

# A Class of Quantum LDPC Codes Derived from *Latin Squares* and Combinatorial Design

Salah A. Aly

Department of Computer Science,

Texas A&M University, College Station, TX 77843-3112, USA

Email: salah@cs.tamu.edu

December 12, 2007

Time: 1:49 CST

**Abstract**—In this paper we construct a class of regular Low Density Parity Check (LDPC) codes derived from Latin squares. The parity check matrices of these codes are constructed by permuting shift-orthogonal Latin squares of order  $n$  in block-rows and block-columns. I show that the constructed LDPC codes are self-orthogonal and their minimum and stopping distances are bounded. This helps us to construct a family of quantum LDPC block codes. Consequently, we demonstrate that these constructed codes have good error correction capabilities and can be decoded using iterative decoding algorithms similar to their classical counterpart.

## I. INTRODUCTION

Low Density Parity Check (LDPC) codes are a capacity approaching (*Shannon limit*) class of codes that first appeared in a seminal work by Gallager [6]. LDPC codes were rediscovered by Tanner [22], in which he showed the interpretation graphical view of these codes (*codes over graphs*). Iterative decoding of LDPC and turbo codes highlighted these codes as important classes of codes (modern coding theory) for communication and storage channels. Furthermore, they have been used intensively in many applications [4], [12]. Rather than, BCH and Reed-Solomon cyclic codes, LDPC codes are often historically constructed by a computer search. Also, their encoding complexity is high in comparison to other codes. However, LDPC codes have high performance and better error correction capabilities because they have iterative decoding algorithms [12], [13], [21], [23].

Quantum information is sensitive to noise and needs error correction strategies. Quantum block and convolutional codes are means to protect quantum information. Quantum block LDPC codes have been introduced using a computer search by MacKay in [14]. He constructed sparse graph quantum codes from classical LDPC codes. Recently, Camara *et al.* derived quantum LDPC codes in an analytical method [3]. Hagiwara and Imai constructed quasi-cyclic (QC) LDPC codes and derived a family of quantum QC LDPC codes from a nested pair of classical codes [7]. In our work we establish sufficient conditions for the parity check matrix  $\mathbf{H}$  of a LDPC code to be self-orthogonal.

In this paper, a new class of quantum LDPC codes based on our construction of LDPC codes is proposed. We derive regular LDPC codes from elements of shift-orthogonal (Latin squares)

and algebraic combinatorics [1]. Quantum LDPC block codes constructed in this note have some advantages; (a) quantum block codes constructed from LDPC are good codes as shown by MacKay *et al.* [14], (b) LDPC codes are capacity achieving codes and have high rates, (c) the constructed codes can be decoded using standard iterative decoding algorithms.

The constructed codes have cycles with length 4 to guarantee the self-orthogonality condition. Moreover, we show that the performance of these codes is reasonable and can be improved by reducing the number of 4-cycles in the parity check matrix. We also note that these codes have high rates. This is due to the fact that we try to have less 4-cycle, dimension of the parity check matrix is reduced, i.e.  $R \geq 1 - k/n$ . Finally, performance of our constructed codes can be improved by shortening and puncturing the parity check matrices of these codes to reduce the number of cycles with length 4.

*Notation:* We will refer to a row of matrices (block) as a block-row and a regular row of elements through out some matrices as a row. This is also applied to a block-column.

## II. CLASSICAL AND QUANTUM LDPC CODES

In this section we introduce quantum and classical LDPC codes. Our goal is to make this note self-contained as possible.

### A. Quantum LDPC Codes

Quantum LDPC first appeared in a paper by Mackay *et al.* in [14]. He showed that good quantum block codes can be constructed from classical codes with low-weight codewords. So, it is not necessary to start with a good classical code that has high minimum distance.

**Proposition 1:** A  $(\rho, \lambda, n)$ -LDPC code is a dual-containing code if it has a parity check matrix  $\mathbf{H}$  over  $\mathbf{F}_2$  such that

- i) Every row has fixed weight  $\lambda$  and every column has fixed weight  $\rho$ .
- ii) Every pair of rows in  $\mathbf{H}$  has an even overlap, and every row has even weight, meaning every pair of rows is *multiplicity even*.

MacKay used the random construction of LDPC codes to derive quantum codes. Recently, Camara *et al.* showed quantum convolutional LDPC codes using analysis methods [3].

They presented a class of quantum codes that can be decoded using iterative algorithms. We now can define quantum LDPC codes using the row and column weights.

*Definition 2:* A quantum LDPC code is  $2^k$  dimensional subspace of the complex space  $2^n$  and can be defined by a stabilizer matrix  $S_{stab}$  that has a pair  $(\rho, \lambda)$  where  $\rho$  is the number of non-zero error operators per column and  $\lambda$  is the number of non-zero error operators per row.

For the binary case, the error operator can be an element in the Pouli group generated by the matrices  $\{I, X, Z, Y = iXZ\}$ .

### B. Classical LDPC Codes

LDPC codes, whether they are block or convolutional, have better encoding and decoding algorithms in comparison to other codes. In fact this class of codes can be encoded using shift register circuits, see for example [14], [20], [21], [23] and the recent survey paper [13]. LDPC codes that have an algebraic structure are superior because i) they perform well in terms of bit and block error probabilities, and ii) they are easy to encode and decode.

We pursue our construction by defining some terms. We can define a QC-LDPC code over the binary field  $\mathbf{F}_2$  as the null-space of a matrix  $\mathbf{H}$  of sparse circulants of equal size. The matrix  $\mathbf{H}$  with parameters  $(\rho, \lambda)$  has the following properties:

- 1)  $\rho$  is the weight of a column  $c_i$ ,
- 2)  $\lambda$  is the weight of a row  $r_i$ .

