On Construction of Two Classes of Efficient Quantum Error-Correction Codes

Peiyu Tan and Jing Li (Tiffany)
Electrical and Computer Engineering Department, Lehigh University, Bethlehem, 18015
{pet3,jingli}@ece.lehigh.edu

***Abstract*— Stabilizer codes are an important branch of quantum codes, but the existing stabilizer codes constructed from the classical binary codes almost exclusively belong to the special subclass of CSS codes. This paper develops simple and systematic constructions for two rich classes of non-CSS quantum stabilizer codes: quantum low density parity check (LDPC) codes based on classical quasi-cyclic LDPC codes and quantum convolutional codes based on classical LDPC-convolutional codes. Both classes enjoy a wide range of lengths and rates, and offer stronger error correction capability than the existing codes.**

I. Introduction

The inception and development of quantum error-correction codes (QECC) is intended to protect the fragile quantum states from unwanted evolutions and to allow robust implementations of quantum processing devices. Analog to a bit in classical systems, a *quantum bit*, or, simply, a *qubit*, is defined as a physical system exhibiting quantum properties in quantum computation and communication systems. Unlike a classical bit, the state of a qubit is not limited to $|0\rangle$ or $|1\rangle$, but includes linear combinations, known as *superposition* in the quantum literature, of these basic states: $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, where α and β are complex numbers satisfying $|\alpha|^2+|\beta|^2 = 1$. The quantum state of n qubits can be expressed as $(\alpha_0|0...00\rangle + \alpha_1|0...01\rangle + ... + \alpha_{2^n-1}|1...11\rangle)$, where $|\alpha_0|^2 + |\alpha_1|^2 + ... + |\alpha_{2^n-1}|^2 = 1$, and the 2^n states ($|0...00\rangle$, $|0...01\rangle$, ..., $|1...11\rangle$) are shorthand for the tensor products $|0\rangle \otimes ... \otimes |0\rangle$, $|0\rangle \otimes ... \otimes |1\rangle$, ..., $|1\rangle \otimes ... \otimes |1\rangle$. Although quantum states and quantum errors both have *continuous* values, it has been proven that continuous quantum errors are nevertheless correctable, provided that the quantum states are appropriately encoded [1]!

Early research focuses on single-error-correcting codes [1]. Recent research effort has resulted in a variety of quantum block codes [11] [12], quantum low density parity check (LDPC) codes [5]–[8], and quantum convolutional codes [9], [10]. A majority of these codes belong to the family of *stabilizer codes*, whose theory is perhaps the only near-mature theory in quantum coding [4]–[13]. Stabilizer codes may generate from classical binary codes or non-binary codes (such as codes in quaternary field F4). This work investigates the first category. A subclass in this category, *CSS codes*, which are based on classical binary dual codes and named after their inventors Calderbank, Shor and Steane [2], [3], has gained particular popularity.

A major reason for the predominance of CSS codes is the lack of formal construction methods to satisfy the *general*, rather than a special case of, *symplectic inner product condition* (also known as *twisted product condition* [6]) required for stabilizer codes (see Section II). This paper fills in the gap by developing systematic ways to construct two rich families of non-CSS or unrestricted stabilizer codes: quantum LDPC codes based on classical binary quasi-cyclic (QC) LDPC codes, and quantum convolutional codes based on classical binary LDPC-convolutional codes. To the best of the authors' knowledge, the LDPC codes developed here are the first non-CSS LDPC quantum codes reported to date (be the base code binary or non-binary), and the convolutional codes developed here are the first unrestricted convolutional quantum codes based on classical binary codes. Because our codes root to powerful classical codes, they are capable of correcting a large number of quantum errors, including bursty errors, in a block. This is much more useful than early quantum codes that correct only a single error in each block. Additionally, both families of codes enjoy a wide range of lengths and rates, including very high rates that approach 1.

Supported in part by National Science Foundation under grant CCF-0430634 and CCF-0635199, and by the Commonwealth of Pennsylvania through the Pennsylvania Infrastructure Technology Alliance (PITA).

II. Quantum Stabilizer Codes

The mathematical representation of an erroneous state $|\psi'\rangle$ is the product of the original state and the noise ($E|\psi\rangle$). It is common practice to neglect the amplitude of E (which can be handled by normalization) and consider E as an arbitrary unitary linear operator.

