Improving the Decoding Performance of High-Rate GLDPC Codes in Low Error-Rate Applications

Sherif Elsanadily
Department of Communications
Faculty of engineering, Helwan university
Cairo, Egypt
Email: sherif_elsanadily@yahoo.com

Ashraf Mahran
Department of Avionics
Military Technical College
Cairo, Egypt
Email: a.mahran@ieee.org

Osama Elghandour
Department of Communications
Faculty of engineering, Helwan university
Cairo, Egypt
Email: osamaelghandour90@gmail.com

***Abstract*—This paper presents a new evaluation and performance comparison of a reliability-based iterative decoder for Generalized LDPC codes with BCH subcodes, using Soft-Input Soft-Output (SISO) Chase algorithm, compared with the same algorithm employing Hamming subcodes. While the single-error correction code with length n is accorded a preference over the double error correction one with length 2n (in terms of error performance over AWGN channels), this article shows that the GLDPC code with BCH subcodes surpasses the Hamming-based one under nearly the same overall code rate. It also converges in less than half the number of iterations compared to different corresponding Hamming-based soft-decision decoding (SDD) systems. As a case study, Gallager-based global LDPC code is applied and the decoder is considered over AWGN channel. The simulation results are showing the performance against the code block length, number of iterations and Chase algorithm parameters.**

***Keywords*—Iterative decoding, APP decoding, Hamming codes, Chase algorithm, LDPC codes.**

I. Introduction

In the last two decades, the low-density parity-check (LDPC) code [1] has attracted researchers attention intensively and is proven to surpass turbo code [2] for large block lengths. The bipartite graph of LDPC, containing the symbol nodes and constraint nodes, represents the parity matrix H of the code which is supposed to be very sparse. LDPC codes can be decoded by many methods such as Message Passing (MP) algorithm which is the most famous and effective one. By Tanner in [3], extending the sparse matrix construction of the conventional LDPC code generates a generalized code where SPC check nodes are replaced with error-correcting block codes, called as subcodes, component codes or constituent codes. Using convenient code constructions and powerful decoders at the check nodes operating in parallel, Generalized LDPC coding schemes can be implemented and better performance, than LDPC, can be obtained [4], [5], [6]. Decisions obtained by the parallel check node decoders are passed to the MP decoder operating on the bipartite graph of the global LDPC code.

Due to high complexity of decoding some linear block codes, the subcodes of G-LDPC are limited to simple ones. Generalized LDPC codes were studied in [7], [8] where Hamming code is employed as component codes and Chase-based SISO decoders at the check nodes were studied over AWGN channel. In [9], the authors also considered the same code over BSC channel employing the weighted bit-flipping voting decoding method. Afterwards, Generalized LDPC codes with BCH codes as subcodes were investigated. They achieve higher coding gain compared to the irregular LDPC code [10]. Boutros in [11] investigated GLDPC codes with BCH subcodes over AWGN channel but with ML decoding. In [12], [13], a MP algorithm with an algebraic bit-flipping decoding performed on the BCH and RS subcodes was utilized over binary symmetric channel (BSC) and binary erasure channel (BEC). Different and modern bit-flipping algorithms were also proposed using the simple Hamming subcodes to deliver superior performance [14], [15].

The authors in [16] used the Progressive Edge Growth Algorithm to construct the global matrix of BCH-GLDPC code where SISO decoding is operated with trellis-based Log-MAP algorithm employed on BCH subcodes. Another approach, utilizing the protograph LDPC as a global construction for GLDPC with arbitrary subcodes over BEC and binary-input symmetric channel, was studied in [17]. A different research was introduced in [18] where Pearls belief propagation and BCJR decoding algorithms were employed for Hamming-based GLDPC codes over Rayleigh channels.

In the literature, not like LDPC codes, the GLDPC codes have not been deeply investigated yet. While multiple researches tend to use different coding structures with various decoding algorithms over several channel models, the performance of GLDPC is not evaluated in terms of different component codes on common conditions (common construction, decoding algorithm, code rate and channel model) In this paper and different from all previous work, we study the performance of Gallager-constructed Generalized LDPC codes with BCH against Hamming component codes over AWGN channel by using APP decoding algorithm. We make use of the channel measurements by employing the MP algorithm on the global LDPC graph nodes and applying a SISO Chase-2 algorithm on the check nodes.

