A greedy search based method with optimized lower bound for QC-LDPC codes

Ambar Bajpai[1], Abhishek Kalsi[2], Suvit Nakpeerayuth[3], Piya Kovintavewat[4], and Lunchakorn Wuttisittikulkij[3]

[1]*Department of Electronics and Communication Engineering, Sreenidhi Institute of Science and Technology, Hyderabad, India*

[2]*Department of Electrical Engineering, Indian Institute of Technology, Ropar, India*

[3]*Department of Electrical Engineering, Faculty of Engineering, Chulalongkorn University, 254, Phyathai Road, Pathumwan, Bangkok, 10330, Thailand*

[4]*Data Storage Technology Research Center, Nakhon Pathom Rajabhat University, Thailand*

E mail: ambarbajpai@gmail.com, abhishekkalsi@iitrpr.ac.in, nsuvit@chula.ac.th, piya@npru.ac.th, and Lunchakorn.W @chula.ac.th

***Abstract*— Autonomous decentralized system is need of the dynamically varying society. Various autonomous decentralized systems architecture proposed and deployed, use of IEEE 802.11n/ac (Wi-Fi) and IEEE 802.16e (Wi-max) standards for surveillance purposes in different geographical region. These standards use LDPC codes as a channel code having better error correcting performance. This article deals with a construction of less computational complexity method for constructing exponent matrix $(3, K)$ having girth 8, 10, and 12 of quasi-cyclic low-density parity-check (QC-LDPC) codes. In this method, we first generate a base matrix and then the same matrix is further used for expanding till desired size of the exponent matrix. The construction of code deals with the generation of base matrix by a simple algorithm for girth 8, 10, and 12. Our method is flexible for any block-column length *K*. Finally, a new method is given with less computational complexity with optimized CPM size.**

Keywords*— LDPC codes; Channel coding; Circulant permutation matrix (CPM); Girth; Quasi-Cyclic (QC) LDPC codes*

I. INTRODUCTION

LDPC stands for Low-density parity-check codes, which are error correcting codes and are known for their high throughput and good decoding performance. LDPC codes were introduced in 1962 by R. Gallager [1]. In the initial phase LDPC codes were ignored for almost three decades due to its high computational complexity for hardware implementation at that time. After that, turbo code was developed in 1992 by Berrou, whose performance was very close to the Shannon limit or channel capacity. This success led to rediscovery of LDPC codes by Mackay and Neal [2] LDPC codes leads to its performance competent to turbo codes over an AWGN (additive white Gaussian noise) channel. Tanner Graph [3] usually known as bipartite graph can be represented by LDPC codes. By bipartite graph, set of nodes further categorized into two subsets such that each subset is independent and is connected to each other. The two subsets are known as variable nodes representing columns and check nodes representing rows of a parity-check matrix respectively.

Due to the fairly large implementation complexity of LDPC codes, it requires huge amount of memory for storage. Hence a new class of LDPC codes was introduced known as quasi-cyclic LDPC (QC-LDPC) codes. QC-LDPC code is a category of LDPC codes in which there is cyclic connection between rows or columns of a sub matrix [5]. The implementation of QC-LDPC codes were based on geometry and algebraic theories and its structural properties [4]. The structural properties of the codes majorly depend on the shift values of sub-matrices and their arrangements. If we use random shift values of identity sub-matrices, then it will result into reduced girth and poor performance. Various construction methods [10-14] have constraints on no-four cycle or girth of 4. Code constructions using these methods have at least girth of six and wide range of code rates and lengths. All construction methods have certain limitations so developed codes are restricted in all or one of the properties such as code rate, code length and girth.

In addition, a recursive approach develop would be applied for wide range of girths, code rates and lengths [15]. In order to have less memory usage and localized memory access [6]-[8], shift registers can be used efficiently by a simple mechanism of address generation for less memory requirement in QC-LDPC codes encoding. The arrangement of sub matrices and the value's by which they are shifted define the structure of QC-LDPC codes. It should be noted that random shifting of sub matrices may result in the poor performance of the QC-LDPC codes.