Definition 1: [Error Operator] Suppose a vector $|\psi\rangle$ is sent through the depolarizing channel. The outcome of the transmission can be written as $E|\psi\rangle$, where the error operator E takes the form of $E = \tau_1 \otimes \tau_2 \otimes ... \otimes \tau_n$, and each τ_i is either $I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$, or one of the following Pauli matrices $X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$, $Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$, $Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$.

Here, X, Z and Y correspond respectively to a bit flip, a phase flip, and a combination of bit and phase flip. A quantum code can correct any unitary type error if it can simultaneously correct X error and Z error.

We first introduce stabilizer codes using the language of finite group theory, which makes easy connection to the binary language of quantum mechanics. In quantum error protection, the structure to store quantum information is a subspace of the total Hilbert space of the physical qubits, thereafter referred to as the code space $\mathcal{C}$.

Definition 2: [Stabilizer Codes] A *stabilizer code* has a code space $\mathcal{C}$ that is the largest subspace stabilized by an Abelian group S acting on the N physical qubits of the code. A *stabilizer* S is some Abelian subgroup of G_N (the tensor products of N matrices in $G = \{I, X, Y, Z\}$) satisfying

(i) neither i nor -1 is in S; and

(ii) S fixes all the codewords, i.e.

$$\mathcal{C} = \{|\psi\rangle, \text{ s.t. } \mathrm{M}|\psi\rangle = |\psi\rangle, \forall \mathrm{M} \in \mathrm{S}\}. \quad (1)$$

The set of valid codewords are eigenvectors of all the operators in S with eigenvalue $+1$. Since the stabilizer S can be described using independent stabilizer generators $\{S_i\}$ such that the set S comprises S_i and all products of S_i, it is sufficient to define a quantum codeword as a state $|\psi\rangle$ such that $S_i|\psi\rangle = |\psi\rangle$ for all i.

When an error operator changes the original state of a qubit, the corrupted state may fall either inside the code space or outside. The former case results in what's known in the (classical) coding literature as *un-detectable errors*, a type of errors which exceed the error-correction capability of any code and which should occur rarely in a well-designed code. The latter case is the task of quantum error correction codes. Since it occurs when the error operator anti-commutes with some element of S, stabilizer generators S_i can act as diagnostic operators to evaluate the commutation property of a given state. This can be achieved by, for example, using S_i to measure the eigenvalue array, known as the *syndrome*, of a given state. Further, a syndrome identifies the error operator that can counter-act the specific error, which can therefore be applied to the corrupted state to recover the original state.

Binary Formalism of Stabilizer Codes: To connect and compare quantum error-correction codes with classical (digital) error-correction codes, below we introduce stabilizer codes using the language of linear algebra in GF(2) field. To differentiate quantum codes and classical codes, we thereafter denote a classical code as an $[N, K]$ code and denote a quantum code as an $[[N, K]]$ code.

The $N-K$ stabilizer generators can be described as the concatenation of a *pair* of $(N-K) \times N$ binary matrices, $\mathbf{A} = (\mathbf{A_1}|\mathbf{A_2})$, such that each row in $\mathbf{A}$ corresponds to a unique stabilizer generator, and each pair of columns (the corresponding columns in $\mathbf{A_1}$ and $\mathbf{A_2}$) correspond to a qubit. A 0 entry in $\mathbf{A}$ represents an I operator, and a 1 entry represents either an X or Y operator if it is in the $\mathbf{A}_1$-matrix, or a Z or Y operator if it is in the $\mathbf{A}_2$-matrix.

For example, the $[[5, 1]]$ cyclic quantum code is defined by the following stabilizer:

$$\begin{array}{ccccc} X & Z & Z & X & I \\ I & X & Z & Z & X \\ X & I & X & Z & Z \\ Z & X & I & X & Z \end{array} \quad (2)$$

which takes a binary form of:

$$\mathbf{A} = \left(\begin{array}{ccccc|ccccc} 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{array}\right).$$

In the binary formalism, the condition that two stabilizers commute with each other translates to that their *symplectic inner product*, or, *twisted product*, is 0. Let $r_m = (x_m, z_m)$ be the m-th row in $\mathbf{A}$, where x_m and z_m denote the m-th rows in $\mathbf{A}_1$- and $\mathbf{A}_2$-matrices respectively. The symplectic inner product $\odot$ of the m-th and the m'-th rows is

$$r_m \odot r_{m'} = x_m \cdot z_{m'} + x_{m'} \cdot z_m \mod 2, \quad (3)$$

where $x_m \cdot z_{m'} = \sum_i x_{mi} z_{m'i}$. If we write them in compact matrix forms, we get:

Lemma 1: [Symplectic Inner Product Condition] A stabilizer with binary representation $\mathbf{A} = (\mathbf{A}_1|\mathbf{A}_2)$ satisfies the symplectic inner product condition if and only if:

$$\mathbf{A}_1\mathbf{A}_2^T + \mathbf{A}_2\mathbf{A}_1^T = 0, \quad (4)$$

where T stands for matrix transpose.