This paper is organized as follows. In Section II, the structure of the Generalized LDPC codes with binary BCH component codes are first introduced. In Section III, a full description of the Chase decoding procedure is presented. In Section IV, the simulation and comparison results are given. In Section V, we conclude the paper.

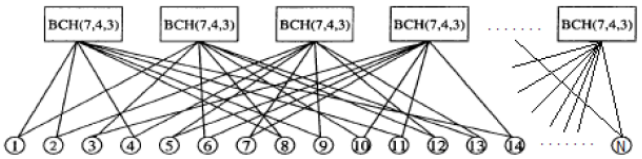

Fig. 1. Graph of $(N, 2, 7)$ GLDPC code

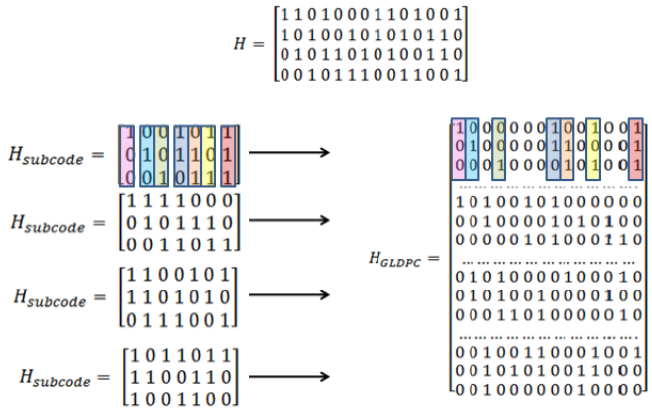

Fig. 2. Generalized LDPC Parity matrix with random assignment

II. CONSTRUCTION OF GLDPC CODES

Generalized LDPC (GLDPC) codes are presented by Tanner in 1981 by replacing the simple constraint nodes (SPC codes) with generalized block codes (subcodes) [3]. Therefore, the Generalized LDPC graph can be defined with the same matrix H of a LDPC code, but with a different interpretation of the check nodes as illustrated in Fig. 1.

The new parity check matrix of this generalized code can be constructed as following:

In each row in H, each 1 is replaced by a column from the block code parity matrix (subcode) at the check node. Each 0 is replaced by a zero column vector. As proved by Tanner, a random assigning for the columns of the constituent code parity matrix to the variable nodes positions can produce good codes with better minimum distance. Starting with a regular LDPC (its column weight J , its row weight n and its $M \times N$ parity matrix H), the n nonzero column positions will be randomly assigned to obtain a $M(n-k) \times N$ parity check matrix H_{GLDPC}. To simplify the construction as followed by Gallager [19], the parity matrix H of the (N, j, n) GLD code is partitioned into j separate submatrices where each one with weight one in every column. The symbol nodes are indexed $v_b, b \in \{1, .., N\}$ and the check nodes $c_a, a \in \{1, .., M\}$. The obtained code has overall rate $\geq 1(n-k)j/n$ and equality is achieved if the resultant matrix has a full rank. a GLDPC matrix with (7,4,3) BCH as subcode is shown as a simple example in Fig. 2.

III. CHASE DECODING ALGORITHM

Our presented decoding algorithm of Generalized LDPC is the sub-optimum MP algorithm with the Chase-2 algorithm [20] employed as APP decoding on the check nodes. Our method will use the BCH as subcode due to its high error correcting capability and therefore a better performance will be obtained. For simplicity, all check nodes are represented with the same BCH code of parameters (n, k, d). First, Let us consider transmitting a binary codeword of length N of (N, j, n) GLDPC code on AWGN channel with binary symbols $\{+1, -1\}$. The observed vector at the receiver denoted R_N and given by $R_N = E_N + G_N$ where E_N is the modulated vector of N symbols and G_N is the N channel noise samples. The received codeword includes varieties of component codewords located at positions according to the code graph where every check node connected to n symbol nodes forming a specified word of the (n, k, d) BCH code.

