilities among all d_c bit-to-check inputs. If we restrict the computations to these Q least reliable inputs, $2\le Q\le d_c$, we have a low-complexity check-node operation as follows. For the mth check node, find the set $\mathbf{S}=\{S_1,S_2,\ldots,S_Q\}$ of the Q least reliable components among all d_c bit-to-check messages such that $\left|\varepsilon_{n_i\to m}\right| \le \left|\varepsilon_{n_j\to m}\right|$, $\forall i\in\mathbf{S}$, $\forall j\in N(m)\backslash\mathbf{S}$, where the LLR reliability is measured in terms of its magnitude. For each $n\in N(m)$, the check-to-bit message can be efficiently generated by only using these Q least reliable inputs in the magnitude computation rather than all d_c inputs as

$$\bar{\varepsilon}_{m\to n}^{(i)} = \prod_{n'\in N(m)\backslash n}\operatorname{sign}\left(\varepsilon_{n'\to m}^{(i-1)}\right) 2\tanh^{-1}\left(\prod_{i\in\mathbf{S},n_i\neq n}\tanh\left(\left|\varepsilon_{n_i\to m}^{(i-1)}\right|/2\right)\right) \quad (10)$$

while the sign computation is still done by an XOR operation over the signs of all d_c inputs. In practical implementations, the magnitude computation is usually performed based on the forward/back recursion over a single-state trellis of single-parity-check (SPC) code as described in [18]. Since the simplified decoder requires only Q check-node inputs in the magnitude computation, the computational cost per check node is reduced to 3Q−4 core operations, where the core operation between the LLR of two random variables is defined in [18]. For

large values of d_c, this is a significant reduction from $3d_c-4$ core operations as required by the conventional check-node operation in (8). For example, the percentage of the complexity reduction with d_c=32 and Q=3 is 90.6%. Of particular interest is the special case of Q=2. The simplified decoder with Q=2 bears much resemblance to the conventional Min-Sum decoder, which is the simplest variant of the SPA, in that the decoder needs to know only the least two inputs to each check node. For the simplified decoder, the magnitude of the generated check-to-bit message on the minimum (or the second minimum, respectively) position is directly given by the second minimum (or the minimum, respectively) magnitude, which is identical to the Min-Sum algorithm; while the magnitudes of check-to-bit messages on the other d_c-2 reliable positions involve the computation of a correction factor given by check node inputs a and b, i.e.,

$$f(a,b)=\log\left(1+\exp\left(-|a+b|\right)\right)-\log\left(1+\exp\left(-|a-b|\right)\right) \qquad (11)$$

which can be approximated by using a lookup table. Hence, the complexity of the simplified LDPC decoder with Q=2 is slightly higher than that of the Min-Sum decoder, as the only computation required is d_c lookup operations to approximate correction factors for all d_c check nodes per iteration.

The decomposition of a 2D channel into a downtrack channel and a crosstrack channel significantly reduces the number of short cycles in the channel graph. For a 2×2 matrix, no cycles exists in the channel graph in either the downtrack or the crosstrack dimension as shown in Fig. 2 (a), thus allowing the SPA to achieve the performance of the optimal trellis-based BCJR algorithm. In contrast, the number of length-4 cycles in the channel graph constructed directly based on the 2D matrix as in [9], [10] is much larger, e.g., 8,320 for a 64×64 bit matrix. On the other hand, a properly designed high-rate LDPC code, though eliminating length-4 cycles, generally has length-6 cycles in the code graph [21], [22]. It is known that the SPA tends to over-estimate the reliabilities of messages propagated over a code graph with short cycles. To improve the performance, we further introduce an attenuation factor to scale down LLR reliabilities produced by either the 2D detector or the LDPC decoder. In the simulations, we reduce the magnitude of check-to-bit messages produced by LDPC check nodes by multiplication with a factor α<1.