A binary matrix $\mathbf{A}$, having size $M_Q \times 2N$ and satisfying the symplectic inner product condition in (4), defines a rate-$(N-M_Q)/N$ quantum code that encodes $N-M_Q$ qubits into N qubits.

Definition 3: [CSS Codes] An $[[N, N-2M]]$ *CSS code* is specified by a stabilizer with the following binary formalism:

$$\mathbf{A} = \left(\begin{array}{c|c} \mathbf{H} & \mathbf{0} \\ \mathbf{0} & \mathbf{G} \end{array}\right), \quad (5)$$

where $\mathbf{H}$ and $\mathbf{G}$ are $M \times N$ matrices (yet not restricted to have the same number of rows) satisfying $\mathbf{H}\mathbf{G}^T = 0$.

One may further restrict $\mathbf{H} = \mathbf{G}$, such that

$$\mathbf{A} = \left(\begin{array}{c|c} \mathbf{H} & \mathbf{0} \\ \mathbf{0} & \mathbf{H} \end{array}\right), \quad (6)$$

and $\mathbf{H}\mathbf{H}^T = 0$. This results in the so-called *CSS codes based on dual-containing codes*, since $\mathbf{H}$ comes from the parity check matrix of a dual-containing (or weakly self-dual) classical code.

The CSS method provides a convenient means to satisfy the symplectic inner product condition, but also significantly narrows the choice of quantum codes. For example, it is not possible to construct the [[5,1]] stabilizer code in (2) using the CSS method. Sad enough, to date, there are very few reported quantum stabilizer codes that fall outside the subclass of CSS codes [2] [3]. In what follows, we will demonstrate perhaps the first systematic ways to construct two rich families of non-CSS stabilizer codes: quantum QC-LDPC codes and quantum LDPC-convolutional codes. These new codes possess simple construction, a wide range of sizes and rates (including very high rates), and the capability of correcting hundreds of errors.

III. Constructing Quantum QC-LDPC Codes

The first construction of quantum LDPC codes, due to Postol [5], uses finite geometry in the CSS method. Mackay developed four more approaches based on cyclic matrices, difference sets and (exhaustive) random search to construct CSS-type quantum LDPC codes [6]. More recently, [8] constructed quantum LDPC codes using group theory, and [7] proposed the method of matrix splitting/merging.

Now our approach is very different from, and arguably much simper than, the existing ones. The key challenge for LDPC stabilizer codes is to construct binary *sparse* matrices $\mathbf{A} = (\mathbf{A}_1|\mathbf{A}_2)$ that satisfy the general symplectic inner product condition in (4) through a *simple* and *systematic* approach. Enlightened by classical quasi-cyclic LDPC codes, we assign $\mathbf{A}_1$ and $\mathbf{A}_2$ cyclic forms, and derive sufficient conditions to satisfy the symplectic inner product condition.

Definition 4: [Cyclic Matrices] Let $\mathbf{I}$ be an $m \times m$ identity matrix. Let $\mathbf{I}_x$ be a shifted identity matrix with the rows of $\mathbf{I}$ cyclically shifted to the right by x positions, where $x \in [0, m-1]$ is called the *offset*. For convenience, we assume $\mathbf{I}_0 = \mathbf{I}$ and $\mathbf{I}_{x \pm km} = \mathbf{I}_x$ for any integer k. A weight-t binary

cyclic matrix is the (binary) sum of t shifted identity matrices of the same dimensionality but different offsets:

$$\mathbf{B} = \mathbf{I}_{x_1} + \mathbf{I}_{x_2} + \cdots + \mathbf{I}_{x_t}, \quad 0 \le x_1 < x_2 < \cdots < x_t < m. \quad (7)$$

