Ldpc Code Construction using Randomly Permutated Copies of Parity Check Matrix[#]

بناء أكواد قليلة الكثافة باستخدام تباديل لمصفوفة فحص التكافؤ

Amr Lulu & Ammar Abu-Hudrouss*

عمرو لولو، وعمار أبو هدروس

Electrical Engineering Department, Islamic University, Gaza, Palestine

*Corresponding author: ahdrouss@iugaza.edu.ps

Received: (8/10/2017), Accepted: (24/1/2018)

Abstract

A construction technique is proposed for low-density parity check (LDPC) codes. It uses a base parity check matrix designed from a random or constructed construction method as Gallager or Quasi-Cyclic LDPC (QC-LDPC) codes in sequence to get codes with multiple lengths and same rate of the base matrix. This is done by using a seed matrix with row and column weights of one, distributed randomly and can be addressed by a number in the base matrix, this method reduces the memory usage and reduce the probability of failing to construct a parity matrix by random approaches. Also, by the use of circulant identity seed matrices, QC-LDPC codes can be obtained keeping the same performance.

Keywords: Low Density Parity Check Codes, seeds matrix, Quasi cyclic codes, Gallager codes.

ملخص

يتطرق هذا البحث إلى دراسة وعرض طرق بناء الأكواد ذات مصفوفات الفحص قليلة الكثافة بطرق مختلفة مثل الاكواد المبنية بطريقه عشوائية التركيب (مثل أكواد جالاجر)، والاكواد المبنية بطريقه منظمة ذات نسق ونموذج معروف (مثل الأكواد ذات الطبيعة الدائرية). ذلك بالإضافة أن

This research is based on the Master's thesis for the student, which was discussed at the Islamic University in Gaza on 13/2/2012.

هذه الورقة تقدم طريقة جديدة لبناء الأكواد ذات مصفوفات الفحص قليلة الكثافة والتي تستخدم مصفوفات سابقة البناء كأساس للبناء وللحصول على أكواد ذات أطوال مضاعفة من نفس طول الكود الأساسي وبنفس المعدل. يتم ذلك عن طريق استخدام مصفوفة وحدة موزعة بشكل عشوائي, ويمكن تمييزها برقم عند إدراجها في المصفوفه الأساسية. هذا الأسلوب يقلل من الذاكرة المستخدمة ويقلل من احتمال حدوث فشل في البناء بالطريقة العشوائية. نتائج المحاكاة ايضاً تدل على أن البناء المقترح يزيد من متوسط طول الدوائر في رسومات تانر الدالة على المصفوفة، ويقلل من عدد الدورات ذات الطول الرباعي في المصفوفة الاساسية.

الكلمات المفتاحية: بناء الأكواد ذات مصفوفات الفحص قليلة الكثافة، المصفوفة النواة، أكواد جالجار، الأكواد شبه الدائرية.

Introduction

Low density parity check codes are a special type of error correcting codes that is known for their good decoding performance and high throughput. Low density parity check codes were first introduced by Robert Gallager in early 60's (Gallager, 1962). His work was ignored for decades because of its high computational complexity for hardware implementation in that time.

LDPC codes are decoded using a subclass of message passing algorithms (Richardson, T. J. & Urbanke, 2001) introduced in Gallager's work named the belief propagation decoding algorithm. Its strength is in the inherent parallelism of the message passing and the iterative decoding behavior that is done by updating bit probabilities.

LDPC codes are designed starting from the parity check matrix, where two sets of separated nodes called check and variable nodes are connected to points in the other set based on some regulations and restrictions. The separation of sets allows parallel decoding computations. In contrary, the decoding operations of turbo codes which are the most competitors to LDPC codes, depends on each other in blocks or windows (Berrou, Glavieux, & Thitimajshima, 1993) which results in serial computations. LDPC codes have simple graphical representation based on Tanner graph (Berrou *et al*., 1993; Tanner, 2002) that leads to accurate analysis of performance, also it helps optimizing the designs of regular and irregular constructions.

Another good property of LDPC codes is having good distance properties, which is one of the main challenges in designing of LDPC codes (Zhang, Yu, Chen, & Li, 2017).

The construction of LDPC codes is categorized mainly into two main categories: Random constructions and structured constructions. The type of construction is determined by the connections between check nodes and variable nodes in Tanner graph. Random codes have better performance compared to structured codes in case of long codes (Chung, Forney, Richardson, & Urbanke, 2001).

