Quasi Cyclic Low Density Parity Check Code for High SNR Data Transfer

Mohammad Rakibul ISLAM [1], *Jinsang KIM* [2]

[1] Dept. of Electrical and Electronic Engineering, Islamic Univ. of Technology, Boardbazar, Gazipur-1704, Bangladesh
[2] Dept. of Electronics and Radio Engineering, Kyung Hee University, Suwon, 449-701, Korea

rakibultowhid@yahoo.com, jskim27@khu.ac.kr

Abstract. *An improved Quasi Cyclic Low Density Parity Check code (QC-LDPC) is proposed to reduce the complexity of the Low Density Parity Check code (LDPC) while obtaining the similar performance. The proposed QC-LDPC presents an improved construction at high SNR with circulant sub-matrices. The proposed construction yields a performance gain of about 1 dB at a 0.0003 bit error rate (BER) and it is tested on 4 different decoding algorithms. Proposed QC-LDPC is compared with the existing QC-LDPC and the simulation results show that the proposed approach outperforms the existing one at high SNR. Simulations are also performed varying the number of horizontal sub matrices and the results show that the parity check matrix with smaller horizontal concatenation shows better performance.*

Keywords

Low Density Parity Check codes, Quasi Cyclic Low Density Parity Check codes, SNR, bit error rate, circulant sub matrix, parity check matrix, generator matrix.

1. Introduction

Low-Density Parity-Check codes (LDPC) have been the subject of intense research lately because of their capacity-achieving performance and linear decoding complexity by using an iterative decoding algorithm, the so-called belief propagation or sum-product algorithm [1]. They were originally proposed in 1962 by Robert Gallager. In the late 90's LDPC codes were rediscovered by Mackay and Neal [3], [4] and also by Wiberg [5]. Current hardware speeds make them a very attractive option for wired and wireless systems. Gallager considered only regular LDPC, i.e., codes that are represented by a sparse parity-check matrix with a constant number of 'ones' (weight) in each column and in each row. Later it was shown that the performance of LDPC codes can be improved by using irregular LDPC codes, i.e., both non uniform weight per column and non uniform weight per row [6], [7]. The parity-check matrix of a code can be viewed as defining a bipartite graph [8] with "variable" vertices corresponding to the columns and "check" vertices corresponding to the rows. Each non-zero entry in the matrix corresponds to an edge connecting a variable to a check.

Quasi-Cyclic (QC)-LDPC has been proposed to reduce the complexity of the LDPC while obtaining the similar performance [9]. Recently, a Construction of Quasi-Cyclic LDPC Codes for AWGN and Binary Erasure Channels has been proposed by L. Lan [14]. Hardware implementations of decoders for Quasi-Cyclic LDPC Codes are being analyzed in some current research works [15], [16]. Some researchers are working on Quantum Quasi-Cyclic LDPC Codes in which, error detection and correction can be performed efficiently for quantum memory [17-19]. Girth of Quasi-Cyclic LDPC codes is an important issue and several current researches are going on this topic [20-23]. It has been shown that increasing the girth or average girth of a code increases its decoding performance. The girth also determines the number of iterations before a message propagates back to its original node. Performance of structured codes could therefore be improved by increasing their girths.

LDPC codes can be decoded by using different decoding algorithms. The same soft decision iterative decoding algorithms can be applied to the QC-LDPC codes. Decoding algorithms using Weighted Bit-Flipping decoding is a current research issue [25]. The significant benefit of the QC-LDPC lies in the code construction where the rows of the generator matrix are just cyclic shifts of the first row. These structured QC-LDPC codes having a relatively simple algebraic construction can be implemented with an inexpensive shift register generator and they greatly simplify the encoder design. The generator matrix *G* or Parity check matrix *H* of an LDPC has been generated randomly, which requires large power and storage space because of its larger size.

In this paper, we have constructed a QC-LDPC code which is suitable for small and medium block length applications. This new QC-LDPC code works better at high SNR and smaller horizontal concatenation. We have tested this code using bit flipping decoding, weighted bit flipping decoding, implementation-efficient reliability ratio based weighted bit flipping decoding and sum product (belief propagation) decoding. The proposed code is compared with an existing QC-LDPC code and our proposal shows better performance at high SNR.

The remainder of this paper is organized as follows: In Section 2, encoding and decoding using QC-LDPC is analyzed where different encoding and decoding techniques are shown. In Section 3, the proposed QC-LDPC technique is discussed. In Section 4, simulation results using BER analysis is shown and compared. Then Section 5 concludes this paper.