In this paper, we presents a reduced complexity algorithm for QC-LDPC encoding, which not only eases memory constraint of shift registers but also takes the least amount of time for computing the girth of parity-check matrix **H**. In our algorithm, we will first construct base matrix of block-rows and block-columns of size 3×5. After that the same matrix is used as a sub matrix for further construction of any generalized block-column length *K*. Thus by using this method we will be able to reduce the time complexity of the encoding method by a good amount and also having less memory storage requirement due to fairly reduced amount of proposed circulant permutation matrix (CPM) size.

978-1-5090-4042-1/17 $31.00 © 2017 IEEE
DOI 10.1109/ISADS.2017.26

In the field of QC-LDPC codes construction, the construction of **H** matrix by voltage graph based LDPC tailbiting codes for $(3, L)$ of girth 8, 10 and 12 [9] is quite substantial. Authors proposed the CPM size for degree 3 matrices whose result is quite comparable to our method. Our results outperformed considerably in terms of the CPM size requirement to construct QC-LDPC codes.

Furthermore, this correspondence organized as follows. Preliminaries of QC-LDPC codes are described as in Section II. Section III deals with the construction of base matrix algorithm and the method to generate an **H** matrix for constructing QC-LDPC codes. Section IV analyzes and compares CPM size requirement by presenting in the tabular formats and finally, Section V concludes this work.

II. PRELIMINARIES

A. QC-LDPC Codes

Consider a regular QC-LDPC code whose parity-check matrix **H** of column weight *j* and row weight *k* is said to be uniform, if **H** matrix has constant row and column weight. The structure of QC-LDPC codes are generalize as $m \times n$ array of $L \times L$ CPM size as

$$\mathbf{H} = \begin{bmatrix} \mathbf{I}_{a_{11}} & \mathbf{I}_{a_{12}} & \cdots & \mathbf{I}_{a_{1n}} \\ \mathbf{I}_{a_{21}} & \mathbf{I}_{a_{22}} & \cdots & \mathbf{I}_{a_{2n}} \\ \vdots & \vdots & \ddots & \\ \mathbf{I}_{a_{m1}} & \mathbf{I}_{a_{m2}} & \cdots & \mathbf{I}_{a_{mn}} \end{bmatrix}, \tag{1}$$

where $a_{ij} \in \{0,1\ldots,L-1,\infty\}$ and $I_{a_{ij}}$ for $(1 \le i \le m, 1 \le j \le n)$ is basically a CPM of size $L \times L$ obtained by cyclic shifting of rows of an identity matrix **I** by a_{ij} times. For $a_{ij} = \infty$, we will be having a zero matrix of size $L \times L$.

The shortest cycle in **H** matrix is called girth. The girth of a QC-LDPC codes will always be less than or equal to 12 for $j \ge 3$ [5]. Final **H** matrix is having a combination of *m* rows and *n* columns. The shifting or exponent matrix of **H** is defined as $\mathbf{E(H)}$ matrix and can be expressed as

$$\mathbf{E(H)} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}. \tag{2}$$

A $2l$-block length code which is mainly a cycle of length $2l$ in the Tanner graph of $\mathbf{E(H)}$. It can be represented as a shifting chain in $\mathbf{E(H)}$ as $(a_{i_1,j_1} \to a_{i_1,j_2} \to a_{i_2,j_2},\ldots,\to a_{i_l,j_l} \to a_{i_l,j_1} \to a_{i_1,j_1})$.

The presence of more short cycles leads to poor decoding performance of LDPC codes, so in order to avoid short cycles of length $2l$ a necessary condition is as follows

$$\sum_{k=1}^{l} (a_{i_k,j_k} - a_{i_{k+1},j_k}) \equiv 0 \bmod L \tag{3}$$

where $i_k \ne i_{k+1}, j_k \ne j_{k+1}$ and $i_{l+1} = i_l$.