From this definition, the minimum distance of the QC-LDPC defined by the null-space of  $\mathbf{H}$  is at least  $\rho+1$ . This is because we can add at least  $\rho+1$  columns in the parity check matrix  $\mathbf{H}$  to get the zero column.

**Definitions.** Let  $\mathbf{F}_q$  denote a finite field of characteristic  $p$  with  $q$  elements. Recall that the set  $\mathbf{F}_q^* = \mathbf{F}_q \setminus \{0\}$  of nonzero field elements is a multiplicative cyclic group of order  $q-1$ . A generator of this cyclic group is called a primitive element of the finite field  $\mathbf{F}_q$ .

Let  $n$  be a positive integer such that  $n = q^m - 1$ , where  $m = \text{ord}_n(q)$  is the multiplicative order of  $q$  modulo  $n$ . Let  $\alpha$  denote a fixed primitive element of  $\mathbf{F}_{q^m}$ . Define a map  $\mathbf{z}$  from  $\mathbf{F}_{q^m}^*$  to  $\mathbf{F}_2^n$  such that all entries of  $\mathbf{z}(\alpha^i)$  are equal to 0 except at position  $i$ , where it is equal to 1. For example,  $\mathbf{z}(\alpha^2) = (0, 1, 0, \dots, 0)$ . We call  $\mathbf{z}(\alpha^k)$  the location (or characteristic) vector of  $\alpha^k$ . We can define the location vector  $\mathbf{z}(\alpha^{i+j+1})$  as the right cyclic shift of the location vector  $\mathbf{z}(\alpha^{i+j})$ , for  $0 \leq j \leq n-1$ , and the power is taken module  $n$ .

*Definition 3:* We can define a map  $A$  that associates to an element  $\mathbf{F}_{q^m}^*$  a circulant matrix in  $\mathbf{F}_2^{n \times n}$  by

$$A(\alpha^i) = \begin{pmatrix} \mathbf{z}(\alpha^i) \\ \mathbf{z}(\alpha^{i+1}) \\ \vdots \\ \mathbf{z}(\alpha^{i+n-1}) \end{pmatrix}. \quad (1)$$

By construction,  $A(\alpha^k)$  contains a 1 in every row and column.

For instance,  $A(\alpha^1)$  is the identity matrix of size  $n \times n$ , and  $A(\alpha^2)$  is the shift matrix

$$A(\alpha^2) = \begin{pmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & 0 & 0 & \dots & 0 \end{pmatrix}. \quad (2)$$

We will use the map  $A$  to associate to a parity check matrix  $H = (h_{ij})$  in  $(\mathbf{F}_{q^m}^*)$  the (larger and binary) parity check matrix  $\mathbf{H} = (A(h_{ij}))$  in  $\mathbf{F}_2^{n \times n}$ . The matrices  $A(h_{ij})$ 's are  $n \times n$  circulant permutation matrices based on some primitive elements  $h_{ij}$  as shown in Definition 3.

Now, we give two definitions to measure the performance of the decoding algorithms of LDPC codes: girth of a Tanner graph and stopping sets. The minimum stopping set is analogous to the minimum Hamming distance of linear block codes.

*Definition 4 (Girth of a Tanner graph):* The girth  $g$  of the Tanner graph is a length of its minimum cycle.

The stopping set of a Tanner graph is a subset of the variable nodes  $V$  such that its neighboring check nodes in  $L$  are connected to at least two nodes in this subset as shown in the following definition. The stopping distance is the size of the smallest stopping set and it determines the number of correctable erasures by an iterative decoding algorithm, see for example [5], [18], [19].

*Definition 5 (Stopping sets):* The set  $S \subseteq C$  is called the stopping set of a graph  $G = (V, C, E)$  if the degree of each vertex in  $\Gamma(S)$  in the induced graph  $G_S$  on  $S \cup \Gamma(S)$  is at least two, where  $\Gamma(S)$  is the set of neighbors of  $S$  in  $V$ .

Let  $s$  be the size of the smallest stopping set, i.e.,  $s$  is the stopping distance (number). We can also define the stopping distance from  $\mathbf{H}$  directly as follows [19].

*Definition 6 (Stopping distance):* The stopping distance of the parity check matrix  $\mathbf{H}$  is defined as the largest integer  $s(\mathbf{H})$  such that every set of  $(s(\mathbf{H})-1)$  or less columns of  $\mathbf{H}$  contains at least one row of weight one.

The stopping ratio  $\sigma$  of the Tanner graph is defined by  $s/n$ . The minimum Hamming distance is a property of the code to measure its performance for maximum-likelihood (ML) decoding, while the stopping distance is a property of the parity check matrix  $\mathbf{H}$  or the Tanner graph  $G$  of a specific code. Hence it varies for different choices of  $\mathbf{H}$  for the same code  $\mathcal{C}$ . The stopping distance  $s(\mathbf{H})$  gives a lower bound of the minimum distance of the code  $\mathcal{C}$  defined by a the low density parity check matrix  $\mathbf{H}$ . Hence,

$$s(\mathbf{H}) \leq d_{min}. \quad (3)$$

It has been shown that finding the stopping sets with minimum cardinality is an NP-hard problem since the minimum-set vertex covering problem can be reduced to it [10]. One can also define the trapping sets for AWGN and BSC communication channels.