At any check node, the input to the chase decoder is $R = \{r_1, , r_j, .., r_n\}$ corresponding to $E = \{e_1, , e_j, .., e_n\}$ and its hard demodulated values $Y = \{y_1, , y_j, .., y_n\}$. Second, a set of codewords is chosen as followed by Chase to be the most probable ones that contain the transmitted codeword with the least number of errors. That can be achieved by exploiting the reliability information and inverting all possible combinations of the $p = \lfloor d/2 \rfloor$ demodulated symbols with the least reliable positions (LRPs). The reliability information is defined in [21] in the log-likelihood ratio (LLR) of decision y_j as

$$\Lambda(y_j) = \ln\left(\frac{pr(e_j = +1/r_j)}{pr(e_j = -1/r_j)}\right) = \left(\frac{2}{\sigma^2}\right) r_j \qquad (1)$$

A set Z of error patterns with all possible errors confined to p LRPs positions of Y is used to modify Y yielding list of test patterns $T^q (q \in \{1, ..., 2^{\lfloor d/2 \rfloor}\}, T^q = Z^q + Y)$. Then every T^q in the list is decoded using algebraic (Berlekamp-Massey) decoder and the valid candidate decoded word C^q is stored in a list Ω. Decision $D = \{d_1, , d_j, .., d_n\}$ is obtained using the rule [22]

$$D = C^q \; if \; |R - C^q|^2 \leq |R - C^l|^2 \text{for every } l \in \Omega \qquad (2)$$

Now, the output soft value of every decoded symbol of the subcode should be calculated to be sent back on the connected edges to the GLDPC symbol nodes as followed by the MP algorithm in an iterative method to obtain a final decision after certain number of iterations or a syndrome condition should be satisfied.

Two codewords $C^{+1(j)}$ and $C^{-1(j)}$ from two sets of Ω with minimum Euclidean distance from R are to be determined to calculate the decision reliability at the decoder output [21]. The decision D is one of them and the second one $C = \{c_1, , c_j, .., c_n\}$ called as competing codeword of D with $c_j \neq d_j$ should be found.

The soft outputs are generated in the LLR domain using the following approximation formula:

$$r_j^{'} = \left(\frac{|R - C|^2 - |R - D|^2}{4}\right) d_j \qquad (3)$$

Due to need of reducing the computational complexity of the chase decoder, the number of test pattern is chosen to be reduced and probability of finding the competing word C

decreases. If not found, the following alternative and efficient formula [22] is used

$$r_j^{'} = \beta \times d_j \text{ with } \beta \geq 0 \tag{4}$$

With taking into account the difference in the standard deviation of the samples at the input and output of the SISO decoder in the decoding steps, a scaling factor preferred to be put for faster convergence rate as follows:

$$W(m+1) = R + \alpha(m)W(m) \tag{5}$$

Where $W(m)$ is the extrinsic information at the output of the SISO decoder during the m-th iteration which is multiplied by the scaling factor, added to the channel observed values and the result is considered as a priori information for the decoder at the next $(m+1)$ iteration.

IV. SIMULATION RESULTS

In [7], Lentmaier studied the GLDPC with $(15, 11, 3)$ Hamming subcodes for low overall code rate applications $(R = 0.46)$ Furthermore, the GLDPC codes with $(64, 57, 4)$ extended Hamming subcodes in [8] were investigated to improve the distance properties and simulated with code rate of $(R = 0.781)$ but the error performance gets much worse.

The GLDPC codes used here by Matlab simulations are constructed based on Gallager representation with $(64, 51, 6)$ BCH subcode chosen with double-error capability to improve the error performance, while keeping a moderate high code rate $(R \cong 0.6)$. Therefore, for keeping this rate, only codes with $j = 2$ are considered.

We choose $(32, 26, 4)$extended Hamming code for comparison purposes as the latter produces GLDPC code with nearly the same overall code rate $(R = 0.625)$ as BCH-based GLDPC. This extended Hamming subcode can also correct one bit successfully from 32 bits of the GLDPC code. That leads to a scheme of correction capability of 2 bits among 64 bits which is supposed to give the same performance as BCH subcode or even better. Simulation results are shown in Fig. 3 for different number of iterations for code block length $N = 2048$.

It is clear that the error rate reached by the presented decoding (with BCH subcodes) is not only much better than the corresponding $(32, 26, 4)$ Hamming but also in less than half the number of iterations, executed by Hamming-based code, yielding faster convergence. The latency of decoding BCH in the SISO decoder cant be neglected but when the number of executed iterations in both systems is put into consideration, the effect of the mentioned latency diminishes. The performance is still improved especially in the error floor region as the code length increases as shown in Fig. 4 and Fig. 5. It is worth mentioning that the $(16, 11)$ ext-Hamming subcode if employed in our GLDPC may often lead to the same error performance as the $(64, 51)$BCH-based GLDPC but at cost of high degradation in the code rate $(R = 0.375)$.