III. BASE MATRIX ALGORITHM AND ITS GENERALIZED FORM FOR HIGHER GIRTH

This section deals with the construction of $\mathbf{E(H)}$ matrix of QC-LDPC codes by means of base matrix method. In this method, we are able to reduce time complexity for constructing **H** matrix by a good amount. Firstly we define some necessary conditions in order to obtain desired $\mathbf{E(H)}$ matrix.

A. Necessary conditions:

There are three easy rules for the generation of base matrix as follow:

- The first row and the first column of an exponent matrix both are fixed to be a zero vector.
- It is mandatory that the 2nd row will always be in the ascending order.
- Repetitions of indices are not allowed, i.e. at different indices we will have different values.

B. Base matrix generation for girth 8, 10, and 12

Let **B** be a base matrix then its CPM size *P* will be $P \le (3 \times 10K) + K \times K$ where K is defined as the size of block-column length. For the generation of $(3, K)$ matrix of girth 8, 10, and 12, actual CPM size will be the last element of the 3rd row plus one i.e. $\mathbf{E}(3, K) + 1$, we will follow a simple algorithm as follows:

The 2nd row of the base matrix is constructed according to the following steps:

- *Step 1*:- For the construction of $(2, K)$ (since 1st row and 1st column of an exponent matrix is a zero vector) matrix, we will substitute the value from 1 to *P* till we obtain the desired girth.
- *Step 2:*- Since repetition of indices is not allowed as per our necessary condition described as in Section II, so we delete that particular number from 1 to CPM size *P* and for the construction of $(2, K+1)$ matrix we will again follow the S*tep 1*.

Let **a** be an empty vector, which will store the indices of the 2nd row, after the working of algorithm i.e. indices after 0 value.

C. Algorithm for generating 3rd row of $(3, K)$ exponent matrix of girth 8, 10 and 12

Let **a**=[ ],

for $t = 1 : P$ **do**

$a = [a\ t]$

// To append all elements in "a" and to have 2nd row of base matrix as "a" //

If girth $(g \geq 8)$ // same for girth 10 and 12

break;

end

end

By using our algorithm, we will first construct $(2,5)$ matrix, which will be our base matrix and it will be used as a base matrix for $(2,6)$ exponent matrix and similarly for $(2,7)$ exponent matrix $(2,6)$ will act as a base matrix and so on. By our algorithm we have constructed 2nd row of base matrix. Now the thing is from which index we have to start, the answer is very simple for $(2,K)$ exponent matrix, we will start substituting number from 1 to CPM size *P*.

By substituting, we mean that we will first put an index at 2nd position of 2nd row leading to obtain a sub matrix of 2×2 and will check it's girth to be 8, 10 or 12 and then we will move on to the 3rd index of the 2nd row for 2×3 sub matrix, by same process we will check its girth and then follow the same procedure till we obtained 2×5 base matrix.

After generating 2×5 base matrix, our further work for the construction of 3rd row of $(3,K)$ matrix is very time convenient. The values obtained at different indices of 2nd row, i.e. $(2,K)$ added by one $(2,K)+1$ will be the corresponding indices of 3rd row $(3,K)$. Hence in this way we can obtain our desired $(3,K)$ matrix of our required girth in less complex and time consuming process. By time efficient we really mean our method to work in seconds because the last index of second row for generating consecutive matrix can be obtained by our algorithm. Further work is to just find the last element of 3rd row which is just the next indices value of 2nd row of those indices. Hence it is a very time efficient algorithm, as it can be seen in Section IV by various examples and comparing results.

IV. ANALYSIS AND RESULTS

In this section, we analyze our proposed algorithm by giving few examples obtained by our code construction method and further compare results with the benchmarked algorithm as in [9].