A disadvantage of random LDPC codes appears when wanted to be implemented it in practical system, the long length needs large memory to be stored and used in decoding and encoding, which affects the computational efficiency of the code. Also, in decoding using sum-product algorithm, they do not converge as fast as in structured LDPC codes.

The main advantages in using structure are the increase in adaptability, and a reduction in cost in terms of complexity, memory usage, and transmission latency.

One important type of LDPC codes is Quasi-cyclic codes, these codes are easy to implement because of their block and cyclic interconnections and at the same time can perform as well as random-like LDPC codes in terms of bit-error probability, block error probability and error floor (L Chen, 2004; Lei Chen, Xu, Djurdjevic, & Lin, 2004).

A method to generate LDPC codes of multiple lengths starting from previously constructed codes is introduced. The method solves the problem of randomness of distribution of ones in randomly constructed LDPC codes as in MacKay's and Gallager's and keeps the performance the same. The idea of construction starts from the same idea of QC-LDPC codes. In constructing QC-LDPC codes, we first construct the model matrix $\check{H}_C$ where its entries refer to the number of shifts applied to an identity matrix placed at the same address in the parity check matrix (Lulu, 2012). The design is introduced in the following sections.

Construction of Ldpc Code Using Identity Seed Matrix

In the following design, we consider having a base parity check matrix H_b of dimensions $N \times m$ represented by:

$$H_b = \begin{bmatrix} c_{0,0} & c_{0,1} & \cdots & c_{0,N-1} \\ c_{1,0} & c_{1,1} & \cdots & c_{1,N-1} \\ \vdots & & \ddots & \vdots \\ c_{m-1,0} & c_{m-1,1} & \cdots & c_{m-1,N-1} \end{bmatrix}$$

H_b must satisfy the constraints of LDPC design, where $g \geq 6$, $\lambda = 0$ or 1. Entries of H_b are binary data denoted by $c_{i,j}$ where $c_{i,j} \in (0, 1)$, $1 \leq j \leq m$ and $1 \leq i \leq N$.

Definition 1: An identity sub-matrix I_p called *identity seed* and all-zeros square sub-matrix O_p with dimensions $p \times p$ is defined, where p is an integer and $p \geq 2$. The construction is done by constructing an all-zeros parity check matrix H of dimensions $Np \times mp$ divided into a number of (Nm) sub-matrices of dimensions $p \times p$, and then the base matrix H_b is used as a model matrix to construct H by replacing each 1 and 0 in H_b by I_p and O_p in H respectively. The resulted H has the same row and column weights of H_b.

In the following, we show an arbitrary example of the construction.

Example: Consider a (2, 3)-regular code of parity check matrix of dimensions (6 × 4) obtained from Euclidean geometry construction and represented by the parity check matrix.

$$H_b = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix} \qquad (1)$$

This matrix defines the regular-LDPC code which has a length of N = 6 with a rate R = 1/3. Let us assume that we want to construct a regular code with length N = 12 of the same rate, thus we define I_2 and O_2 by:

$$I_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \qquad O_2 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

Now we replace each 1 by I_2 and each 0 by O_2, and the resulted H matrix is:

$$H = \begin{bmatrix} I_2 & I_2 & I_2 & O_2 & O_2 & O_2 \\ I_2 & O_2 & O_2 & I_2 & I_2 & O_2 \\ O_2 & I_2 & O_2 & I_2 & O_2 & I_2 \\ O_2 & O_2 & I_2 & O_2 & I_2 & I_2 \end{bmatrix},$$

which is equivalent to:

$$H = \left[\begin{array}{cc:cc:cc:cc:cc:cc} 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hdashline 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ \hdashline 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ \hdashline 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \end{array}\right]$$

$$H_b = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix} \qquad H = \left[\begin{array}{cc:cc:cc:cc:cc:cc} 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hdashline 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ \hdashline 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ \hdashline 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \end{array}\right]$$

Figure (1): A graphical representation of cycles of girth g = 6 resulted from the construction in definition 1.

***Theorem 1*:** A parity check matrix constructed by definition 1 has a girth equal to the girth of the base matrix H_b.

***Proof*:** The girth is calculated by counting the straight edges that connect between '1's starting and ending at the same 1 entry in the parity check matrix (see Fig. 1.).Since the construction expands the parity check matrix with identity and all zeros matrices, then the construction does not change horizontal and vertical positions of the ones, nor adds ones between existed '1's in the matrix, where the length of the edges just increases by multiples equal to p. The insertion of identity seed results in new cycles of the same length which keeps the girth equal to the same girth in the base matrix.