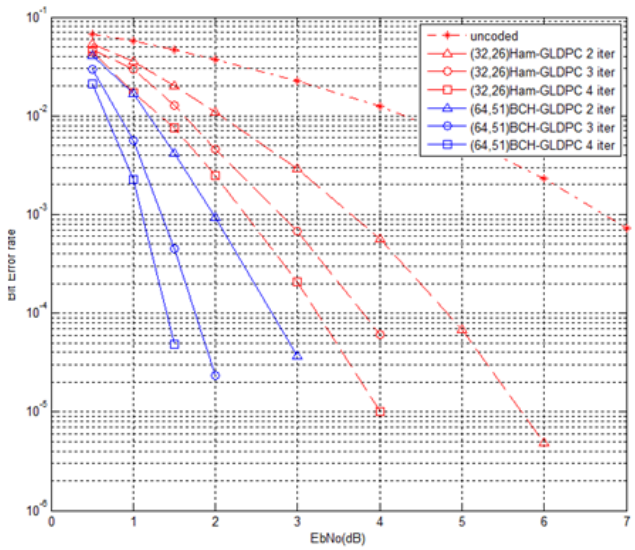

Fig. 3. Comparison of BERs of GLDPC codes with length $N = 2048$ for both Hamming and BCH component codes decoded by Chase decoder at $p = \lfloor d/2 \rfloor$ LRPs

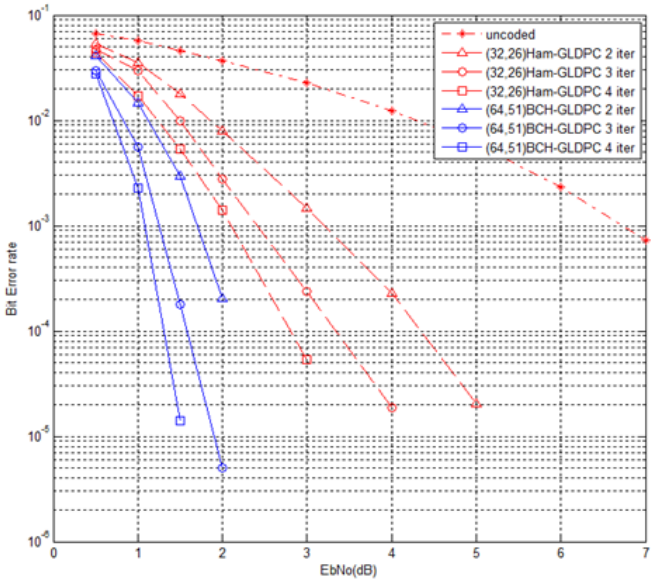

Fig. 4. Comparison of BERs of GLDPC codes with length $N = 4096$ for both Hamming and BCH component codes decoded by Chase decoder at $p = \lfloor d/2 \rfloor$ LRPs

V. CONCLUSION

We have shown that the minimum distance of the BCH-based GLDPC codes for large block lengths is growing linearly with the code length and leads to a better performance compared to Hamming-based GLDPC codes when it is used with a good SDD algorithm such as Chase decoding over AWGN channel. This scheme can be used in very low error-rate applications with a high code rate if multi-core high speed processors are provided to overcome any decoding latency of BCH subcodes.

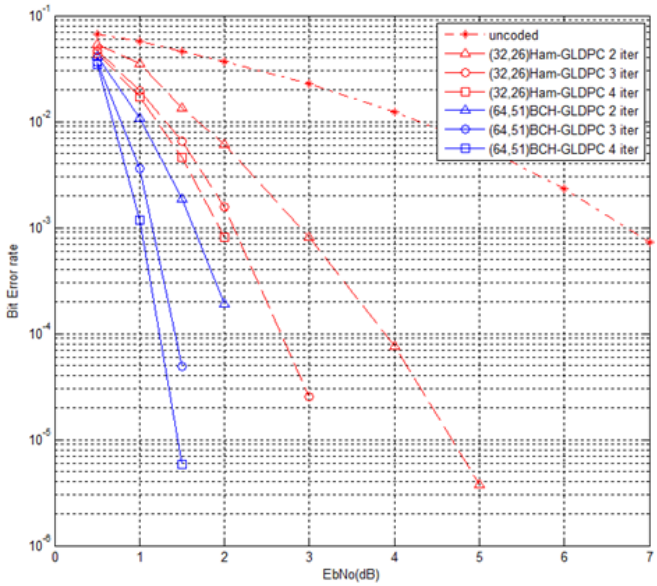

Fig. 5. Comparison of BERs of GLDPC codes with length $N = 8192$ for both Hamming and BCH component codes decoded by Chase decoder at $p = \lfloor d/2 \rfloor$ LRPs

REFERENCES

[1] D. J. C. MacKay and R. M. Neal. Near shannon limit performance of low density parity check codes. *Electron. Lett.*, 32:1645–1646, 1996.