2. Encoding and Decoding using QC-LDPC Codes

In this section the encoding and decoding of QC-LDPC code will be discussed.

2.1 QC-LDPC Encoding

This paper discusses an algebraic construction for the regular and irregular QC-LDPC codes [9]. The regular LDPC codes have the same number of ones in every row and column. The irregular LDPC codes have a different number of ones in columns and rows. The QC-LDPC codes consist of horizontally concatenated circulant sub-matrices. Each circulant sub-matrix is a square matrix for which every row is the cyclic shift of the previous row, and the first row is obtained by the cyclic shift of the last row. In this way, every column of each circulant sub-matrix is automatically the cyclic shift of the previous column, and the first column is obtained by the cyclic shift of the last column. The ***H*** matrix of dimension $(m \times L_m)$ for the QC-LDPC can be written as

$$H = \left[H_1 \; H_2 \; H_3 \cdots H_L \right] \qquad (1)$$

where $\boldsymbol{H}_i$ is the *i*-th circulant sub-matrix of dimension $(m \times m)$, *i*=1, …, *L*. For the circulant matrices, the row weight and column weight are the same and fixed. Once the parity check matrix ***H*** is defined, the generator matrix is obtained. The matrices are created such that they should satisfy the constraint $\boldsymbol{GH}^T$=0. All the bits to be encoded are run through the generator matrix, and, therefore, all valid code words obey the property $\boldsymbol{CH}^T$=0 where *C* is the codeword.

The Quasi-Cyclic Generator matrix of rate *R*=(*L*-1)/*L* has the following structure:

$$G = \begin{bmatrix} P_2^T & I_m & 0 & 0 & \cdots & 0 \\ P_3^T & 0 & I_m & 0 & \cdots & 0 \\ P_4^T & 0 & 0 & I_m & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ P_L^T & 0 & 0 & 0 & 0 & I_m \end{bmatrix}$$

As one of the requirements is $\boldsymbol{GH}^T$=0, we can write

$$GH^T = \begin{bmatrix} P_2^T & I_m & 0 & 0 & \cdots & 0 \\ P_3^T & 0 & I_m & 0 & \cdots & 0 \\ P_4^T & 0 & 0 & I_m & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ P_L^T & 0 & 0 & 0 & 0 & I_m \end{bmatrix} \times \begin{bmatrix} H_1^T \\ H_2^T \\ H_3^T \\ \vdots \\ H_L^T \\ \end{bmatrix} = 0 . \qquad (2)$$

From the above relation, we can get $\boldsymbol{P}_i$=$\boldsymbol{H}_1^{-1}\boldsymbol{H}_i$ ***H***, where *i*=1, …, *L*. The inverse of a circulant matrix is a circulant, and the product of two circulant matrices is also a circulant matrix.

Therefore, the QC-LDPC of different rates (*L*-1)/*L* can be produced from the above-defined generator matrix ***G***. By using this construction, the quasi-cyclic nature of generator matrix is preserved. Since the generator matrix is quasi-cyclic, the first row of each circulant sub-matrix is stored, and successive rows can be generated by a shift register generator. This greatly simplifies the encoder design. It is crucial that the circulant sub-matrix $\boldsymbol{H}_1$ must be a nonsingular matrix. In order to maintain the nonsingularity of the circulant sub matrix $\boldsymbol{H}_1$, polynomial representation of its first row should not be a factor of x^m - 1. This point is illustrated with an example given below.

Let *m* = 15. So, we have

$$\begin{aligned} x^{15} - 1 = (x+1)(x^2+x+1)(x^4+x+1) \\ (x^4+x^3+1)(x^4+x^3+x^2+x+1) \end{aligned} \qquad (3)$$

If the weight of the circulant sub-matrix is 3, then the polynomial representation of its first row, or any one of the remaining cyclic shifts of its first row, should not be a factor of x^{15}- 1. For instance, if the first row of $\boldsymbol{H}_1$ is [1 0 0 0 0 1 0 0 0 1 0 0 0 0 0], the corresponding polynomial representation, is $(1 + x^5 + x^9)$. $\boldsymbol{H}_1$ is invertible, since neither its first row polynomial representation nor any of its cyclic shifts is a factor of x^{15}- 1. These QC LDPC codes should also avoid cycles of length 4 as defined by Mackay and Bresnan [4], [9]. For this to be possible, the separation between two nonzero positions in a row of length *m* in the circulant sub matrix $\boldsymbol{H}_i$ must satisfy the following equation [9]

$$ab = \min[(a-b) \bmod m, (b-a) \bmod m] \qquad (4)$$

where *a* and *b* are the nonzero positions in the first row of the circulant sub-matrix.