Lemma 2: [Symplectic Inner Product Condition for Even-Weight Cyclic Matrices] Let $\mathbf{A}_1 = \mathbf{I}_{a_1} + \mathbf{I}_{a_2} + \cdots + \mathbf{I}_{a_t}$ and $\mathbf{A}_2 = \mathbf{I}_{b_1} + \mathbf{I}_{b_2} + \cdots + \mathbf{I}_{b_t}$, where t is even. A sufficient condition to satisfy the symplectic inner product condition is:

$$\begin{aligned} & a_1 + a_2 = a_3 + a_4 = \cdots = a_{t-1} + a_t \\ = & b_1 + b_2 = b_3 + b_4 = \cdots = b_{t-1} + b_t, \quad (mod\ m), \quad (8) \\ \text{and} \quad & a_1 \neq \cdots \neq a_t \neq b_1 \neq \cdots \neq b_t, \quad (9) \end{aligned}$$

Proof: (Proof by induction) First, consider weight-2 cyclic matrices: $\mathbf{A}_1 = \mathbf{I}_{a_1} + \mathbf{I}_{a_2}$ and $\mathbf{A}_2 = \mathbf{I}_{b_1} + \mathbf{I}_{b_2}$.

$$\begin{aligned} & \mathbf{A}_1\mathbf{A}_2^T + \mathbf{A}_2\mathbf{A}_1^T \\ = & (\mathbf{I}_{a_1} + \mathbf{I}_{a_2})(\mathbf{I}_{b_1} + \mathbf{I}_{b_2})^T + (\mathbf{I}_{b_1} + \mathbf{I}_{b_2})(\mathbf{I}_{a_1} + \mathbf{I}_{a_2})^T \\ = & (\mathbf{I}_{a_1} + \mathbf{I}_{a_2})(\mathbf{I}_{-b_1} + \mathbf{I}_{-b_2}) + (\mathbf{I}_{b_1} + \mathbf{I}_{b_2})(\mathbf{I}_{-a_1} + \mathbf{I}_{-a_2}) \\ = & \mathbf{I}_{a_1-b_1} + \mathbf{I}_{a_1-b_2} + \mathbf{I}_{a_2-b_1} + \mathbf{I}_{a_2-b_2} \\ & + \mathbf{I}_{b_1-a_1} + \mathbf{I}_{b_1-a_2} + \mathbf{I}_{b_2-a_1} + \mathbf{I}_{b_2-a_2}. \quad (10) \end{aligned}$$

It is easy to verify that the last equality becomes 0 if $a_1 + a_2 = b_1 + b_2 \quad (mod \quad m)$ and $a_1 \neq a_2 \neq b_1 \neq b_2$.

Now suppose that the condition is sufficient for some even weight t. Let us evaluate the case of $t+2$. Suppose $\mathbf{A}_1 = \mathbf{I}_{a_1} + ... + \mathbf{I}_{a_t}$ and $\mathbf{A}_2 = \mathbf{I}_{b_1} + ... + \mathbf{I}_{b_t}$ have zero symplectic inner product. Evaluate the symplectic inner product of $(\mathbf{A}_1 + \mathbf{I}_{a_{t+1}} + \mathbf{I}_{a_{t+2}})$ and $(\mathbf{A}_2 + \mathbf{I}_{b_{t+1}} + \mathbf{I}_{b_{t+2}})$.

$$\begin{aligned} & (\mathbf{A}_1 + \mathbf{I}_{a_{t+1}} + \mathbf{I}_{a_{t+2}})(\mathbf{A}_2 + \mathbf{I}_{b_{t+1}} + \mathbf{I}_{b_{t+2}})^T \\ & + (\mathbf{A}_2 + \mathbf{I}_{b_{t+1}} + \mathbf{I}_{b_{t+2}})(\mathbf{A}_1 + \mathbf{I}_{a_{t+1}} + \mathbf{I}_{a_{t+2}})^T \\ = & \mathbf{A}_1{\mathbf{A}_2}^T + \mathbf{A}_1(\mathbf{I}_{b_{t+1}} + \mathbf{I}_{b_{t+2}})^T + (\mathbf{I}_{a_{t+1}} + \mathbf{I}_{a_{t+2}}){\mathbf{A}_2}^T \\ & + \mathbf{A}_2{\mathbf{A}_1}^T + \mathbf{A}_2(\mathbf{I}_{a_{t+1}} + \mathbf{I}_{a_{t+2}})^T + (\mathbf{I}_{b_{t+1}} + \mathbf{I}_{b_{t+2}}){\mathbf{A}_1}^T \\ & + (\mathbf{I}_{a_{t+1}} + \mathbf{I}_{a_{t+2}})(\mathbf{I}_{b_{t+1}} + \mathbf{I}_{b_{t+2}})^T \\ & + (\mathbf{I}_{b_{t+1}} + \mathbf{I}_{b_{t+2}})(\mathbf{I}_{a_{t+1}} + \mathbf{I}_{a_{t+2}})^T \\ = & \underbrace{\mathbf{A}_1(\mathbf{I}_{b_{t+1}} + \mathbf{I}_{b_{t+2}})^T + (\mathbf{I}_{b_{t+1}} + \mathbf{I}_{b_{t+2}}){\mathbf{A}_1}^T}_{p_1} \\ & + \underbrace{(\mathbf{I}_{a_{t+1}} + \mathbf{I}_{a_{t+2}}){\mathbf{A}_2}^T + \mathbf{A}_2(\mathbf{I}_{a_{t+1}} + \mathbf{I}_{a_{t+2}})^T}_{p_2} \\ & + (\mathbf{I}_{a_{t+1}} + \mathbf{I}_{a_{t+2}})(\mathbf{I}_{b_{t+1}} + \mathbf{I}_{b_{t+2}})^T \\ & + (\mathbf{I}_{b_{t+1}} + \mathbf{I}_{b_{t+2}})(\mathbf{I}_{a_{t+1}} + \mathbf{I}_{a_{t+2}})^T \quad (11) \end{aligned}$$