### III. CONSTRUCTING LDPC CODES FROM LATIN SQUARES

In this section we construct self-orthogonal algebraic Low Density Parity Check (LDPC) codes derived from Latin squares. The class that we show has a quasi-cyclic (QC) structure and hence is called QC-LDPC codes. There have been some constructions of LDPC and QC LDPC based on Latin squares such as the construction in [24] based on mutually orthogonal and cyclic Latin squares. Also, in [11], [17] the authors designed LDPC codes based on idempotent and symmetric Latin squares. These constructions are beneficial because they have girth of at least 6 and the codes are regular and irregular with arbitrary rates. In addition, the authors computed the stopping sets to measure performance of LDPC codes over the binary erasure channel.

#### A. Latin Square

*Definition 7:* Let  $S$  be a set of elements  $\{a_1, a_2, \dots, a_n\}$ . An  $n \times n$  matrix  $L = (a_{ij})$  is called a Latin square of order  $n$  if each row and column contains each element of  $S$  exactly once.

Let  $\mathbf{Z}_n = \{1, 2, \dots, n\}$  be the set of integers of size  $n$ . We will consider the set  $S$  to be  $\mathbf{Z}_n$ . Hence, a Latin square of order  $n$  is an integer square matrix of size  $n \times n$  such that each element  $i \in \mathbf{Z}_n$  appears only once in every row and column. Clearly many Latin squares can be defined over the same alphabet, but the exact number is not known for large  $n$ . Latin squares have been used in many applications and there are various methods to construct them. In addition, there is a connection between Latin squares and permutation groups. In other words, one can look at a permutation group of order  $n$  as a Latin square of order  $n$ . We can define the *main* and *isotopy* classes of Latin squares as follows, see [8], [11], [15].

*Definition 8:* Let  $L$  and  $L'$  be two Latin squares of order  $n$ .

- i) If the square  $L'$  can be obtain from  $L$  under row, column and symbol permutations, then  $L$  is isotopy to  $L'$ . The set of all Latin squares isomorphic to  $L$  is called *isotropy* class.
- ii) The *main* class of  $L$  is given by the set of all squares which are isomorphic to some conjugate of  $L$ . *Paratopic* squares are a set of squares which belong to the same *main* class.
- iii) We call a Latin square  $L$  of order  $n$  reduced if  $(1, 2, 3, \dots, n)$  appears in the first row and column.
- iv) For  $1 \leq k \leq n$ , a Latin rectangle is an array of size  $k \times n$  such that every element appears once in a row and may or may not appear in a column. Clearly, Latin squares are special cases of Latin rectangles where  $k = n$ , see [16].

Let  $R_n$  be the total number of reduced Latin squares, the total number of Latin squares of order  $n$  is given by

$$L_n = n!(n-1)!R_n.$$

We can also study properties of some classes of Latin squares.

*Definition 9:* Let  $L$  and  $L'$  be two Latin squares of order  $n$

- i)  $L$  is shift-orthogonal to  $L'$  if the cell  $(i, j)$  in  $L$  is different from the cell  $(i, j)$  in  $L'$  for all  $2 \leq i \leq n$  and  $1 \leq j \leq n$ .
- ii)  $L$  and  $L'$  are *half-reduced shift-orthogonal Latin* squares if they are shift-orthogonal except in the first row.
- iii)  $L$  and  $L'$  are *orthogonal Latin* squares if all the  $n^2$  pairs  $(a_{ij}, b_{ij})$  are different where  $a_{ij} \in L$  and  $b_{ij} \in L'$ .
- iv) There are at most  $n-1$  *mutually orthogonal Latin* squares of order  $n$ . Therefore, the set  $L_1, L_2, \dots, L_n$  is *mutually orthogonal* if  $L_i$  and  $L_j$  are orthogonal for  $1 \leq i < j \leq n$ .

One main difference between orthogonal and shift-orthogonal Latin squares of order  $n$  is that every orthogonal Latin square is a shift-orthogonal Latin square but the converse is not true. For example, there are no orthogonal Latin squares of order 6, however, one can also always construct shift-orthogonal Latin squares of any order.

As an example, two orthogonal Latin squares of order  $n = 4$  are given by

$$L_1 = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 1 & 4 & 3 \\ 3 & 4 & 1 & 2 \\ 4 & 3 & 2 & 1 \end{pmatrix}, L_2 = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 3 & 4 & 1 & 2 \\ 4 & 3 & 2 & 1 \\ 2 & 1 & 4 & 3 \end{pmatrix}. \quad (4)$$

One way to obtain all shift-orthogonal Latin squares is by fixing the first row and permute all other rows by one to obtain a new Latin square matrix. Therefore, we have  $n-1$  permuted *shift-orthogonal Latin* squares.

Let  $n = p$  a prime, there are  $n-1$  pairwise orthogonal Latin squares of order  $n$ . They can be formed using the matrix

$$L_j = \begin{pmatrix} 0 & 1 & \dots & n-1 \\ j & j+1 & \dots & n-1+j \\ 2j & 2j+1 & \dots & n-1+2j \\ \vdots & \vdots & \vdots & \vdots \\ (n-1)j & (n-1)j+1 & \dots & n-1-j \end{pmatrix}, \quad (5)$$

for  $j = 1, 2, \dots, n-1$  and all elements in  $L_j$  are reduced mod  $p$ . Clearly,  $L_i$  and  $L_j$  are shift-orthogonal Latin squares if  $j \neq i$ . This can be generalized in case of  $n$  be a prime power  $q = p^m$ .