Fig. 3 shows the block diagram of the graph-based receiver. The LDPC code graph, which forms the upper two layers, is a Tanner graph describing connections between check nodes and bit nodes. The third layer denotes the set of crosstrack check nodes, which are concatenated to LDPC code bits. This layer is also connected to the layer of symbol nodes in the downtrack graph to provide the soft information of downtrack symbols; while the channel output is input to the layer of downtrack check nodes. In addition to the overall iteration between the 2D detector and the LDPC decoder, various types of local iterations can be introduced to achieve tradeoffs between performance and complexities, e.g., local iterations between check nodes and symbol (bit, respectively) nodes in the downtrack (crosstrack, respectively) graph, iterations between the downtrack and the crosstrack detector, iterations between the crosstrack detector and the LDPC decoder, and iterations between LDPC check nodes and LDPC bit nodes. Since the LDPC code graph is concatenated with the 2D channel graph to form an overall graph, we can effectively propagate soft messages through the combined graph in a flooding schedule.

IV. Performance Results

The LDPC code used in the simulation is constructed based on extended Reed-Solomon (eRS) codewords [21] defined in GF(2^5), where a 4×32 sub-array of 32×32 permutation matrices is taken to form a 128×1024 parity-check matrix with column weight 4 and row weight 32. The null space of the parity-check matrix gives a (4, 32)-regular LDPC code of rate 0.889 and code length 1024. The LDPC codeword is written into a 32×32 matrix before transmission over a 2×2 2D channel.

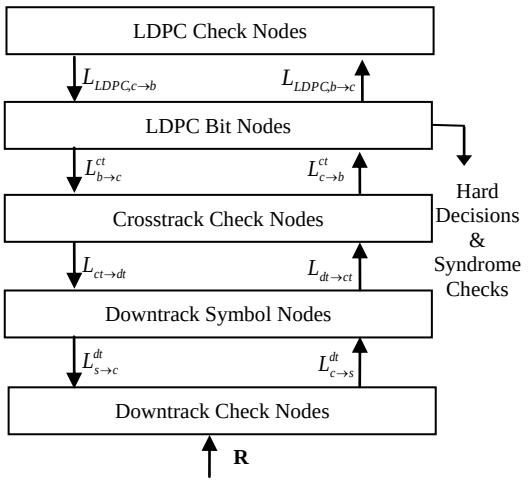

Fig. 3. Block diagram of the graph-based receiver for LDPC coded 2D channels.

We first present the bit-error-rate (BER) and frame-error-rate (FER) performance with various detector/decoder schedules. The conventional BCJR-based receiver [11] performs 5 turbo iterations between the 2D detector and the LDPC decoder; while a maximum of 30 iterations is used in the LDPC decoding per turbo iteration (i.e., schedule labeled 5/30). The graph-based receiver adopts the flooding schedule, i.e., performs at most 50 iterations between the 2D detector and the LDPC decoder; while one message-passing step between LDPC check nodes and bit nodes is performed per iteration (i.e., labeled 1/1). Fig. 4 shows that the flooding schedule performs noticeably better than the conventional schedule, and the performance gain tends to widen with increasing SNR. Moreover, the graph-based detector achieves performance identical to the BCJR (1/1)

detector, since no cycle exists in the downtrack/crosstrack graph for 2×2 2D channels. Hence, in the following we only focus on the flooding schedule.

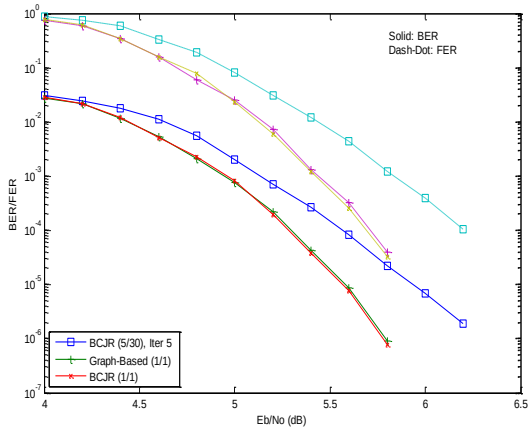

Fig. 4. Performance of the graph-based iterative receiver with various detector/decoding schedules over a 2×2 2D channel.