***Theorem 2*:** The rank of H is equal to $\text{Rank}(H_b) \times p$.

Proof: The rank of matrix H_b is defined by the maximum number of independent rows in the matrix, the construction of H in definition 1 keeps the positions of horizontal and vertical '1's fixed without adding '1's between the original '1's, which preserves the relation of dependency or independency between rows fixed. Since row reduction method operations between rows will result in the same rows in H_b but with zeros from the expansion added to the row. The shifted expanded copies of the original rows are independent of original rows which results in $(p - 1)$ copies of original rows independent from the original and have the same relations with their rows, which leads to the relation:

$$\text{Rank}(H) = p \bullet \text{Rank}(H_b) \quad (6)$$

Corollary 1: The rate R of H is equal to the rate R_b of H_b.

The rate of H_b is given by $R_b = (N - \text{Rank}(H_b))/N$, the rate of H is given by $R = (pN - p\bullet\text{Rank}(H_b))/(pN) = p(N - \text{Rank}(H_b)/(pN) = (N - \text{Rank}(H_b))/N$. Thus $R = R_b$.

Sparsity of H

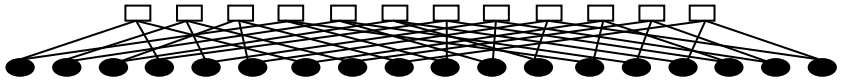
The proposed construction keeps the degree of variable nodes and check nodes fixed. The insertion of the identity matrices leads to sparser H matrix. Suppose that in H_b the total number of 1's is given by e, and the total number of entries is s, then the density D_b of H_b is e/s. The number of

1's in H matrix is then equal to (pe) and the total entries is now equal to p^2s, the density of H is $pe/(p^2s) = e/(ps)$. So the relation between D and D_b is given by:

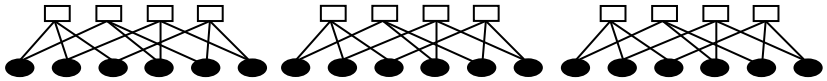
$$D = D_b/p \tag{7}$$

$$H = \begin{bmatrix}
1&0&0&1&0&0&1&0&0&0&0&0&0&0&0&0&0&0\\
0&1&0&0&1&0&0&1&0&0&0&0&0&0&0&0&0&0\\
0&0&1&0&0&1&0&0&1&0&0&0&0&0&0&0&0&0\\
1&0&0&0&0&0&0&0&0&1&0&0&1&0&0&0&0&0\\
0&1&0&0&0&0&0&0&0&0&1&0&0&1&0&0&0&0\\
0&0&1&0&0&0&0&0&0&0&0&1&0&0&1&0&0&0\\
0&0&0&1&0&0&0&0&0&1&0&0&0&0&0&1&0&0\\
0&0&0&0&1&0&0&0&0&0&1&0&0&0&0&0&1&0\\
0&0&0&0&0&1&0&0&0&0&0&1&0&0&0&0&0&1\\
0&0&0&0&0&0&1&0&0&0&0&0&1&0&0&1&0&0\\
0&0&0&0&0&0&0&1&0&0&0&0&0&1&0&0&1&0\\
0&0&0&0&0&0&0&0&1&0&0&0&0&0&1&0&0&1
\end{bmatrix}$$

(a)



(b)



(c)

Figure (2): (a) Construction of H matrix with an identity seed of size 3 × 3. (b) Corresponding Tanner graph of the H matrix. (c) The same Tanner graph in part (b) after rearrangement of nodes shows the separation of edges between the three copies.

Graphical Perspective

The construction using identity seed is similar to protograph construction describe in (T. Richardson & Urbanke, 2002; Thorpe, 2003). The corresponding Tanner graph results in p separated copies of the base matrix H_b. The copies are not connected by any edge between their check and variable nodes as shown in Fig. 2. The advantage of this construction is

decreasing the time of decoding since each copy can be implemented individually and semi parallel with other copies as in (Lee et al., 2004). The problem in this type of construction is in the decoding process when a data bit is received at a variable node that belongs to one of the copies, the outgoing messages of that bit are sent only among its parent copy, and at the same time it receives messages only from the check nodes from the same parent copy. Thus, it cannot benefit from the information sent from bits in the other copies with strong probabilities of receiving a correct bit. Also the girth average of variable nodes is not changed where increasing the number of columns and rows with decreasing the density of ones in the matrix is expected to increase the girth average of its Tanner graph, where increasing the girth average enhances the decoding performance (McGowan & Williamson, 2003).