Latin squares have been used to construct efficient LDPC codes, see [11], [17].  $L$  is symmetric if the cells  $(i, j)$  and  $(j, i)$  for  $1 \leq i < j \leq n$  contain the same symbol. We define a special class of Latin squares called Cayley Latin squares where the elements  $\{1, \dots, n\}$  form a cyclic group of order  $n$ .

*Theorem 10:* The Latin square  $L$  derived from the Cayley table of a group  $G$  is atomic if and only if  $G$  is a cyclic group of prime order.

*Proof:* See [25]. ■

Clearly, the transpose of a (shift-orthogonal) Latin square is also a (shift-orthogonal) Latin square. We can also define the minimum distance between two rows in a Latin square as the number of nonzero elements in the difference among these two rows. We can see that the Hamming distance between any two rows of an  $n \times n$  Latin square is  $n$ .

## B. A Class of LDPC

Let  $q$  be a prime integer. We construct a class of LDPC codes based on shift-orthogonal Latin squares with elements from  $\mathbf{Z}_n$ , where  $n = q$ .

Let  $\alpha^i$  be an element in corresponding to  $i$  in  $\mathbf{Z}_n$  for  $1 \leq i \leq n$ .  $\mathbf{Z}_n \simeq S = \{\alpha^1, \alpha^2, \dots, \alpha^{n-1}\} \cup \{\alpha^n = 0\}$ . We can form the matrix  $G$  of size  $n \times n$  as a result of the multiplicative group  $\mathbf{Z}/n\mathbf{Z}$

$$G = \begin{pmatrix} g_1 \\ g_2 \\ \vdots \\ g_n \end{pmatrix} = ( h_1 \ h_2 \ \dots \ h_n ) \\ = \begin{pmatrix} \alpha^1 & \alpha^2 & \alpha^3 & \dots & \alpha^n \\ \alpha^2 & \alpha^4 & \alpha^6 & \dots & \alpha^{n-1} \\ \vdots & \vdots & \vdots & \vdots & \\ \alpha^n & \alpha^{n-1} & \alpha^{n-2} & \dots & \alpha^1 \end{pmatrix}, \quad (6)$$

where  $g_i$  is the  $i$ th row in  $G$  and  $h_j$  is the  $j$ th column in  $G$ . The matrix  $G$  has the following structure:

- i) any two distinct rows differ in all positions.
- ii) any two distinct columns differ in any positions.
- iii) all elements of  $S$  are presented in a row (column).

This matrix  $G$  is equivalent to the Latin square of order  $n$ . Let  $\mathbf{G}$  be an  $n \times n(n-1)$  binary matrix of  $G$  where every element  $\alpha^i$  written by the characteristic vector  $\mathbf{z}(\alpha^i)$  of length  $n-1$  and  $\mathbf{z}(0)$  is zero vector of length  $n-1$ . We know that there are  $n-1$  shift-orthogonal Latin squares of order  $n$ , we call them  $B_1, B_2, \dots, B_{n-1}$  where  $G = B_1$ . Furthermore  $\mathbf{B}_i$  is the binary matrix corresponding to  $B_i$ . We form the matrix  $B$  by permuting rows of the matrix  $G$  in a certain order. So, the matrix  $B_j$  is a permutation of the matrix  $B_i$  under row permutation where the first row is fixed.

$$B = ( B_1 \ B_2 \ \dots \ B_{n-1} ). \quad (7)$$

We have formed an  $n \times (n-1)n$  matrix  $B$  where every row in  $G$  is extended horizontally  $(n-1)$  times. All matrices  $B_1, B_2, \dots, B_{n-1}$  have the same first row and they represent *shift-orthogonal half-reduced Latin squares* of order  $n$ . In addition  $\mathbf{B}$  is an  $n \times n(n-1)^2$  binary matrix derived from  $B$ .

*Corollary 11:* Any two rows in the matrix  $B$  differ in all positions. I.e.,  $\mathbf{B}$  is a self-orthogonal matrix.

*Proof:* This is a direct consequence of our construction. Any two rows of the matrix  $B_j$  satisfies this condition. Therefore, any two rows in all matrices  $B_j$ 's are orthogonal. Also, for any length  $n$ , the multiplication  $n(n-1)^2$  is even. Therefore, the inner product of two rows in  $\mathbf{B}$  always vanishes. ■

We can also see that the Hamming distance between any two rows of the matrix  $B$  is  $n(n-1)$ . This is because any two rows in the sub-matrix  $B_i$  have Hamming distance equal to  $n$ .

We can also extend every matrix  $B_j$  in  $B$  vertically to form the matrix  $H_j$  as follows. Let  $\alpha B_j$  represents the cyclic shift

of every row in  $B_j$  i.e. multiplying every row in  $B_j$  by  $\alpha \pmod n$ .

Let

$$B_j = ( h_{j,1} \ h_{j,2} \ \dots \ h_{j,n} ),$$

where  $h_{j,i}$  is a column vector.

$$H_j = \begin{pmatrix} B_j \\ \alpha B_j \\ \vdots \\ \alpha^{\rho-1} B_j \end{pmatrix} \quad (8)$$

$$= \begin{pmatrix} B_1 & B_2 & \dots & B_{n-1} \\ \alpha B_1 & \alpha B_2 & \dots & \alpha B_{n-1} \\ \vdots & \vdots & \vdots & \vdots \\ \alpha^{\rho-1} B_1 & \alpha^{\rho-1} B_2 & \dots & \alpha^{\rho-1} B_{n-1} \end{pmatrix} \quad (9)$$

Now the matrix  $H_j$  has size  $(\rho)n \times n$ . Therefore we formed a  $(\rho)n \times (n-1)n$  matrix  $H$ .

$$H = ( H_1 \ H_2 \ H_3 \ \dots \ H_{(n-1)} ). \quad (10)$$

The matrix  $H_j$  has the following properties:

- i) Every  $n$  components of every column are distinct and they form all the  $n$  nonzero elements of  $S$ .
- ii) any two columns differ in every position.
- iii) Any two rows have even number of elements in common.