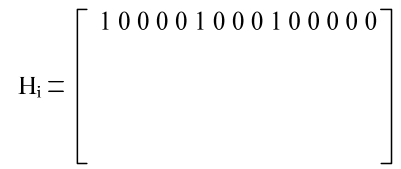

Fig. 1. Separation between the nonzero positions in a circular sub matrix.

This is demonstrated in Fig.1. In this example, $a = 1$, $b = 6$, $c = 10$ and $|ab| = 5$. In each circulant submatrix, the separation between any two nonzero positions should be calculated and placed in a set. The total number of elements in set t is given as

$$t = \sum_{i=1}^{L} \binom{w}{2} \tag{5}$$

where $\binom{w}{2} = \frac{n!}{(n-k)!k!}$, w is the weight of the circulant sub-matrix, and L is the total number of circulant sub-matrices in $\boldsymbol{H}$. For any two nonzero positions, the separation between them is calculated by using equation (5) and adding them to the set. If either two elements of the set are equal or an element of set is equal to $m/2$ (for m is even), then a cycle of length 4 exists in $\boldsymbol{H}$ [9].

2.2 QC-LDPC Decoding

The LDPC and QC-LDPC codes can use the same decoding algorithm using the appropriate parity check matrix $\boldsymbol{H}$. The LDPC is a linear block code defined by a parity check matrix $\boldsymbol{H}$ with m rows and n columns that mostly contain zeros and very few ones. LDPC codes can be represented by a Tanner graph that contains two different nodes: Bit nodes and Check nodes [4]. Each bit node corresponds to a code bit, and the check node corresponds to one parity check constraint on the bits defining a codeword. In other words, these bit nodes and the check nodes relate to the columns and rows of $\boldsymbol{H}$, respectively. An edge between a bit node and a check node exits if, and only if, the bits participate in the parity check equation represented by the check node.

2.2.1 Bit Flipping (BF) Decoding Algorithm

Bit flipping decoding of LDPC codes was devised by Gallager in the early 1960's [2], [11]. The same decoding can be used in our proposed QC-LDPC code. Let $\boldsymbol{H}$ be the parity check matrix of a QC-LDPC code with J rows and n columns. Let $\boldsymbol{h}_1$, $\boldsymbol{h}_2$, …, $\boldsymbol{h}_J$ denote the rows of $\boldsymbol{H}$, where $\boldsymbol{h}_J = (h_{J,0}, \boldsymbol{h}_{J,1}, \ldots, \boldsymbol{h}_{J,n-1})$.

For $\{1 \le j \le J\}$,

$$\mathbf{s} = \mathbf{z}.\mathbf{H}^{\mathrm{T}} \tag{6}$$

gives the syndrome of the received sequence $\boldsymbol{z}$, where the j th syndrome component s_j is given by the check sum

$$\begin{aligned} s_j &= \mathbf{z}.\mathbf{h_j} \\ &= \sum_{l=0}^{n-1} z_l h_{j,l} \end{aligned} \tag{7}$$

The received vector $\boldsymbol{z}$ is a codeword if and only if $s = 0$. When detectable errors occur during transmission, there will be parity check failures in the syndrome $\boldsymbol{s} = (s_1, s_2, \ldots, s_J)$, and some of the syndrome bits will be equal to 1.

BF decoding is based on the change of the number of parity failures in $\{\boldsymbol{zh}_j\colon 1 \le j \le J\}$ when a bit in the received sequence $\boldsymbol{z}$ is changed or flipped. The steps in BF decoding algorithm are shown as follows [12]

Step 1: Compute the parity check sums (syndrome bits) based on (6) and (7). If all the parity check sums are zero, stop the decoding.

Step 2: Find the number of failed parity check equations for each bit, denoted by f_j, $j = 0, 1, \ldots, n - 1$.

Step 3: Identify the set S of bits for which f_j is the largest.

Step 4: Flip the bits in set S

Step 5: Repeat steps 1 to 4 until all the parity check sums are zero or a preset maximum number of iterations is reached.