[2] Berrou C. and Glavieux A. Near optimum error correcting coding and decoding: turbo-codes. *IEEE Trans. Commun.*, 44:1261–1271, 1996.

[3] M. Tanner. A recursive approach to low complexity codes. *IEEE Trans. Inf. Theory*, IT-27:533–547, 1981.

[4] G. Liva, W. Ryan, and M. Chiani. Quasi-cyclic generalized ldpc codes with low error floors. *IEEE Trans. on Communications*, 56(1):49–57, January 2008.

[5] D. Mitchell, M. Lentmaier, and D. Costello. On the minimum distance of generalized spatially coupled ldpc codes. *Proc. IEEE ISIT*, pages 1874–1878, July 2013.

[6] P. Olmos, D. Mitchell, and D.J. J. Costello. Analyzing the finite-length performance of generalized ldpc codes. *Proc. IEEE ISIT*, pages 2683–2687, June 2015.

[7] M. Lentmaier and K.S. Zigangirov. On generalized low-density parity check codes based on hamming component codes. *IEEE Commun. Lett.*, 3(8):248250, 1999.

[8] S. Hirst and B. Honary. Application of efficient chase algorithm in decoding of generalized low-density parity-check codes. *IEEE Commun. Lett.*, 6(9):385–387, 2002.

[9] S. Hirst and B. Honary. Decoding of generalized low-density parity check codes using weighted bit-flip voting. *IEE Proc.-Commun.*, 149(9):1–5, Feb. 2002.

[10] K.Chung. Generalised low-density parity-check codes with binary cyclic codes as component codes. *IET Communications*, pages 1710–1715, March 2012.

[11] J. Boutros, O. Pothier, and G. Zemor. Generalized low density (tanner) codes. *Proceedings ICC 99*, 1(9):441–445, June 1999.

[12] Mladinovic N. and Fossorier M.P.C. Generalized ldpc codes with reed solomon and bch codes as component codes for binary channels. *Globecom*, 2005.

[13] Mladinovic N. and Fossorier M.P.C. generalized ldpc codes and generalized stopping sets. *IEEE Trans. Commun.*, 56:201212, 2008.

[14] S. Elsanadily, A. Mahran, and O. Elghandour. Two-side state-aided bit-flipping decoding of generalized low density parity check codes. *IEEE Commun. Lett.*, 21(10):2122–2125, Oct. 2017.

[15] S. Elsanadily, A. Mahran, and O. Elghandour. Classification-based algorithm for bit-flipping decoding of gldpc codes over awgn channels. *IEEE Commun. Lett.*, 22(8):1520–1523, Aug. 2018.

[16] Ping Wang, Guangxia Li, Hongpeng Zhu, and Xinying Sun. Generalized ldpc codes for deep space communication systems. *Signal Processing (ICSP), IEEE 11th International Conference on Signal Processing*, 2:1279–1282, 2012.

[17] Asit Kumar Pradhan and Andrew Thangaraj. Near-capacity protograph doubly-generalized ldpc codes with block thresholds. *The 2016 IEEE International Symposium on Information Theory (ISIT)*, pages 2534–2538, 2016.

[18] Yangwen Yu, Yanan Han, Lijun Zhang, and Lee Lung Cheng. (7, 4)-hamming generalized ldpc code for rayleigh fading channel. *The 2015 IEEE 9th International Conference on Anti-counterfeiting, Security, and Identification (ASID)*, pages 171–175, 2015.

[19] R. G. Gallager. Low-density parity-check codes. *Cambridge, MA: M.I.T. Press*, 1963.

[20] D. Chase. A class of algorithms for decoding block codes with channel measurement information. *IEEE Trans. Inform. Theory*, IT-18:170179, Jan. 1972.

[21] R. M. Pyndiah. Near-optimum decoding of product codes: Block turbo codes. *IEEE Trans. Commun.*, 47:10031010, Aug. 1998.

[22] R. Pyndiah, A. Glavieux, A. Picart, and S. Jacq. Near optimum decoding of products codes. *Proc. IEEE GLOBECOM94 Conf., San Francisco, CA*, 1(3):339343, Nov. 1994.