The sum of the last two terms becomes 0, if $a_{t+1} + a_{t+2} = b_{t+1} + b_{t+2}$ and a_{t+1}, a_{t+2}, b_{t+1}, b_{t+2} are all distinct (see the discussion of (10)).

By grouping the weight-1 cyclic matrices in $\mathbf{A}_1$ in matching pairs, the term p_1 can be rewritten as

$$\begin{aligned} p_1 = & \mathbf{A}_1(\mathbf{I}_{b_{t+1}} + \mathbf{I}_{b_{t+2}})^T + (\mathbf{I}_{b_{t+1}} + \mathbf{I}_{b_{t+2}})\mathbf{A}_1{}^T \\ = & \big((\mathbf{I}_{a_1} + \mathbf{I}_{a_2}) + ... + (\mathbf{I}_{a_{t-1}} + \mathbf{I}_{a_t})\big)(\mathbf{I}_{b_{t+1}} + \mathbf{I}_{b_{t+2}})^T \\ & + (\mathbf{I}_{b_{t+1}} + \mathbf{I}_{b_{t+2}})\big((\mathbf{I}_{a_1} + \mathbf{I}_{a_2})^T + ... + (\mathbf{I}_{a_{t-1}} + \mathbf{I}_{a_t})^T\big) \end{aligned}$$

If $a_1 + a_2 = a_3 + a_4 = \cdots = a_{t-1} + a_t = a_{t+1} + a_{t+2} = b_{t+1} + b_{t+2}$, and $a_1, \cdots, a_{t+2}$, b_{t+1}, b_{t+2} are all distinct, then $p_1 = 0$. Similarly, we can show that $p_2 = 0$, when $b_1 + b_2 = b_3 + b_4 = \cdots = b_{t-1} + b_t = b_{t+1} + b_{t+2} = a_{t+1} + a_{t+2}$ and $b_1, \cdots, b_{t+2}$, a_{t+1}, a_{t+2} are all distinct.

Hence, the sufficient condition for zero symplectic inner product is: $a_1 + a_2 = a_3 + a_4 = \cdots = a_{t-1} + a_t = b_1 + b_2 = b_3 + b_4 = \cdots = b_{t-1} + b_t$ and $a_1, \cdots, a_t$, $b_1, \cdots, b_t$ are all distinct, where t is the even weight. □

Lemma 2 reveals a convenient construction for unrestricted stabilizer codes, except the practicality that the matrix $\mathbf{A}$ has dimensionality $m \times 2m$, leaving the resulting quantum code zero-rate and useless. One simple way to rectify this is to concatenate cyclic (square) matrices to form quasi-cyclic (non-square) matrices.

Lemma 3: [Concatenation Preserves Symplectic Inner Product Condition] Consider two families of cyclic (square) matrices having the same dimensionality: $\{\mathbf{A}_{1i}\}$ and $\{\mathbf{A}_{2i}\}$, $i = 1, 2, \cdots, n$. Let $\mathbf{A}_1 \triangleq [\mathbf{A}_{11}|\mathbf{A}_{12}|...|\mathbf{A}_{1n}]$ and $\mathbf{A}_2 \triangleq [\mathbf{A}_{21}|\mathbf{A}_{22}|...|\mathbf{A}_{2n}]$ be their respective concatenations. If $\mathbf{A}_{1i}\mathbf{A}_{2i}^T + \mathbf{A}_{2i}\mathbf{A}_{1i}^T = 0$ for any integer $i \in [1, n]$, then $\mathbf{A}_1\mathbf{A}_2^T + \mathbf{A}_2\mathbf{A}_1^T = 0$.