2.2.2 Weighted Bit Flipping (WBF) Decoding Algorithm

The simple hard decision one step BF decoding can be improved to achieve better performance by including some kind of reliability information of the received symbols in their decoding decisions. Consider the soft decision received sequence $\boldsymbol{y} = (y_1, y_2, \ldots, y_{n-1})$ at the output of the receiver matched filter. For an AWGN channel, a simple measure of the reliability of a received symbol y_l is its magnitude, $|y_l|$. The larger the magnitude $|y_l|$ is, the larger is the reliability of the hard decision digit z_l.

Consider a QC-LDPC code specified by a parity check matrix $\boldsymbol{H}$ with J rows, $\boldsymbol{h}_1$, $\boldsymbol{h}_2$, …, $\boldsymbol{h}_J$. For $\{0 \le l \le n - 1\}$ and $\{1 \le j \le J\}$ we define

$$\left|y_j\right|_{\min}^{(l)} = \{\min\{|y_i|\} : 0 \le i \le n-1, h_{j,i} = 1\} \tag{8}$$

and

$$E_l \overset{\Delta}{=} \sum\nolimits_{s_J^{(l)} \in S_l} (2s_j^{(l)} - 1)\left|y_j\right|_{\min}^{(l)} . \tag{9}$$

E_l is simply a weighted checksum that is orthogonal on the code bit position l. Steps in weighted BF decoding can be summarized as follows:

Step 1: Compute the parity check sums (syndrome bits) based on (6) and (7). If all the parity check sums are zero, stop the decoding.

Step 2: Compute E_l based on (9), for $\{0 \le l \le n - 1\}$.

Step 3: Find the bit position l for which E_l is largest.

Step 4: Flip the bit z_l.

Step 5: Repeat steps 1 to 4 until all the parity check sums are zero or a preset maximum number of iterations is reached.

2.2.3 Implementation-Efficient Reliability Ratio Based Weighted Bit Flipping (IRRWBF) Decoding Algorithm

Bit-flipping-based LDPC code decoding algorithms, such as weighted bit-flipping (WBF) and modified weighted bit-flipping (MWBF) algorithms [26] are considered as good trade-off between error-correcting performance and decoding complexity compared to belief-propagation-based (BP-based) decoding algorithms. BP-based algorithms yield excellent error-correcting capability, but their decoding complexity is also higher. Therefore, sometimes it is more practical to use bit-flipping decoding algorithms in energy-sensitive mobile devices. It was recently shown that the implementation efficient reliability ratio based bit flipping (IRRWBF) [13] algorithm performs best among existing bit flipping-based algorithms. To explain this decoding algorithm, four steps are needed: Initialization, Check node, Variable node, and Decision steps. Following equations describe this algorithm. For further details see the reference [13].

Step 1:

$$T_m = \sum\nolimits_{n\in N(m)} |r_n| \tag{10}$$

Step 2:

$$s_m = \sum_{n=1}^{N} z_m H_{mn} \tag{11}$$

Step 3:

$$E_n = \frac{1}{|r_m|} \sum\nolimits_{m\in M(n)} (2s_m - 1)T_m \tag{12}$$

Step 4:

Flip the bit for z_n for n = arg max$_{n'}E_{n'}$.

Symbols used in the previous steps can be summarized as follows: H_{mn} represents the m-th row and n-th column of parity-check matrix $\boldsymbol{H}$, r_n represents the n-th bit received from the channel, z_n represents Hard decision of r_n, $N(m)$ represents the set of variable nodes that participate in the m-th check node and $M(n)$ represents the set of check nodes in which the n-th variable node participates.

2.2.4 Standard Belief Propagation Algorithm

This section describes the decoding algorithm of LDPC codes based on a Belief Propagation or sum product decoding algorithm, according to Mackay and Neal [3], [4] and [10]. The decoder can be represented as a soft decision iterative algorithm called a message passing or belief propagation algorithm based on a tanner graph made up of check and bit nodes. Messages which are extrinsic information based on a-posteriori probability (APP) are passed along edges. A full iteration is defined as a cycle of message passing from bit nodes to check nodes and check nodes to bit nodes. The decoder is initialized by a soft-decision information received codeword. A full iteration is completed by a hard decision syndrome calculation. If a syndrome is detected, iteration continues. If no syndrome is detected, a valid codeword is found, and the decoder stops. The decoding problem is to find the most likely vector $\boldsymbol{x}$ such that $\boldsymbol{Hx} = 0$ with the likelihood of $\boldsymbol{x}$ given by