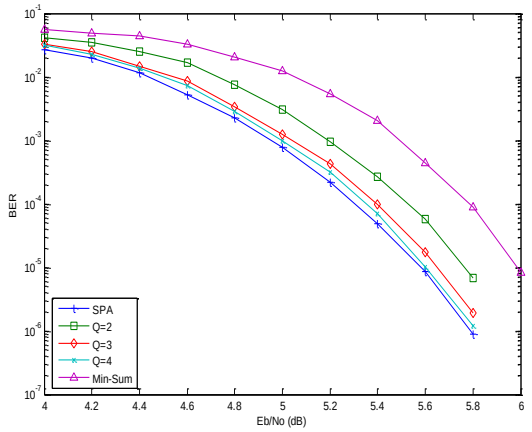

Fig. 5. BER performance of the graph-based iterative receiver with various simplified LDPC decoders over a 2×2 2D channel.

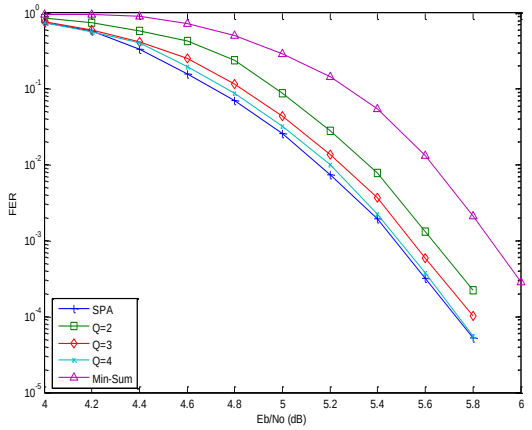

Fig. 6. FER performance of the graph-based iterative receiver with various simplified LDPC decoders over a 2×2 2D channel.

In Fig. 5, we present the performance of the graph-based receiver with the simplified LDPC decoder. It is shown that the Min-Sum decoder performs worse than the conventional SPA decoder by 0.5 dB at the BER of 10^{-5}. The simplified decoder with Q=2 achieves a performance gain of 0.2 dB over the Min-Sum decoder. The decoder with Q=4 performs very close to the SPA by restricting the check node computations to 4 instead of all 32 bit-to-check inputs, thus achieving a complexity reduction of 87.5%. The frame error rate (FER) performance of the receiver is shown in Fig. 6, where similar performance differences are observed.

Fig. 7 shows the average number of overall iterations required for various receivers. It is shown that the receiver with the Min-Sum decoder requires the largest number of iterations throughout the simulated SNR range. At E_b/N_o=5.8 dB, the receivers with simplified LDPC decoders need approximately 5 iterations out of 50 iterations to find a valid LDPC codeword

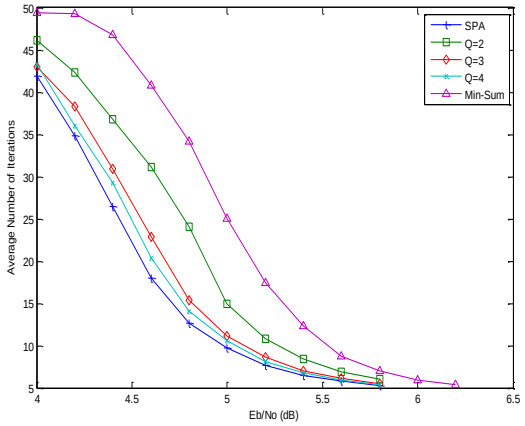

Fig. 7. Average number of overall iterations required by the graph-based iterative receiver with various simplified LDPC decoders over a 2×2 2D channel.

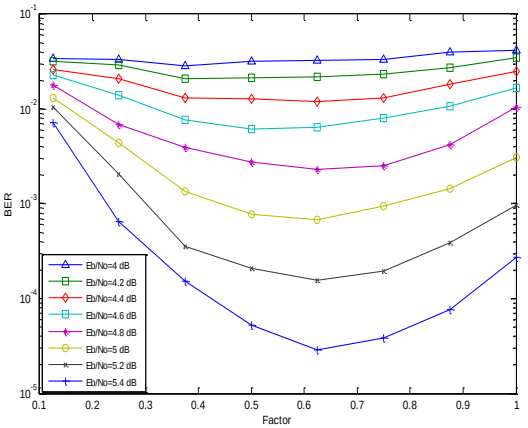

Fig. 8. Performance of the graph-based iterative receiver with various attenuation factors over a 2×2 2D channel.