Example 1: By using our algorithm, the exponent matrix **E(H)** for the case of $K = 9$ having girth 8 is given by

$$\mathbf{E(H)} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 3 & 5 & 7 & 9 & 11 & 13 & 15 \\ 0 & 2 & 4 & 6 & 8 & 10 & 12 & 14 & 16 \end{bmatrix}$$

TABLE I: Base matrix of $(3,5)$ for girth 8

Block-row index \ Block-column index	1	2	3	4	5
1	0	0	0	0	0
2	0	1	3	5	7
3	0	2	4	6	8

The matrix is generated by the same procedure explained and the CPM sixe of $(3,9)$ matrix is last element i.e. 16+1=17.

TABLE II: The construction I refer to method I of Voltage Graph Based LDPC Tailbiting Codes [9] and the table compares the CPM size of Construction I and II for our algorithm for girth 8

K	5	6	7	8	9	10	11	12
I	13	18	21	25	30	35	41	47
II	9	11	13	15	17	19	21	23

Example 2: By using our algorithm, the exponent matrix **E(H)** for the case of $K = 9$ having girth 10 is obtained as

$$\mathbf{E(H)} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 6 & 11 & 19 & 29 & 44 & 60 & 81 \\ 0 & 3 & 7 & 12 & 20 & 30 & 45 & 61 & 82 \end{bmatrix}$$

TABLE III: Base matrix of $(3,5)$ for girth 10

Block-row index \ Block-column index	1	2	3	4	5
1	0	0	0	0	0
2	0	2	6	11	19
3	0	3	7	16	20

In our example 1, we consider $(3,9)$ exponent matrix. The CPM size obtained for $(3,9)$ matrix is 138. The base matrix is generated by finding the indices of the 2nd row by our algorithm as given in Table I for girth 8. For higher block-column length the $(3,9)$ sub matrix will act as a base matrix for $(3,10)$ matrix, we just need to calculate 10th index of 2nd row by our algorithm, i.e. by substituting numbers from 1 to CPM size *P* and regarding the 10th index of 3rd row, it can be achieved by just increasing the number obtained for 10th indices of the 2nd row. Thus after getting our base matrix, we just have to find one element that too the last element (i.e. K^{th} column index of the 3rd row) of $(3,K)$ matrix. Thus all preceding matrix act as a base matrix for the consecutive exponent matrix block-length. Hence, we are reducing computational complexity of constructing **H** matrix up to a good extent. In addition, this algorithm also works for higher order of girth, for an example 2, we have shown exponent matrix **E(H)** for girth 10, and base matrix is given in Table III. The comparison of CPM's size obtained in our method

with Voltage Graph based LDPC Tailbiting Codes [9] as shown in Table II and Table IV for girth 8 and 10 respectively.

TABLE IV: The construction I refer to method I of Voltage Graph Based LDPC Tailbiting Codes [9] and the table compares the CPM size of Construction I and II for our algorithm for girth 10

K	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
I	61	101	159	219	319	430	560	737
II	35	51	79	97	138	283	391	503

V. CONCLUSION

In this paper, we presented a simple less time consuming construction method for **H** matrix, having girth 8, 10 and 12. Our proposed method is based on upon first obtaining the base matrix, which leads to generate consecutive block-column sub matrices for the desired **E(H)** matrix size. We obtained a class of CPM-QC-LPDC codes as explained in our examples in section IV. The performance of proposed QC-LDPC codes is comparable in terms of CPM size requirements, which is significantly good as compared to the existing work. We also compared ours obtained CPM size for block-column length up to 12 for girth 8 and 10 with well-known existing work. The results are useful in construction of good binary regular QC-LDPC codes. Our codes can be deployed as a channel code in upcoming Wi-Fi/Wi-max standards for telecommunication purposes such as surveillance in vehicular communication, hybrid wireless networking and in wireless communication in autonomous decentralized sub-systems.