*Lemma 12:* For  $1 \leq i, j \leq \rho n$ ,  $i \neq j$ , any two rows  $g_i$  and  $g_j$  in  $H$  have no common symbol from  $S$  or they have at most one symbol in common.

*Proof:* The proof is straightforward from the construction of the matrix  $H$  and permutations of its rows and columns. It can be stated in two cases. a) if  $g_i$  and  $g_j$  lie in the same block-row, then clearly they do not have any element in common. This is from the Latin square property. b) if  $g_i$  and  $g_j$  lie in two different blocks  $\alpha^i B$  and  $\alpha^j B$ . The first row in  $\alpha^i B$  is different from the first row  $\alpha^j B$ . Also, the first row in  $\alpha^i B$  has one common element with the rows of  $\alpha^j B$  except for its first row. The rest of rows of  $\alpha^i B$  has the same property with the elements from  $\alpha^j B$ . ■

We now can replace every entry in  $H$  by its location vector  $\mathbf{z}$  to obtain a  $(\rho)n \times n(n-1)^2$  matrix  $\mathbf{H}$ . We construct the  $\rho \times (n-1)n$  matrix  $\mathbf{H}$  of  $n \times n$  submatrices over  $\mathbf{F}_2$ .

$$\mathbf{H} = ( \mathbf{H}_1 \ \mathbf{H}_2 \ \dots \ \mathbf{H}_{n-1} ) \\ = \begin{pmatrix} \mathbf{H}_{1,1} & \mathbf{H}_{2,1} & \dots & \mathbf{H}_{n-1,1} \\ \mathbf{H}_{1,2} & \mathbf{H}_{2,2} & \dots & \mathbf{H}_{n-1,2} \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{H}_{1,\rho} & \mathbf{H}_{2,\rho} & \dots & \mathbf{H}_{n-1,\rho} \end{pmatrix} \quad (11)$$

and the matrices  $\mathbf{H}'_{i,j}$ 's are  $n \times n(n-1)$  circulant permutation matrices of Latin squares.

By this construction we built an  $\rho n \times n(n-1)^2$  matrix  $\mathbf{H}$  over  $\mathbf{F}_2$ , where we replace  $\alpha^i$  by 1 at position  $i$  in the vector  $\mathbf{z}(\alpha^i)$ . The previous steps are summarized in algorithm 1. We notice that the row weight of  $\mathbf{H}$  is  $(n-1)n$  and the column weight is  $\rho$ .

- 1: Input: A Latin square of order  $n$ ,
- 2: Output: A parity check matrix  $\mathbf{H}$  of size  $\rho n \times (n-1)n^2$ .
- 3: Construct the matrix  $G$  as the multiplication group of  $\mathbf{Z}_n$ , Latin square of order  $n$ .
- 4: **for**  $j = 1$  to  $(n-1)$  **do**
- 5:   construct the sub-matrices  $B_1, B_2, \dots, B_{n-1}$  as *shift-orthogonal* Latin squares.
- 6: **end for**
- 7: **for**  $j = 1$  to  $n-1$  **do**
- 8:   for each sub-matrix  $B_j$  construct the column submatrices  $H_{ji}$  for  $1 \leq i \leq \rho$ .
- 9: **end for**
- 10: Form the matrix  $H$ .
- 11: Convert every element in  $H$  to a locator vector  $\mathbf{z}$  to form the matrix  $\mathbf{H}$ .

Fig. 1. Constructing LDPC codes based on elements of a finite field (Latin Square)

*Lemma 13:* The rank of parity check matrix  $\mathbf{H}$  is given by  $(\rho n) - (\rho - 1)$ .

*Proof:* The proof of this lemma can be shown by mathematical induction for  $1, 2, \dots, \delta - 1 \leq n$ . We know that every row-block is linearly independent.

- i) Case i. Let  $\delta = 1$ , the statement is true since every row-block has only 1 in every column, the first  $n$  columns represent the identity matrix.
- ii) Case ii-1. Assume the statement is true for  $\delta - 1$ . In this case, the matrix  $\mathbf{G}$  has a full rank given by  $(\delta)n - (\delta - 2)$ . So, we have

$$\mathbf{G} = \begin{pmatrix} h_{11} & h_{12} & h_{13} & \dots & \dots & h_{1n} \\ 0 & h_{22} & h_{23} & \dots & \dots & h_{2n} \\ 0 & 0 & h_{33} & \dots & \dots & h_{3n} \\ 0 & 0 & 0 & \vdots & \vdots & h_{in} \\ 0 & 0 & \dots & h_{(\delta-1)(\delta-1)} & \dots & h_{(\delta-1)n} \end{pmatrix}.$$

The elements  $h'_{ii}$ s have 1's in the diagonal and zeros everywhere using simple Gauss elimination method.