In Fig. 8, we present the performance of the receiver with various attenuation factors. The value of Q in the LDPC decoding is set to Q=2. It is shown that at lower values of E_b/N_o, BER changes slightly with the factor. With increasing SNR, BER performance can be significantly improved by properly scaling down LLR reliabilities propagated over the channel or the code graph. Moreover, the BER curve is shown to be a good cost function for optimizing the attenuation factor. Fig. 8

shows that the optimal factor is α=0.625 for E_b/N_o=5.4 dB. Ideally, the optimal factor should be chosen according to SNR values, the 2D channel matrix, and LDPC coding schemes, which, however, is computationally intensive when determined by simulations.

To reduce the complexity, we assume that the attenuation factor a remains constant throughout the simulation, i.e., we set a=0.625 for the considered 2D matrix. In fact, little performance improvement is obtained when the factor is optimized specifically for different iteration or SNR. Fig. 9 shows that with the introduced attenuation factor, the graph-based receiver using the simplified LDPC decoder (Q=2) performs even better than the BCJR-based scheme (1/1) with the full-complexity SPA-based LDPC decoder by a non-negligible gain at the BER of 10^{-5}. Note that the LDPC decoder with Q=2 succeeds in reducing the number of core operations required per check node by a significant percentage of 93.75%. Similar performance gain is also observed for a longer (4096, 3750) eIRA code. The eIRA code is constructed based on the method in [15], whose parity-check matrix has a column weight 5 in the randomly constructed left submatrix and a column weight 2 in the double-diagonal right submatrix; while the rightmost column has weight 1. Compared with using a fixed factor of a=0.625 for all columns of the irregular parity-check matrix, there is little performance improvement in using attenuation factors individually optimized for various column weights. Moreover, Fig. 9 suggests that the performance gain tends to become more evident with increasing SNR values.

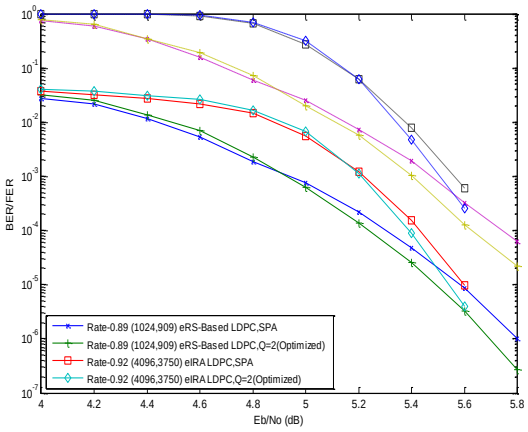

Fig. 9. Performance of the graph-based iterative receiver with the Q=2 simplified LDPC decoder over a 2×2 2D channel.

V. Conclusion

In this paper, we propose a fully graph-based iterative detection and decoding scheme for LDPC coded *generalized* 2D channels. The proposed 2D detector consists of a symbol-level SPA-based downtrack detector and a bit-level SPA-based crosstrack detector. A simplified LDPC decoder is also proposed to produce soft information for the 2D detector, which significantly reduces the decoding complexity by restricting check-node computations to dominant bit-to-check inputs. An attenuation factor is further introduced and optimized to compensate for the over-estimation of LLR reliabilities. The decomposition of a 2×2 2D channel into a downtrack channel and a crosstrack channel completely removes short cycles in the channel graph in either the downtrack or the crosstrack dimension. Simulation results show that with properly adjusted parameters, the proposed fully-graph-based receiver achieves better performance and also significantly reduces the decoding complexity as compared with the previous trellis-based scheme for generalized 2×2 2D channels.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

Yingying Li and Zhiliang Qin conducted the research and wrote the paper; Lianghui Zou, Yu Qin, and Qidong Lu analyzed the data and contributed to the simulation studies. All authors had approved the final version.