Construction of Ldpc Code using Randomly Permutated Identity Seed Matrix

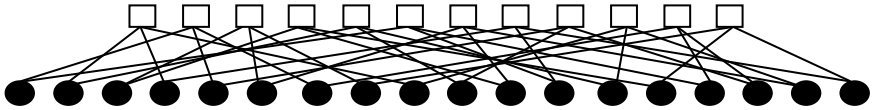
To solve the problem of independency between copies we introduce a modification in definition 2 that interchanges the connections of edges between all variable nodes and check nodes all over the copies and increases the girth average of the code.

***Definition 2*:** The identity seed matrix I_p is replaced by a randomly permutated identity matrix of dimensions $p \times p$ with regular column and row weights of 1, in other words each row contains 1 in a unique random column. And each 0 in the base matrix is replaced as in definition 1.

The graphical correspondence of permutations in definition 2 is that for a variable node V_1 connected to check node C_1 by edge E. The edges of the p copies of V_1 are permutated across the p copies of C_1 where each copy of V_1 is connected with only one copy of C_1. For an example see Fig. 3.

$$H = \begin{bmatrix}
0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1
\end{bmatrix}$$

(a)



(b)

Figure (3): (a) H matrix of LDPC code of length $N = 18$, $R = 1/3$, with $j = 2$ and $k = 3$ constructed by definition 2. (b) the corresponding Tanner graph of H.

The proposed construction can be represented by a model matrix since each 1 in the original matrix is replaced by a randomly permutated identity matrix $I_p(r)$ called *random seed* selected from a space of $p!$ different matrices, where $0 \leq r \leq p! - 1$. Each permutated identity matrix is assigned a number r which indicates which $I_p(r)$ to be replaced. Fig. 4 shows an example of a 6 random seeds generated by choosing $p = 3$, each assigned by a number, and Fig. 5 shows the model matrix for H in Fig. 3.

$$I_3(0)=\begin{bmatrix}1&0&0\\0&1&0\\0&0&1\end{bmatrix},\ I_3(1)=\begin{bmatrix}0&1&0\\0&0&1\\1&0&0\end{bmatrix},\ I_3(2)=\begin{bmatrix}0&0&1\\1&0&0\\0&1&0\end{bmatrix}$$

$$I_3(3)=\begin{bmatrix}1&0&0\\0&0&1\\0&1&0\end{bmatrix},\ I_3(4)=\begin{bmatrix}0&1&0\\1&0&0\\0&0&1\end{bmatrix},\ I_3(5)=\begin{bmatrix}0&0&1\\0&1&0\\1&0&0\end{bmatrix}$$

Figure (4): The 6 possible randomly permutated matrices generated from $p = 3$.

$$\breve{H}_c=\begin{bmatrix}I_3(4)&I_3(0)&I_3(2)&O_3&O_3&O_3\\I_3(5)&O_3&O_3&I_3(4)&I_3(4)&O_3\\O_3&I_3(2)&O_3&I_3(1)&O_3&I_3(2)\\O_3&O_3&I_3(2)&O_3&I_3(3)&I_3(4)\end{bmatrix}$$

Figure (5): The model matrix $\breve{H}_c$ for H in Fig. 3a.

The proposed construction differs from protograph LDPC code construction is the ability of storing the addresses of permutated matrices and calling the addresses by a simple address generation mechanism from the model matrix. This helps in reducing the memory for storing the $m \times N$ matrix. Another advantage of construction from definition 2 is when constructing starting by a bad seed matrix generated with a random method containing 4 cycles, the construction can reduce the number of removed bit resulted by any method of removing loops algorithms (McGowan & Williamson, 2003) in order to enhance the decoding performance of the code as illustrated in the following theorem.

***Theorem 3*:** The Girth of code constructed by definition 3 is greater than or equal to the girth of the base matrix.

***Proof*:** The girth of the proposed code is guaranteed to be greater than or equal to the girth of the base matrix, this belongs to the fact that the insertion of random seeds in the base matrix, adds cycles of the same length of the existed cycles in the base matrix distributed by the randomness of the positions of 1's in each seed matrix which allows the path of old cycles to pass over the possible position of old ones to construct larger

cycles. So, the random distribution of ones in each seed results in enlarging the length of cycles in corresponding Tanner graph and keeps the girth of the base matrix which is the reason why the proposed construction can reduce number of 4 cycles in bad base matrices.