- iii) Case iii-1. We can form the sub-matrix  $\mathbf{H}_2$  of size  $(\delta)n \times (\delta)n$  by adding one row-block to the matrix  $\mathbf{G}$ . The last row-block is generated by the latin square  $L_\delta$

Now, all  $n - 1$  rows of the last row-block are linearly independent and can not be generated from the previous  $\delta - 1$  row-blocks. Now, in order to obtain the last row-block to be zero at positions  $h_{(\delta)1}, h_{(\delta)2}, \dots, h_{(\delta)(\delta-1)}$ , we can add the element  $h_{jj}$  to the element  $h_{(\delta)j}$ . In addition, the last row (row indexed by  $(\delta)n$ ) of row-block  $\delta$  can be generated by adding all elements of the first row-block to to the first  $n - 1$  rows of the last row-block.

$$\mathbf{G} = \begin{pmatrix} h_{11} & h_{12} & h_{13} & \dots & \dots & h_{1n} \\ 0 & h_{22} & h_{23} & \dots & \dots & h_{2n} \\ 0 & 0 & h_{33} & \dots & \dots & h_{3n} \\ 0 & 0 & 0 & \vdots & \vdots & h_{in} \\ 0 & 0 & \dots & h_{(\delta-1)(\delta-1)} & \dots & h_{(\delta-1)n} \end{pmatrix}.$$

Therefore, the matrix  $\mathbf{G}$  has rank of  $(\delta - 1)n - (\delta - 2) + n - 1 = (\delta)n - (\delta - 1)$ . We notice that the matrix  $\mathbf{H}$  has the same rank as the matrix  $\mathbf{G}$ , hence the proof is completed. ■

### C. Parameters of LDPC Codes

Let  $\rho$  and  $\lambda$  be two integers such that  $1 \leq \rho < n$ . Let  $H(\rho, \lambda)$  be a sub-matrix of the matrix  $\mathbf{H}$  satisfying the row (column) constraints as above. The parameter  $\rho$  represents the number of nonzero positions in a column;  $\rho$  is a weight of a column. Also, the parameter  $\lambda$  represents the number of nonzero positions in a row;  $\lambda$  is a weight of a row. We can always assume that  $\lambda = (n - 1)^2$  for the Latin square construction. The null-space of the matrix  $\mathbf{H}(\rho, \lambda)$  gives a  $(\rho, \lambda)$  regular dual-containing LDPC code of length  $\lambda n$  and rate  $(\lambda - \rho)/\lambda$ . This construction gives a class of regular LDPC codes.

*Theorem 14:* For a prime integer  $n$ , the regular LDPC code generated by the parity check matrix  $\mathbf{H}$  is dual-containing and it has rate  $\frac{\lambda - \rho}{\lambda}$ .

*Proof:* We need to show that the matrix  $\mathcal{G}_j$  is also self-orthogonal as well as  $\mathcal{G}_j \times \mathcal{G}_i^T = 0$  for  $1 \leq i \leq \rho$ .

- i) Since  $n$  is a prime, then  $n - 1$  is an even integer. Let

$$\mathbf{H} = \begin{pmatrix} \mathbb{H}_1 \\ \mathbb{H}_2 \\ \vdots \\ \mathbb{H}_{n-1} \end{pmatrix} \quad (12)$$

Let  $g_l$  and  $g_k$  be two rows in  $\mathbb{H}_j$  over  $\mathbf{F}_2$ . Then  $g_k$  must be a permutation of the row  $g_l$  for  $k \neq l$ , hence they do not intersection at any position from the Latin square property. So,  $g_l * g_k^T = 0$ . Now, for  $l = k$ , from the assumption  $n$  is prime and  $g_l$  has exactly  $(n-1)$  nonzero element, each of length  $(n-1)$ , therefore,  $g_l$  has even weight (multiplicity even), hence it is self-orthogonal.

- ii) Now, let us choose any two arbitrary rows  $g_{jl}$  in  $\mathbb{H}_j$  and  $g_{ik}$  in  $\mathbb{H}_i$ . By Lemma 12 and using a similar argument as in i) one can show that  $g_{jl} * g_{ik}^T = 0$ .

- iii) The claim about the rate comes from our algorithm in Fig. 1. The result follows. ■

*Lemma 15:* The stopping distance of LDPC codes derived from Latin squares is exactly  $n$ .

*Proof:* By applying Definition 6, one can see that the number of columns that have rows with weight one is  $n$ . ■

By a similar argument one can also compute the stopping set and number of cycles with length 4.

We finish this construction by giving an example.

*Example 1:* Let  $q = 5 = n$  and  $\alpha$  be a primitive element in  $\mathbf{F}_q$ . Let  $\lambda = (n - 1)^2 = 16$  and  $\rho = 2$ , the generator matrix



A quantum code  $Q$  is defined as +1 joint eigenstates of the stabilizer  $S$ . Therefore, a codeword state  $|\psi\rangle$  belongs to the code  $Q$  if

$$S_j|\psi\rangle = |\psi\rangle \text{ for all } S_j \in S.$$

For more details on the error operators and stabilizer formalism, see for example [2], [9], [14]. In short a Pauli error operator  $E = E_1 \otimes E_2 \otimes \dots \otimes E_n$  where  $E_i$  in  $P$  can be mapped to a binary string of length  $2n$ . We take the elements  $X$  in  $S$  and map them to 1 and other elements are mapped to 0, so we form the matrix  $H_X$ . Similarly, the elements  $Z$  in  $S$  are mapped to 1 to form  $H_Z$ .