$$L(x) = \prod\nolimits_n f_n^{x_n} \tag{13}$$

where

$$f_n^1 = \frac{1}{1+\exp(-2ay_n/\sigma^2)}, \tag{14}$$

$$f_n^0 = 1 - f_n^1 \tag{15}$$

a = 1, σ^2 is the variance of the additive white Gaussian noise (AWGN), and y_n is the channel output at time n. The elements of $\boldsymbol{x}$ behave as the bit nodes and the rows of $\boldsymbol{H}$ as the check nodes. The set of bits n that participate in check m is denoted by $N(m) = \{n\colon H_{mn} = 1\}$. The set of checks in which bit n participates is defined as $M(n) = \{m\colon H_{mn} = 1\}$. A set $N(m)$ with bit n excluded is denoted by $N(m)\backslash n$. The decoding algorithm describes two parts, in which quantities q_{mn} and r_{mn} associated with each of the ‘ones’ in the $\boldsymbol{H}$ matrix are updated in an iterative fashion. The quantity $q^x{}_{mn}$ is the probability that bit n of $\boldsymbol{x}$ is $x = 0$ or 1, given the information obtained via checks other than the check m. The quantity $r^x{}_{mn}$ is the probability of check m being satisfied if bit n of $\boldsymbol{x}$ is considered fixed at x and the other bits have a separable distribution given by the probabilities $\{q_{mn'}\colon n' \in N(m)\backslash n\}$ [5]. The main steps involved in the Belief propagation algorithm are summarized as follows:

Initialization:

The variables $q^0{}_{mn}$ and $q^1{}_{mn}$ are initialized to the values $f^0{}_n$ and $f^1{}_n$ as $q^0{}_{mn} = f^0{}_n$ and $q^1{}_{mn} = f^1{}_n$.

Step 1: Compute

$$\delta q_{mn} = q^0_{mn} - q^1_{mn} \tag{16}$$

for each m and n. And for x = 0 and 1, compute

$$\delta r_{mn} = \prod\nolimits_{n'\in N(m)\backslash n} \delta q_{mn'}\,, \tag{17}$$

$$r^x_{mn} = (1/2)(1+(-1)^x \delta r_{mn})\,. \tag{18}$$

Step 2: For each m and n and for x = 0 and 1, update

$$q^x_{mn} = \alpha_{mn} f^x_n \prod\nolimits_{m'\in M(n)\backslash m} r^x_{m'n} \tag{19}$$

where α_{mn} is chosen such that $q^0{}_{mn} + q^1{}_{mn} = 1$. For each n and x = 0 and 1, we can update the pseudo posterior probabilities $q^0{}_n$ and $q^1{}_n$ as

$$q^x_n = \alpha_n f^x_n \prod\nolimits_{m\in M(n)} r^x_{mn} \tag{20}$$

where α_n is chosen such that $q^0{}_n + q^1{}_n = 1$.

Step 3: First, create $\hat{x} = [\hat{x}_n]$ such that $\hat{x}_n = 1$ if $q^1{}_n > 0.5$ and $\hat{x}_n = 0$ if $q^1_n \le 0.5$.

- If $H\hat{x}_n = 0$, then $\hat{x}$ is considered as a valid codeword, and the decoding algorithm comes to a halt.
- If the valid codeword is not found, then the algorithm repeats from step 1.
- The algorithm is terminated when it reaches the maximum number of iterations, and a failure is declared.

3. Proposed QC-LDPC Code

H matrix for proposed QC-LDPC code can be written as

$$H = \left[H_{L-1} \;\; \cdots \;\; H_2 \;\; H_1 \;\; H_2 \;\; \cdots \;\; H_L \right]. \qquad (21)$$

The Quasi-Cyclic Generator matrix of rate $R = ½$ has the following structure:

$$G = \begin{bmatrix} 0 & \cdots & 0 & P_2^T & I_m & 0 & \cdots & 0 \\ 0 & \cdots & P_3^T & 0 & 0 & I_m & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ P_L^T & \cdots & 0 & 0 & 0 & 0 & \cdots & I_m \end{bmatrix}.$$