Proof: Proof of Lemma 3 follows directly from the definitions of matrix transpose and matrix multiplication.□

Lemma 3 is very useful. It provides an efficient way to construct quasi-cyclic sparse matrix $\mathbf{A} = [\mathbf{A}_1|\mathbf{A}_2]$ of size $m \times 2m\tau$, by properly concatenating τ pairs of matching cyclic (square) matrices. The resulting code is a QC-LDPC quantum code with rate $\frac{\tau-1}{\tau}$ and codeword length $m\tau$. With $\tau = 2, 3, ...,$ the code rate covers $1/2$, $2/3$, $\cdots$ and can get arbitrarily close to 1. This is noteworthy, since (1) the existing constructions of quantum codes do not usually support such a rich pool of rates in such a systematic way, and (2) the quantum codes reported in literature are mostly low-rate codes.

For good error correction capability, we recommend building quantum QC-LDPC codes from classical QC-LDPC codes that have good performance. Here is an illustrating example.

Example 1: [Quantum Regular QC-LDPC Codes] Suppose $\tau = 4$. [15] listed a good [828, 651] (classical) regular QC-LDPC code with column weight j = 3 and row weight k = 12, where the base identity matrices have uniform sizes of 217×217, and the parity check matrix is in the form of

$$\Big[\mathbf{I}_0 + \mathbf{I}_{121} + \mathbf{I}_{137} \,\Big|\, \mathbf{I}_8 + \mathbf{I}_{79} + \mathbf{I}_{85} \,\Big|\, \mathbf{I}_1 + \mathbf{I}_{11} + \mathbf{I}_{100} \,\Big|\, \mathbf{I}_{29} + \mathbf{I}_{165} + \mathbf{I}_{207}\Big].$$

Following Lemma 2, we add an additional weight in each sub-matrix to make the weight even and to satisfy (9):

$$\begin{aligned} \mathbf{A}_1 = & [\,\mathbf{A}_{11}|\mathbf{A}_{12}|\mathbf{A}_{13}|\mathbf{A}_{14}\,] \\ = & \Big[\mathbf{I}_0 + \mathbf{I}_{121} + \mathbf{I}_{137} + \mathbf{I}_{41} \,\Big|\, \mathbf{I}_8 + \mathbf{I}_{79} + \mathbf{I}_{85} + \mathbf{I}_{156} \,\Big|\, \mathbf{I}_1 + \mathbf{I}_{11} + \mathbf{I}_{100} + \mathbf{I}_{110} \,\Big|\, \mathbf{I}_{29} + \mathbf{I}_{165} + \mathbf{I}_{207} + \mathbf{I}_{124}\Big]. \end{aligned}$$

Again, we use Lemma 2 to get a QC matrix $\mathbf{A}_2$ that matches $\mathbf{A}_1$. One possible choice is:

$$\mathbf{A}_2 = \Big[\mathbf{I}_1 + \mathbf{I}_{120} + \mathbf{I}_{138} + \mathbf{I}_{40} \,\Big|\, \mathbf{I}_6 + \mathbf{I}_{81} + \mathbf{I}_{83} + \mathbf{I}_{158} \,\Big|\, \mathbf{I}_2 + \mathbf{I}_{10} + \mathbf{I}_{101} + \mathbf{I}_{109} \,\Big|\, \mathbf{I}_{34} + \mathbf{I}_{160} + \mathbf{I}_{212} + \mathbf{I}_{121}\Big].$$

$\mathbf{A}_1$ and $\mathbf{A}_2$ are capable of correcting bit-flip errors and phase-flip errors respectively. Both have column weight 4 and row weight 16. Concatenating them leads to the stabilizer (in its binary form) of a regular QC-LDPC code with rate 0.75 and block length 868 (qubits), as shown in Fig. 1.

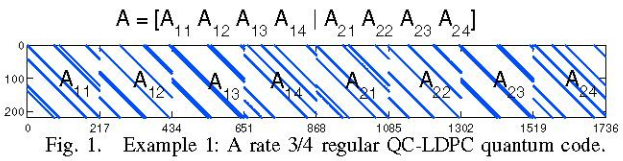

Fig. 1. Example 1: A rate 3/4 regular QC-LDPC quantum code.