**CSS Construction:** A well-known construction of quantum codes from two classical nested codes is called CSS (i.e Calderbank, Shor and Steane). The CSS construction assumes that the stabilizer subgroup (matrix) can be written as

$$S = \left( \begin{array}{c|c} \mathbf{H} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{G} \end{array} \right) \quad (19)$$

where  $\mathbf{H}$  and  $\mathbf{G}$  are  $k \times n$  matrixes satisfying  $\mathbf{H}\mathbf{G}^T = \mathbf{0}$ . The quantum code with stabilizer  $S$  is able to encode  $n-2k$  logical qubits into  $n$  physical qubits. If  $\mathbf{G} = \mathbf{H}$ , then the stabilizer has the form

$$S = \left( \begin{array}{c|c} \mathbf{H} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{H} \end{array} \right) \quad (20)$$

and the self-orthogonality or dual-containing condition becomes  $\mathbf{H}\mathbf{H}^T = \mathbf{0}$ . If  $C$  is a code that has a parity check matrix  $\mathbf{H}$ , then  $C^\perp \subset C$ , where  $C^\perp$  is the dual code.

**Constructing Dual-containing LDPC Codes:** Let us construct the stabilizer matrix

$$S_{stab} = \left( \begin{array}{c|c} H_X & \mathbf{0} \\ \hline \mathbf{0} & H_Z \end{array} \right). \quad (21)$$

The matrix  $\mathbf{H}$  is a binary self-orthogonal matrix, where we replace every nonzero element in  $\mathbf{H}$  by the Pauli matrix  $X$  to form the matrix  $H_X$ . Similarly, we replace every nonzero element in  $\mathbf{H}$  by the Pauli matrix  $Z$  to form the matrix  $H_Z$ . Therefore the matrix  $S_{stab}$  is also self-orthogonal. We can assume that the matrix  $H_X$  corrects the bit-flip errors, while the matrix  $H_Z$  corrects the phase-flip errors, see [14].

**Proposition 16:** A quantum LDPC code  $Q$  with rate  $(n - 2k)/n$  is a code whose stabilizer matrix  $S_{stab}$  of size  $2k \times 2n$  has a parity check matrix  $\mathbf{H}$  with pair  $(\rho, \lambda)$  where  $\rho$  is the number of non-zero error operators in a column and  $\lambda$  is the number of non-zero error operators in a row.

We now give a family of quantum LDPC codes constructed from self-orthogonal LDPC codes that is based on elements of Latin squares.

**Lemma 17:** Let  $n$  be the order of a Latin square where  $q = n$  for some prime  $q$ . Let  $\mathbf{H}(\rho, \lambda)$  be a parity check matrix of a LDPC code over  $\mathbf{F}_2$  with column weight  $\rho$  and row weight  $\lambda$ . Then, there exists a quantum LDPC code with parameters  $[[\lambda n, \lambda n - 2n\rho, \geq \rho]]_2$ .

*Proof:* We know that there exists a regular LDPC code with a parity check matrix  $\mathbf{H}$  constructed from Latin squares of order  $n$ , see steps in Fig. 1. The matrix  $\mathbf{H}$  of size  $\rho n \times \lambda n$  has row weight  $\rho$  and column weight  $\lambda = (n - 1)^2$ . From Theorem 14, the parity check matrix  $\mathbf{H}$  is self-orthogonal and by Proposition 16 it defines a stabilizer matrix in the form  $S_{stab} = \left( \begin{array}{c|c} \mathbf{H} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{H} \end{array} \right)$ .

The quantum code is also defined over  $\mathbf{F}_2$  and has parameters  $[[N, M, d_{min}]]$  where  $N = \lambda n$  and  $M = \lambda n - 2\rho n$ , and  $d_{min} \geq \rho$ . ■

The stabilizer matrix of the quantum code  $Q$  is derived from a QC-LDPC code. Consequently, we can use any classical iterative decoding algorithm to estimate error operators. A step in this regard has been taken by Camara *et al.* in [3]. They also constructed regular LDPC code from group theory. We can conclude that our method of constructing QC-LDPC codes is simple and benefits from iterative decoding algorithms as well as easy encoders.

## V. DISCUSSION

We note that the constructed codes have reasonable performance in comparison to MacKay's work in random constructions of LDPC codes.

LDPC codes shown in [17] and [24] have good performance because these constructions of LDPC based on Latin squares do not need the parity check matrices to be self-orthogonal. So, they have fewer (orthogonal) Latin squares spread in the parity check matrices. In comparison to our work, we have reasonable performance, and our parity check matrices are self-orthogonal, consequently they have some cycles of length 4. Based on our work, we can highlight the following issues:

- i) It will be interesting to bound the maximum number of 4-cycle in the parity check matrix. In our construction, it can be checked that the upper bound is the length of the Latin squares, but this is not a tight bound since many rows in the parity check matrix have at most 2 or 4 positions in common.
- ii) Other constructions of LDPC codes based on finite geometry might give better performance of self-orthogonal LDPC codes. In addition, the minimum distance and the stopping set of these codes can be computed easily.
- iii) Cyclic LDPC and QC LDPC are beneficial codes because, in addition to their iterative decoding algorithms, they have efficient encoding algorithms using shift registers.

## VI. CONCLUSION

We introduced a family of quantum LDPC codes based on Latin squares. Our construction is simple in comparison to other constructions that use random approaches. Furthermore, one can use iterative decoding algorithms to decode these codes.