Construction of Ldpc Code using Randomly Permutated Identity Seed Matrix

The proposed construction in definition 2 helps in reducing the memory to store and constructed random entries of the LDPC H matrix by a factor of $1/p^2$. An example is when a construction of a LDPC code with of length $N = 3000$ and rate $R = 1/2$ is done using a random seed with $p = 100$, it results in a model matrix with dimensions of 15 by 30 instead of 1500 by 3000. The construction can be achieved by a simple random generation of numbers that refers to a predefined random seed. It also preserves the performance of same length code constructed by the same method the base matrix is constructed by. A drawback of the proposed construction in definition 2 is when choosing a large seed matrix in order to get a long length code from a small base matrix, the number of all possible random seeds of dimensions $p \times p$ is growing to exceed the number of 1's in the base matrix. And each 1 in the base matrix has the possibility to be replaced with a different seed matrix, which leads to storing a number of seeds that are probably greater than the number of ones in the sparse base matrix, and that reduces the efficiency of the code design. Recalling the base matrix represented in example 1, if p is chosen to be 4, the number of possible matrices to be generated is $4! = 24$, but the number of matrices to be used is upper-bounded by the total number of ones in the base matrix which is 12 in this case, so to overcome this problem we can limit the number of seeds by the total number of 1's in the base matrix in order not to store unused seeds. Another problem in this type of construction appears when using a large p seeds with a relatively large base matrix with many 1's entries, in this case the random generation will result in too many different seeds to be stored, and the advantage of reducing the memory space of H matrix will be violated by the large number of stored seeds.

So, it is recommended to use of codes from definition 2 only when the dimensions of base matrix are small with large p seeds, or when the base matrix is large using small p seeds.

In order to overcome the restrictions in constructing codes by definition 2, we suggest degreasing the number of choices to be equal to p instead of $p!$ by taking the seed matrix I_p with its $p - 1$ circulant matrices to be the only available choices, the insertion of only p circulant matrices turns the code into a QC-LDPC code similar to (McGowan & Williamson, 2003). The randomness in choosing the seed to be inserted keeps the girth of the constructed code greater than or equal to the original girth of the base matrix as in theorem 2. The quasi cyclic property of the code allows linear encoding with shift registers (Peterson & Weldon, 1972)[pp. 256–261]. The performance of the QC-LDPC codes generated by this method is found to be similar in BER performance of codes from definition 2.

Simulation and Results

In this section, we represent the performance results for the proposed LDPC codes by comparing the BER of codes designed by the proposed construction with codes from random constructions such as Gallager codes and with classic random QC-LDPC codes. In additive white Gaussian noise (AWGN) channel, we use Implementation-efficient Reliability Ratio Based Weighted Bit-Flipping (IRRWBF) [56] to decode LDPC codes. And all simulations use maximum iteration number of 80. In the first example we compare the performance of a LDPC Gallager code (504, 3, 6) with a LDPC code constructed using a base matrix H_b constructed by a LDPC Gallager code with (168, 3 ,6) and a random seed with $p = 3$. The resulted H matrix is of a LDPC code of length $N = 504$ of the same rate.

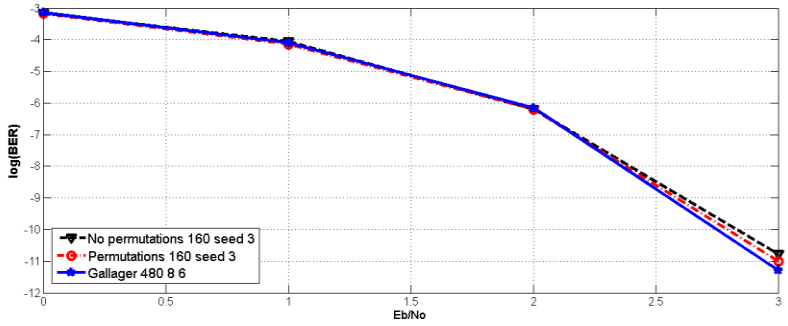


Figure (6): BER performance of (504, 3, 6) proposed code designed with base of Gallager H with random seed of $p = 3$, and with an identity seed of $p = 3$, compared to Gallager code with length $N = 504$.