As one of the requirements is $\boldsymbol{GH}^T$=0, we can write

$$\begin{bmatrix} 0 & \cdots & 0 & P_2^T & I_m & 0 & \cdots & 0 \\ 0 & \cdots & P_3^T & 0 & 0 & I_m & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ P_L^T & \cdots & 0 & 0 & 0 & 0 & \cdots & I_m \end{bmatrix} \times \begin{bmatrix} H_{L-1}^T \\ \vdots \\ H_2^T \\ H_1^T \\ H_2^T \\ \vdots \\ H_L^T \end{bmatrix} = 0. \qquad (22)$$

From the above equation, we can get several relations

$$\begin{aligned} P_2^T H_1^T &= H_2^T \\ P_3^T H_2^T &= H_3^T \\ &\vdots \\ P_L^T H_{L-1}^T &= H_L^T \end{aligned} \qquad (23)$$

The previous equation concludes $P_i = H_{i-1}^{-1} H_i$, where $i = 2,\dots, L$. The inverse of a circulant matrix is circulant, and the product of two circulant matrices is also a circulant matrix. By using this construction, the quasi-cyclic nature of generator matrix is preserved. Since the generator matrix is quasi-cyclic, the first row of each circulant sub-matrix is stored, and successive rows can be generated by a shift register generator.

4. Simulation Results

The simulation parameters used here are shown in Tab. 1. We experimented on two types of QC-LDPC codes: The existing one and the proposed one. At first we compare these two schemes in Fig. 2. We used Standard Belief Propagation algorithm (sum product decoding) for simulating these QC-LDPC codes. Codeword size, code rate, iterations, modulation and channel model are the same for both the proposed and existing QC-LDPC codes. Results show that the proposed QC-LDPC code outperforms the existing QC-LDPC at high SNR. The proposed construction yields a performance gain of about 1 dB at a 0.0001 bit error rate (BER).

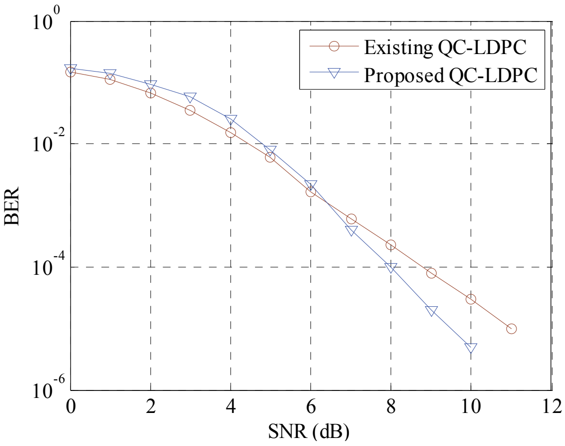

Fig. 2. Comparison between proposed and existing schemes using sum product decoding.

Again the proposed QC-LDPC code is tested on Bit Flipping (BF) [2], Weighted Bit Flipping (WBF) [2] and Implementation efficient Reliability Ratio based Weighted Bit Flipping (IRRWBF) decoding [13]. Fig. 3 shows the comparison between these decoding using our proposed QC-LDPC code. The result matches with the literature where IRRWBF is the most BER efficient decoding technique amongst these three. The effect of horizontal concatenation to develop matrix is then analyzed and shown in Fig. 4. The result shows that the increase in the number of horizontal concatenation increases BER. During simulation, we used the basis sub matrix dimension of 200×200. Using $L = 2$, the parity check matrix dimension becomes 200×400 where $L = 3$ makes the parity check matrix dimension 200×800. When we use the horizontal concatenation parameter $L = 6$, the parity check matrix dimension becomes 200×2000. So, the parity check matrix with larger rate shows better performance.

Simulation tool used	Matlab
Codeword Size	200-2000 bit
Code rate	½, ¼, 1/10
Iterations	10
Modulation	BPSK
Channel Model	AWGN
Maximum number of iterations	10
Number of rows in H matrix for Fig. 2 and Fig. 3	50
Number of rows in H matrix for Fig. 2 and Fig. 3	200
Number of 1's in each row per sub matrix	2
L for Fig. 2 and Fig. 3	3

Tab. 1. System parameters.

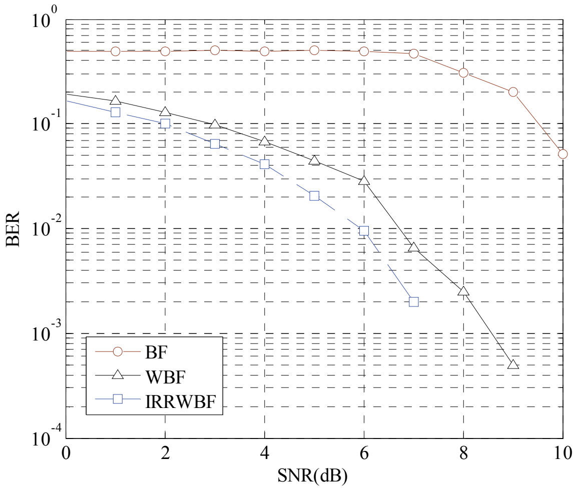

Fig. 3. Comparison between different decoders using the proposed scheme.