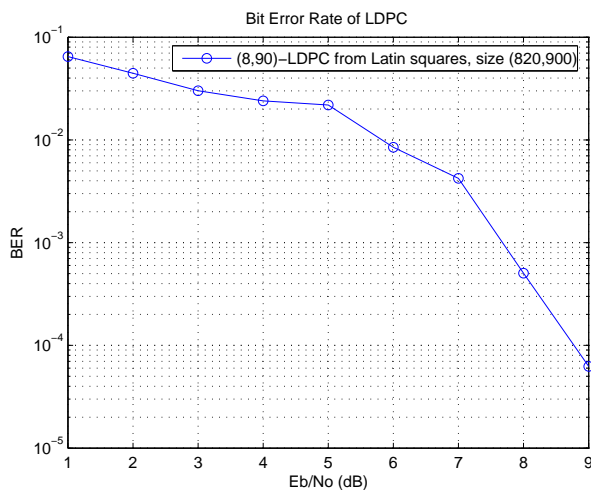


Fig. 3. Performance of an (8,90) LDPC code with parameters (820,900) based on Latin squares

## REFERENCES

- [1] S. A. Aly, A. Klappenecker, and P. K. Sarvepalli. Quantum convolutional codes derived from Reed-Solomon and Reed-Muller codes. *submitted to ISIT07, 2007*. quant-ph/0701037v1.
- [2] A. R. Calderbank, E. M. Rains, P. W. Shor, and N. J. A. Sloane. Quantum error correction via codes over GF(4). *IEEE Trans. Inform. Theory*, 44:1369–1387, 1998.
- [3] T. Camara, H. Ollivier, and J. P. Tillich. Constructions and performance of classes of quantum LDPC codes. 2005. quant-ph/00000.
- [4] M.C. Davey and D.J.C. MacKay. Low density parity check codes over GF(q). *IEEE Commun. Lett.*, 2(6):165–67, 1998.
- [5] C. Di, I.E. Proietti, Telatar, T.J. Richardson, and R. Urbanke. Finite-length analysis of low-density parity check codes on the binary erasure channel. *IEEE Trans. Inform. Theory*, 48:1570–1579, June 2000.
- [6] R.G. Gallager. *Low Density Parity Check Codes*. MIT Press, Cambridge, 1963.
- [7] M. Hagiwara and H. Imai. Quantum quasi-cyclic LDPC codes. *submitted to ISIT07, 2007*. quant-ph/071020v1.
- [8] C. Kelley, D. Sridhara, and J. Rosenthal. Tree-based construction of LDPC codes having good pseudocodeword weights. *IEEE Trans. Inform. Theory*, 2006.
- [9] A. Ketkar, A. Klappenecker, S. Kumar, and P. K. Sarvepalli. Nonbinary stabilizer codes over finite fields. *IEEE Trans. Inform. Theory*, 52(11):4892–4914, 2006.
- [10] K. M. Krishnan and P. Shankar. Computing the stopping distance of a tanner graph is NP-hard. *IEEE Trans. Inform. Theory*, To appear, 2007.
- [11] S. Laendner and O. Milenkovic. LDPC codes based on Latin squares: Cycle structure, stopping set, and trapping set analysis. *IEEE Trans. Inform. Theory*, 55(2):303–312, 2007.
- [12] S. Lin and D.J. Costello. *Error Control Coding*. Pearson, Prentice Hall, 2004.
- [13] G. Liva, S. Song, Y. Ryan W. Lan, L. Zhang, and S. Lin. Design of LDPC codes: A survey and new results. *to appear in J. Comm. Software and Systems*, 2006.
- [14] D. J. C. MacKay, G. Mitchison, and P. L. McFadden. Sparse-graph codes for quantum error correction. *IEEE Trans. Inform. Theory*, 50(10):2315–2330, 2004.
- [15] W. Matsumoto and H. Imai. Irregular extended euclidean geometry low-density parity-check codes.
- [16] B.D. McKay, A. Meynert, and W. Myrvold. Small Latin squares, quasigroups, and loops. *Journal of Combinatorial Designs*, 15(2):98–119, month = , note = , abstract = , keywords = , source = , 1998.
- [17] O. Milenkovic and S. Laendner. Analysis of the cycle-structure of LDPC codes based on Latin squares. *IEEE communications society*, pages 777–781, 2004.
- [18] A. Orlitsky, K. Viswanatham, and J. Zhang. Stopping set distribution of LDPC code ensembles. *IEEE Trans. Inform. Theory*, 51(3):929–949, March 2005.
- [19] M. Schwartz and A. Vardy. On the stopping distance and the stopping redundancy of codes. *IEEE Trans. Inform. Theory*, 55(3):922–932, March 2006.
- [20] S. Song, L. Lan, S. Lin, and K. Abdel-Ghaffar. Construction of quasi-cyclic LDPC codes based on the primitive elements of finite fields. 2006.
- [21] S. Song, L. Zeng, S. Lin, and K. Abdel-Ghaffar. Algebraic constructions of nonbinary quasi-cyclic LDPC codes. *Proc. 2006 IEEE Intl. Symp. Inform. Theory*, pages 83–87, 2006.
- [22] R.M. Tanner. A recursive approach to low complexity codes. *IEEE Trans. Inform. Theory*, 27:533–47, 1981.
- [23] R.M. Tanner, D. Sridhara, A. Sridharan, T. Fuja, and D. Costello Jr. LDPC block and convolutional codes based on circulant matrices. *IEEE Trans. Inform. Theory*, 50(12):2966–2984, December 2004.
- [24] B. Vasic, E. Kurtas, and A. Kuznetsov. LDPC codes based on mutually orthogonal Latin rectangles and their application in perpendicular magnetic recording. *IEEE. trans. Magnetics*, 38(5, part: 1):2346–2348, 2002.
- [25] I.M. Wanless. Atomic Latin squares based on cyclotomic orthomorphisms. *the electronic journal of combinatorics*, 12, 2005.