References

[1] K. Wei, J. Li, L. Kong, F. Shu, and F. C. M. Lau, "Page-Based dynamic partitioning scheduling for LDPC Decoding in MLC NAND flash memory," *IEEE Trans. on Circuits and Systems II: Express Briefs*, vol. 66, pp. 2082-2086, Dec. 2019.

[2] S. Jeong and J. Lee, "Signal detection under multipath intersymbol interference in staggered bit-patterned media recording systems," *IEEE Magnetics Letters*, vol. 10, Jan. 2019.

[3] Z. Li, W. Chen, C. Rea, M. G. Blaber, *et al*., "Head and media design for curvature reduction in heat-assisted magnetic recording," *IEEE Trans. Magnetics*, vol. 53, July 2017.

[4] P. J. Bouvet and Y. Auffret, "On the achievable rate of multiple-input–multiple-output underwater acoustic communications," *IEEE Journal of Oceanic Engineering*, vol. 45, pp. 1126-1137, June 2019.

[5] Y. Zhang, Y. C. Liang, M. W. Chia, and E. C. Y. Peh, "FBMC duplexing: Advantages and problems," in *Proc. IEEE ICICS*, Singapore, Dec. 2015.

[6] R. A. Ashrafi and A. E. Pusane, "An efficient equalization technique for multi-level cell flash memory storage systems," in *Proc. 25th Signal Processing and Communications Applications Conference (SIU)*, Antalya, Turkey, May 2017.

[7] M. K. Cheng, J. Campello, and P. H. Siegel, "Soft-decision Reed-Solomon decoding on partial response channels," in *Proc. IEEE GlobeCom*, Taipei, Taiwan, Nov. 2002, pp. 1026–1030.

[8] M. Carosino, J. Yu, Y. Chen, M. Mehrnoush, *et al*., "Iterative detection and decoding for TDMR with 2-D intersymbol interference using the four-rectangular-grain model," *IEEE Trans. Magnetics*, vol. 51, Feb. 2015.

[9] Y. Wu, J. A. O'Sullivan, N. Singla, and R. S. Indeck, "Iterative detection and decoding for separable two-dimensional intersymbol interference," *IEEE Trans. Magnetics*, vol. 39, pp. 2115-2120, July 2003.
[10] C. K. Matcha, S. Roy, M. Bahrami, B. Vasic, and S. Garani Srinivasa, "2-D LDPC codes and joint detection and decoding for two-dimensional magnetic recording," *IEEE Trans. Magnetics*, vol. 54, Aug. 2017.
[11] J. Yao, K. C Teh, and K. H. Li, "Joint iterative detection/decoding scheme for discrete two-dimensional interference channels," *IEEE Trans. Commun*., vol. 60, pp. 3548-3555, Dec. 2012.
[12] K. Pituso, C. Warisarn, D. Tongsomporn, and P. Kovintavewat, "An intertrack interference subtraction scheme for a rate-4/5 modulation code for two-dimensional magnetic recording," *IEEE Magnetics Letters*, vol. 7, June 2016.
[13] M. Shaghaghi, K. Cai, Y. L. Guan, and Z. Qin, "Markov chain Monte Carlo based detection for two-dimensional intersymbol interference channels," *IEEE Trans. Magn.*, vol. 47, pp. 471-478, Feb. 2011.
[14] K. Liu, M. El-Khamy, and J. Lee, "Finite-Length algebraic spatially-coupled quasi-cyclic LDPC codes," *IEEE Journal on Selected Areas in Communications*, vol. 34, pp. 329-344, Dec. 2015.
[15] H. Li and L. Zheng, "Efficient puncturing scheme for irregular LDPC codes based on serial schedules," *IEEE Communication Letters*, vol. 19, pp. 1508-1511, Sept. 2015.
[16] F. Vatta, A. Soranzo, and F. Babich, "More accurate analysis of sum-product decoding of LDPC Codes using a Gaussian approximation," *IEEE Communication Letters*, pp. 230-233, Feb. 2019.
[17] I. E. Bocharova, B. D. Kudryashov, V. Skachek, and Y. Yakimenka, "BP-LED decoding algorithm for LDPC codes over AWGN channels," *IEEE Trans. Inform. Theory*, vol. 65, pp. 1677-1693, Aug. 2018.
[18] Z. Qin, K. Cai, and S. Zhang, "Reduced-complexity decoding of high-rate LDPC codes over partial-response channels," in *Proc. IEEE ICCS*, Singapore, 2010.
[19] G. M. Vitetta, P. D. Viesti, E. Sirignano, and F. Montorsi, "Multiple Bayesian filtering as message passing," *IEEE Trans. Signal Processing*, vol. 68, pp. 1002-1020, Jan. 2020.
[20] T. Sopon and W. Wongtrairat, "Investigation of graph-based detection in BPMR system with multi-track processing," in *Proc. International Conference on Electrical Engineering, Computer, Telecommunications and Information Technology (ECTI-CON)*, July 2016.
[21] M. Asif, W. Zhou, J. Saidi Ally, and N. A. Khan, "An Algebraic construction of quasi-cyclic LDPC codes based on the conjugates of primitive elements over finite fields," in *Proc. IEEE 18th International Conference on Communication Technology (ICCT)*, Chongqing, China, Oct. 2018.
[22] Q. Diao, J. Li, S. Lin, and I. F. Blake, "New classes of partial geometries and their associated LDPC codes," *IEEE Trans. Inform. Theory*, vol. 62, pp. 2947-2965, Dec. 2015.