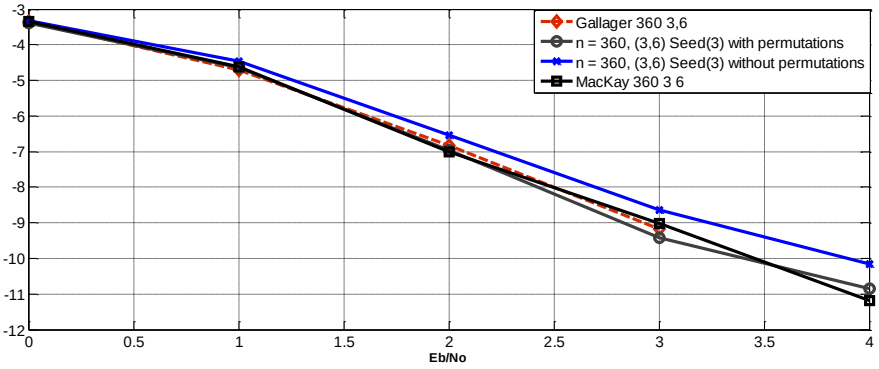


Figure (7): Performance of proposed code with $N = 360$ and $j = 3$, $k = 6$, once with a seed of random $I_3(r)$, with MacKay's and Gallager's code of the same length.

It can be seen in Fig. 6 and Fig. 7 that the BER performance of the proposed codes is very close to Gallager's and MacKay's codes. Since both of constructions have similar BER performance, the proposed code still has an advantage over random codes such as Gallager's and MacKay's is that it reduces the memory usage by having addresses of each random seed. The construction with identity seeds shows degradation of the performance of the extended code. The second example represents a comparison between the proposed code with rate $R = 1/2$, $j = 3$, $k = 6$ constructed with classic random QC-LDPC base matrix of $m = 9$ with a seed $I_3(r)$ to give a block length $N = 162$, and with a classic random QC-LDPC of the same length with circulant identity matrices with $m = 27$. The results in Fig. 8 show that proposed codes constructed by a QC-LDPC base, outperform the original code constructed with the same QC-LDPC matrix. The proposed code also outperforms the classic QC-LDPC codes with different rates and block lengths as shown in Fig. 9.

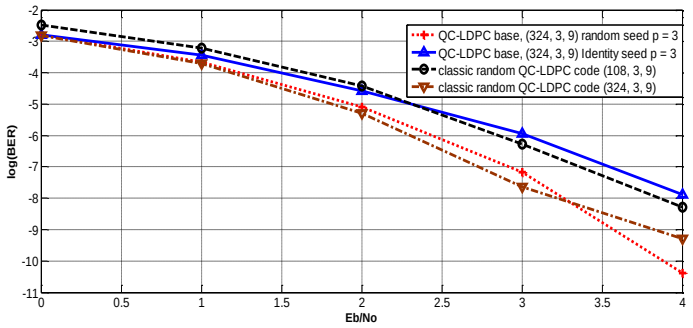


Figure (8): BER performance of proposed codes with rate $R = 2/3$ and a classic random QC-LDPC code with the same rate and block length $N = 324$.

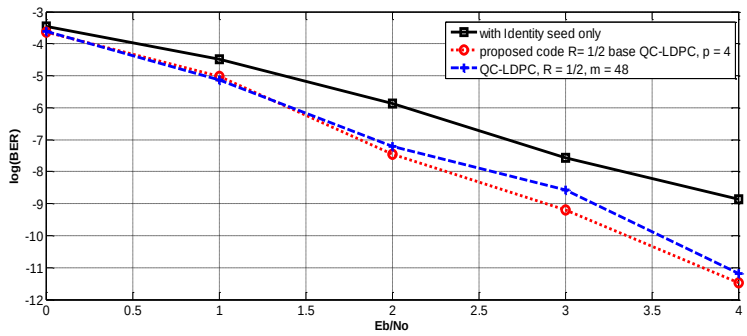


Figure (9): BER performance of proposed codes with rate R =1/2 and a classic random QC-LDPC code with the same rate and block length N = 288.

Fig. 10 shows the performance of proposed QC-LDPC code compared to code generated by definition 2. Both constructions appear to have the same BER performance, taking in mind the advantage of QC-LDPC codes over other proposed design in encoding process and in decreasing the number of stored seed matrices.