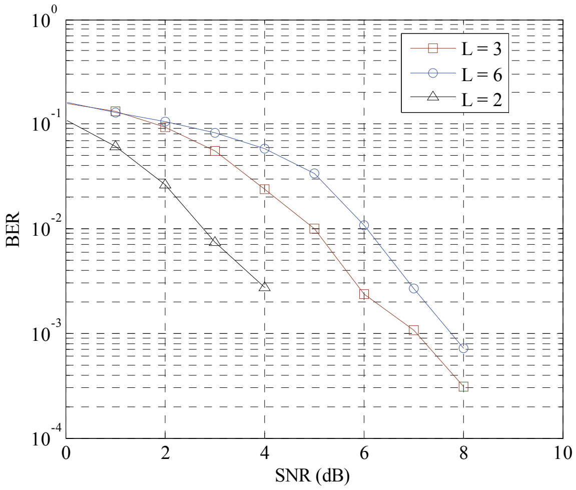

Fig. 4. Effect of horizontal concatenation in ***H*** matrix.

5. Conclusion

In this paper, an improved construction of circulant sub matrices based QC-LDPC code is proposed. The proposed QC-LDPC shows better performance than the existing QC-LDPC code at high SNR. The proposed construction yields a performance gain of about 1 dB at a 0.0003 bit error rate (BER). The proposed code is tested on four different decoding algorithms and compared. Simulation is also performed varying the number of horizontal sub matrix. The proposed structured QC-LDPC code has a relatively simple algebraic construction, which greatly simplifies the encoder design. This code construction can also be easily extended to the irregular QC-LDPC codes.

Acknowledgements

This work was supported by Korea Research Foundation (grant no.: 2009-0074806).

References

[1] KSCHISCHANG, F. R., FREY, B. J., LOELIGER, H. A. Factor graphs and sum-product algorithm. *IEEE Trans. Inform. Theory*, 2001, vol. 47, no. 2, pp. 498-519.

[2] GALLAGER, R. G. *Low-Density Parity-Check Codes*. Cambridge, MA: MIT Press, 1963.

[3] MACKAY, D. J. C., NEAL, R. M. Near Shannon limit performance of low density parity check codes. *IEE Electron Letter*, Aug. 1996, vol. 32, no. 18, pp. 1645-1646.

[4] MACKAY, D. J. C. Good error-correcting codes based on very sparse matrices. *IEEE Trans. Inform. Theory*, March 1999, vol. IT-45, no. 2, pp. 399-431.

[5] WIBERG, N. Codes and decoding on general graphs. *Linkoeping Studies in Science and Technology*, no. 440, 1996.

[6] RICHARDSON, T. J., SHOKROLLAHI, A., URBANKE, R. Design of capacity approaching low-density parity-check codes. *IEEE Trans. Inform. Theory*, Feb. 2001, vol. 47, pp. 619-637.

[7] LUBY, M., MITZENMACHER, M., SHOKROLLAHI, A., SPIELMAN, D. Analysis of low density codes and improved designs using irregular graphs. In *Proc. 30th Annu. ACM Symp. Theory of Computing*, 1998, pp. 249-258.

[8] KSCHISCHANG, F. R. Codes defined of graphs. I*EEE Commun. Mag.*, Aug. 2003, vol. 41, no. 8, pp. 118-125.

[9] BRESNAN, R. Novel code construction and decoding techniques for LDPC codes. *Master's thesis*, Dept. of Elec. Eng., UCC Cork, 2004.

[10] FOSSORIER, M., MIHALJEVIC, M., IMAI, H. Reduced complexity iterative decoding of Low-Density Parity Check Codes based on Belief Propagation. *IEEE. Trans. on Commun.*, May.1999, vol. 47, no. 5.

[11] GALLAGER, R. G. Low Density Parity Check Codes. *IRE Transactions on Information Theory*, IT-8: 21-28, January 1962.

[12] LIN, S., COSTELLO, D. J. *Error Control Coding*. Pearson Prentice Hall, 2004.