ACKNOWLEDGMENT

This work is part of research fund rewarded to Ambar Bajpai from the 90th year Chulalongkorn University scholarship. The work implemented within the allocated framework. In addition, thanks to Sreenidhi Institute of Science and technology (SNIST) for the financial assistance to present this work.

REFERENCES

[1] R. G. Gallagar, "Low-density parity-check codes," *IRE Trans. Inform. Theory,* vol. IT-8, pp. 21-28, Jan 1962.

[2] D. J. C. MacKay and R. M. Neal, "Near Shannon limit performance of low density parity check codes," *Electron. Lett.,* vol. 33, pp. 457-458, Jun. 1997.

[3] R. M. Tanner, "A recursive approach to low complexity codes," *IEEE Trans. Inform. Theory,* vol. 27, pp. 533-547, May 1981.

[4] M. P. C. Fossorier, "Quasi-cyclic low-density parity-check codes from circulant permutation matrices," *IEEE Trans. Inform. Theory,* vol. 50, pp. 1788-1793, Aug 2004.

[5] G. Malema, "Low density Parity-Check Codes: Construction and Implementation," Ph.D. dissertation, Faculty of Eng. Comp. and Math. Sci., Univ. of Adelaide, Australia, 2007.

[6] H. Fujita and K. Sakaniwa, "Some Classes of Quasi-Cyclic LDPC Codes: Properties and Efficient Encoding Methods," IEICE Trans. Fundam. Electr. Commun. Comp. sci., vol. 88, pp. 3627-3635, 2005.

[7] Y. Chen and K. Parhi, "Overlapped Message Passing For Quasi-Cyclic Low Density Parity Check Codes," IEEE Trans. Circuits syst., vol. 51, pp. 1106-1113, June 2004.

[8] G. Zhang and R. Sun, "Several Explicit Constructions for (3, L) QC-LDPC Codes with Girth at Least Eight," IEEE Communication Letters, vol. 17, no. 9, September 2013.

[9] I. E. Bocharova, F. Hug, R. Johannesson, B. D. Kudryashov, and R. V. Satyukov, "Searching for voltage graph-based LDPC tailbiting codes with large girth," IEEE Transactions on Information Theory, vol. 58(4), pp. 2265-2279. Apr. 2012.

[10] A. Bajpai, G. Srirutchataboon, P. Kovintavewat, and L. Wuttisittikulkij, "A New Construction Method for Large Girth Quasi-Cyclic LDPC Codes with Optimized Lower Bound using Chinese Remainder Theorem," Wireless Personal Communications, vol. 91(1), pp.369-381, 2016.

[11] M. Saadi, A. Bajpai, Y. Zhao, P. Sangwongngam, and L. Wuttisittikulkij L. "Design and Implementation of Secure and Reliable Communication using Optical Wireless Communication," Frequenz, vol. 68 pp. 501-509, Nov.-Dec. 2014.

[12] Y. Zhang and X. Da, "Construction of girth-eight QC-LDPC codes from arithmetic progression sequence with large column weight," Electronics Letters, vol. 51, no. 16, pp. 1257-1259, 2015.

[13] S. V. Ranganathan, D. Divsalar, and R. D. Wesel, "On the Girth of (3, L) Quasi-Cyclic LDPC Codes based on Complete Protographs" In IEEE International Symposium on Information Theory (ISIT), pp. 431-435, 14 Jun - 19 Jun. 2015.

[14] F. Hachiro and K. Sakaniwa, "Some classes of quasi-cyclic LDPC codes: Properties and efficient encoding method," IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences vol. 88, pp. 3627-3635, Dec. 2005.

[15] X. Jun, L. Chen, L. Zeng, L. Lan, and S. Lin. "Construction of low-density parity-check codes by superposition," IEEE transactions on communications vol. 53, pp. 243-251, Feb. 2005.