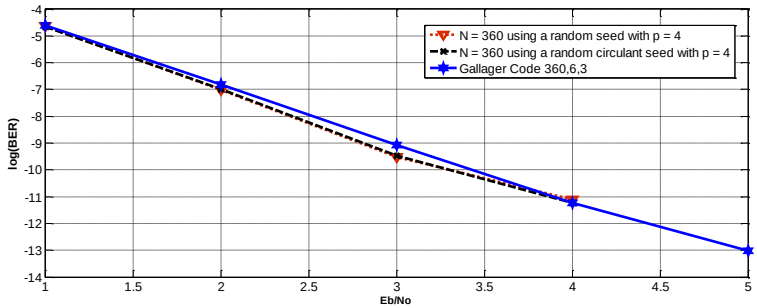


Figure (10): BER performance of proposed QC-LDPC code with N = 360 and code with random seeds, compared to Gallager code.

Conclusions

The main issue between random construction methods and structured methods is the trade-off between the need of high memory storage for randomly constructed codes and the easy implementation of structured codes, taking in account the outperforming of random codes in large block lengths over structured codes.

Thus, we need to construct LDPC codes that have good BER performance, and are also easy to be implemented in hardware. In this work we introduce a LDPC code construction method that reduces the memory usage for storing H matrix by a factor of $1/p^2$ and performs similar to fully random codes as Gallager's and MacKay's in small and medium lengths. Also, experimental results show that the proposed code outperforms classic random QC-LDPC codes. Another advantage is that it can reduce the number of ones to be removed in order to get rid of 4 cycles which degrade the decoding performance. We constructed LDPC codes with various lengths and rates and we show that the proposed method works well in designing block type LDPC codes.

References

- Berrou, C., Glavieux, A., & Thitimajshima, P. (1993). *Near Shannon limit error-correcting coding and decoding: Turbo-codes. 1*. Paper presented at the Communications, 1993. ICC'93 Geneva. Technical Program, Conference Record, IEEE International Conference on.
- Chen, L. (2004). An algebraic method for constructing quasi-cyclic LDPC codes. *Proc. 2004 ISITA, Parma, Italy, October 10-13*.
- Chen, L., Xu, J., Djurdjevic, I., & Lin, S. (2004). Near-Shannon-limit quasi-cyclic low-density parity-check codes. *IEEE Transactions on Communications, 52*(7), 1038-1042.
- Chung, S.-Y., Forney, G. D., Richardson, T. J., & Urbanke, R. (2001). On the design of low-density parity-check codes within 0.0045 dB of the Shannon limit. *IEEE Communications letters, 5*(2), 58-60.

- Gallager, R. (1962). Low-density parity-check codes. *IRE Transactions on information theory, 8*(1), 21-28.
- Lee, J. K.-S., Lee, B., Thorpe, J., Andrews, K., Dolinar, S., & Hamkins, J. (2004). A scalable architecture of a structured LDPC decoder.
- Lulu, A. Y. (2012). *CONSTRUCTION OF LDPC CODES USING RANDOMLY PERMUTATED COPIES OF PARITY CHECK MATRIX*. Islamic University of Gaza.
- McGowan, J. A., & Williamson, R. C. (2003). *Removing loops from LDPC codes.* Paper presented at the Proceedings of the Australian Communications Theory Workshop.
- Peterson, W. W., & Weldon, E. J. (1972). *Error-correcting codes*: MIT press.
- Richardson, T., & Urbanke, R. (2002). *Multi-edge type LDPC codes.* Paper presented at the Workshop honoring Prof. Bob McEliece on his 60th birthday, California Institute of Technology, Pasadena, California.
- Richardson, T. J., & Urbanke, R. L. (2001). The capacity of low-density parity-check codes under message-passing decoding. *IEEE Transactions on information theory, 47*(2), 599-618.
- Tanner, R. M. (2002). On graph constructions for LDPC codes by quasi-cyclic extension. In *Information, Coding and Mathematics* (pp. 209-220): Springer.
- Thorpe, J. (2003). Low-density parity-check (LDPC) codes constructed from protographs. *IPN progress report, 42*(154), 42-154.
- Zhang, J.-b., Yu, Y., Chen, Z.-y., & Li, Z.-Z. (2017). A new method for constructing parity check matrix of QC-LDPC codes based on shortening RS codes for wireless sensor networks. *Sustainable Computing: Informatics and Systems*.