Yingying Li received the B.Eng. in the School of Communication Engineering from Harbin Institute of Technology, Weihai, China, in 2008. and the M.Eng. degree in the School of Electronic and Communication Engineering from Shandong University, Weihai, China, in 2015. She is currently with Weihai Beiyang Electrical Group co. Ltd, Weihai, China. Her current research interests include fault diagnosis, artificial intelligence, computer vision.

Zhiliang Qin was born in 1974. He obtained the B. Eng. degree from the Beijing Institute of Technology (BIT) in 1995, the M. Eng. degree from the Graduate School of China Academy of Engineering Physics (CAEP) in 1998, and the Ph. D. Degree from the Nanyang Technological University (NTU), Singapore in 2003. From 2002 to 2019, he worked at the Agency for Science, Technology, and Research (A*STAR) in Singapore as the Scientist in the area of algorithm developments for machine learning, signal processing, data analytics, optimization theories, and data storage systems. From 2019 to present, he is the Deputy Chief Engineer at the Weihai Beiyang Electric Group. Co. Ltd, Weihai, Shandong, China. Dr Qin published around 80 SCI/EI technical papers and authored three U.S. patents. He frequently takes the role of being the reviewer of research journals and being the Technical Committee Member (TPC) of international academic conferences on artificial intelligence (AI) and signal processing, including the ICSPS 2020, MLMI2020, ICCCR2021, AIACT 2021, ICEEMT 2021, etc. He is an IEEE member and serves on the Editorial Board of Journal of Communications (JCM) and the American Journal of Computer Science and Technology (AJCST).

Lianghui Zou was born in Weihai, China, in 1993. He received the B. Eng. and the M. Eng. degree from Hebei Technological University and Inner Mongolia Technological University, in 2015 and 2017, respectively. He is currently with Weihai Beiyang Electrical Group Co., Ltd, Weihai, China. His research interests include computer visions, and data analytics.

Yu Qin was born in Weihai, China, in 1992. He received the B.S. and the M.S. degree in the School of Mathematics from Shandong University, Jinan, China, in 2014 and 2017, respectively. He is currently with Weihai Beiyang Electrical Group Co., Ltd, Weihai, China. His current research interests include artificial intelligence, speech recognition, signal processing, and natural language processing.

Qidong Lu was born in Yantai, China, on January 17, 1992. He received the B.Eng. in the College of Mechanical and Electronic Engineering and the M.Eng. degree in the College of Electrical Engineering and Automation from Shandong University of Science and Technology, Qingdao, China, in 2016 and 2019, respectively. He is currently with Weihai Beiyang Electrical Group Co., Ltd, Weihai, China. His current research interests include fault diagnosis, artificial intelligence, and speech recognition.