This simple method allows us to borrow results from classical QC-LDPC codes and to assemble quantum QC-LDPC codes with a large variety of code lengths and code rates.

Since the additive structure of GF(4) accords with the multiplicative structure of the Pauli operators, a stabilizer code can be viewed as a code over GF(4)= $\{ 0, 1, \omega, \bar{\omega} \}$, where $1+\omega+\omega^2 = 1+\omega+\bar{\omega} = 0$. Hence, decoding is performed by converting $\mathbf{A} = [\mathbf{A}_1|\mathbf{A}_2]$ into a parity check matrix in GF(4), and then applying GF(4) message-passing.

The downside of this construction is the existence of length-4 cycles in the code graph when a sub-matrix has column weight $t \geq 4$, which may impair the performance of message-passing decoding. Two improvements are investigated to make the code perform better. The first is to make the structure of $\mathbf{A}$ slightly irregular by allowing the cyclic sub-matrices in $\mathbf{A}$ to have different weights. Like their classical counterparts, irregular quantum QC-LDPC codes generally provide better performance.

Example 2: [Quantum Irregular QC-LDPC Codes] The code in Example 2 has uniform weight 4 in all columns. Here we modify $\mathbf{A}_{14}$ and $\mathbf{A}_{24}$ to make their column weights 2. Specifically, we choose:

$$\mathbf{A}_1 = \Big[\mathbf{I}_0 + \mathbf{I}_{121} + \mathbf{I}_{137} + \mathbf{I}_{41} \,\Big|\, \mathbf{I}_8 + \mathbf{I}_{79} + \mathbf{I}_{85} + \mathbf{I}_{156} \,\Big| \, \mathbf{I}_1 + \mathbf{I}_{11} + \mathbf{I}_{100} + \mathbf{I}_{110} \,\Big|\, \mathbf{I}_{29} + \mathbf{I}_{165} \Big], \quad (12)$$

$$\mathbf{A}_2 = \Big[\mathbf{I}_1 + \mathbf{I}_{120} + \mathbf{I}_{138} + \mathbf{I}_{40} \,\Big|\, \mathbf{I}_6 + \mathbf{I}_{81} + \mathbf{I}_{83} + \mathbf{I}_{158} \,\Big| \, \mathbf{I}_2 + \mathbf{I}_{10} + \mathbf{I}_{101} + \mathbf{I}_{109} \,\Big|\, \mathbf{I}_{34} + \mathbf{I}_{160} \Big]. \quad (13)$$

The stabilizer of the resulting irregular quantum QC-LDPC code (in its binary form) has uniform row weight of 14, and slightly non-uniform column weight with 75% of 4 and 25% of 2. To demonstrate the effectiveness of the proposed construction, we simulate their corresponding classical codes over additive white Gaussian noise (AWGN) channels, and compare them to some of the best classical QC-LDPC codes. Each quantum stabilizer code has a pair of classical counterparts using $\mathbf{A}_1$ and $\mathbf{A}_2$ as their parity check matrices respectively. The decoding results, obtained after up to 100 message-passing iterations, are presented in Figure 2. The two solid curves with circle-marks indicate the pair of regular QC-LDPC codes from Example 1, the two dashed curves with circle-marks indicate the pair of irregular QC-LDPC codes from Example 2, and the other two curves indicate the best and the worst regular QC-LDPC [868, 651] ($j = 3, k = 12$) codes reported in literature [15]. We observe that the codes in Example 1 perform on average, and the codes in Example 2 beat the best regular QC-LDPC code reported in [15]. Although the AWGN channel simulated here differs from the realistic quantum channels, this evaluation is nevertheless useful: it clearly indicates that the proposed construction is not only simple and valid, but will also generate good codes. Additional code examples will be provided in Section V.

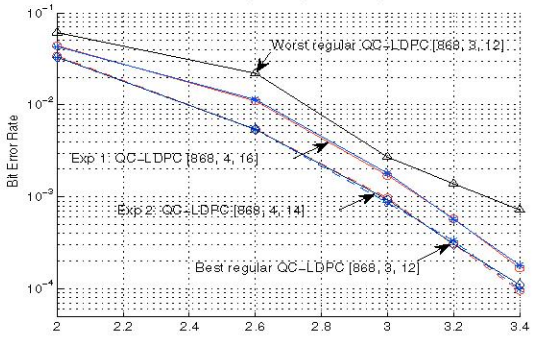

Fig. 2. Performance comparison of QC-LDPC codes in the classical error correction domain.