[13] LEE, C.H., WOLF, W. Implementation-efficient reliability ratio based weighted bit-flipping decoding for LDPC codes. I*EE Electronics Letters*, June 2005, vol. 41 no. 13.

[14] LAN, L., ZENG, L., TAI, Y. Y., CHEN, L., LIN, S., GHAFFAR, K. A. Construction of Quasi-Cyclic LDPC Codes for AWGN and binary erasure channels: A finite field approach. *IEEE Transactions on Information Theory*, July 2007, vol. 53, no. 7.

[15] ARABACI, M., DJORDJEVIC, I. An alternative FPGA implementation of decoders for quasi-cyclic LDPC codes. In *TELFOR*, 2008.

[16] SUN, Y., KARKOOTI, M., CAVALLARO, J. R. VLSI Decoder architecture for high throughput, variable block-size and multi-rate LDPC codes. In *ISCAS* 2007.

[17] HAGIWARA, M., IMAI, H. Quantum quasi-cyclic LDPC codes. In *IEEE International Symposium on Information Theory*, June 2007.

[18] HSIEH, M., BRUN, T., DEVETAK, I. *Quantum Quasi-Cyclic Low-Density Parity-Check Codes.* 2008. Available at http://arxiv.org/abs/0803.0100v1

[19] ZHAO, S., ZHENG, B., WANG, W. Construction of quantum Low Density Parity Check Code based on quasi-cyclic sparse sequence. In *International Conference on Communications and Networking in China, 2008.*

[20] WU, X., YOU, X., ZHAO, C. A necessary and sufficient condition for determining the girth of Quasi-Cyclic LDPC Codes. *IEEE*

Transactions on Communications, June 2008, vol. 56, no. 6, pp. 854-857.

[21] MALEMA, G., LIEBELT, M. Quasi-cyclic LDPC codes of column-weight two using a search algorithm. *EURASIP Journal on Advances in Signal Processing*, 2007. doi:10.1155/2007/45768.

[22] WANG, Y., YEDIDIA, J. S., DRAPER, D S. C. Construction of high-girth QC-LDPC codes. In *International Symposium on Turbo Codes and Related Topics,* 2008.

[23] KIM, S., NO, J. S., CHUNG, H., SHIN, D. J. Cycle analysis and construction of protographs for QC LDPC codes with girth larger than 12. In *IEEE International Symposium on Information Theory*, June 2007.

[24] KIM, J., RAMAMOORTHY, A., MCLAUGHLIN, S. W. Design of Efficiently-Encodable Rate-Compatible Irregular LDPC Codes. ICC, 2006.

[25] WU, X., LING, C., JIANG, M., XU, E., ZHAO, C., YOU, X. Towards understanding weighted bit-flipping decoding. In *IEEE International Symposium on Information Theory*, June 2007.

[26] ZHANG, J., FOSSORIER, M. A modified weighted bit-flipping decoding of low-density parity-check codes. *IEEE Communication Letter*, 2004, pp. 165–167.

About Authors

Mohammad Rakibul ISLAM received the B.Sc.Engg. and M.Sc.Engg. degree in Electrical and Electronic Engineering from Bangladesh University of Engineering and Technology (BUET), Bangladesh in 1998 and 2004 respectively. He also received MBA degree in Marketing from the Institute of Business Administration (IBA) under the University of Dhaka. He completed his Ph.D. in Electronics and Radio Engineering from Kyung Hee University, South Korea in 2010. He joined the Department of Electrical and Electronic Engineering, Islamic University of Technology (IUT) as a faculty in 1999 and is currently serving as an assistant professor. His research interests include cooperative technique for wireless sensor networks, LDPC and QC-LDPC codes, secrecy capacity and other wireless applications.

Jinsang KIM received the B.S. and M.S. degrees in Electronic Engineering from Kyung Hee University, Seoul, Korea in 1985 and 1987, respectively, and the Ph.D. degree in Electric and Computer Engineering from Colorado State University, Fort Collins, in 2000. From 1990 to 2001, he was with Korea Telecom R&D Center, Seoul, Korea, as a Member of Technical Staff, engaged in research on telecommunication circuits and systems protocols, as well as Internet multimedia services. In 2001, he joined the School of Electronics and Information, Kyung Hee University, where he is currently associate professor. His research interests include multimedia signal processing and VLSI system design for arithmetic units, and multimedia and wireless applications.