IV. Quantum LDPC-Convolutional Codes

Going from regular LDPC codes to irregular LDPC codes is a rather expected improvement. The second enhancement developed here surpasses the boundary of LDPC codes. Classical convolutional codes have been derived from classical LDPC codes, and have demonstrated certain advantages. We propose to perform similar transformation in the quantum domain. Our previous construction of QC-LDPC quantum codes has paved the way to convolutional quantum codes, since *any* QC-LDPC codes, yet *only QC codes*, will in principal have convolutional counterparts.

Example 3: [Quantum LDPC-Convolutional Codes] Consider the QC-LDPC code in Example 2, the basic building block (cyclic square matrix) has a dimensionality of 217×217. Using polynomial expressions, $\mathbf{A}_1$ becomes a 1×4 matrix in the $\mathcal{D}$-domain:

$$\mathbf{A}_1(D) = \begin{pmatrix} 1 + D^{121} + D^{137} + D^{41}, \\ D^8 + D^{79} + D^{85} + D^{156}, \\ D + D^{11} + D^{100} + D^{110}, \\ D^{29} + D^{165} \end{pmatrix}^T. \quad (14)$$

Similarly, we have

$$\mathbf{A}_2(D) = \begin{pmatrix} D + D^{120} + D^{138} + D^{40} \\ D^6 + D^{81} + D^{83} + D^{158} \\ D^2 + D^{10} + D^{101} + D^{109} \\ D^{34} + D^{160} \end{pmatrix}^T, \quad (15)$$

It can be verified that $\mathbf{A}_1(D)$ and $\mathbf{A}_2(D)$ **satisfy** the symplectic inner product condition for quantum convolutional codes. $\mathbf{A}(D) = [\mathbf{A}_1(D)|\mathbf{A}_2(D)]$ thus defines a rate 0.75 quantum convolutional code. To perform decoding, similar to the case of QC-LDPC codes, one needs to first convert the pair of binary semi-infinite matrices, $\mathbf{A}_1(D)$ and $\mathbf{A}_2(D)$, into one semi-infinite parity check matrix in GF(4). Then a sliding-window message-passing (SWMP) algorithm in GF(4), similar to the binary version developed for binary LDPC-convolutional codes [14], can be applied to decode this quantum LDPC-convolutional code. By definition [14], the memory size of the SWMP decoder is $m_s \leq m - 1$. Since the performance improves with m_s, one may set the memory size $m_s = m-1 = 216$ to maximize the coding gain, at the cost of a larger set of hardware and a higher complexity.

While LDPC-convolutional quantum codes are good in their own merits (e.g. they have flexible lengths), one of our initiatives was to reduce length-4 cycles in the QC-LDPC quantum codes we constructed earlier. This goal is fulfilled, since it is proven that the cycles in the LDPC-convolutional code can only result from the originating QC-LDPC code, yet not all the cycles in the QC-LDPC code will result in cycles in the

convolutional code. Further, the free distance d_{free} of the LDPC-convolutional code is lower-bounded by the minimum distance d_{min} of the QC-LDPC code.

It is also worth noting that the proposed LDPC-convolutional stabilizer codes with rate $(\tau-1)/\tau$ are, to the best of the authors' knowledge, the first class of high-rate quantum convolutional codes. There are only a few quantum convolutional codes reported in literature, and all of them have rather low rates (e.g. [10] [9]).

Remark: A simple but very useful observation we made is *concatenation preserves the symplectic inner product condition*. This observation enables us, in the design of QC-LDPC codes, to divide $\mathbf{A}_1$ and $\mathbf{A}_2$ into multiple square matrices $\{\mathbf{A}_{1i}\}$ and $\{\mathbf{A}_{2i}\}$, and to find proper quasi-cyclic patterns for each pair of sub-matrices to individually satisfy the symplectic inner product condition: $\mathbf{A}_{1i}\mathbf{A}_{2i}^T+\mathbf{A}_{2i}\mathbf{A}_{1i}^T=\mathbf{0}$. The application of this observation extends far beyond QC-LDPC codes. It actually provides a simple means of building new quantum stabilizer codes by concatenating existing quantum stabilizer codes. The new code will have a higher rate than any of the base code.

V. Simulation Results

There are several ways to characterize a quantum channel, some of which relate to classical channel models [6]. The simulation in this paper uses the depolarizing channel model, one of the most recognized model in quantum mechanics, in which X errors, Y errors, and Z errors occur independently with equal probability $f/3$. The total flip probability for a qubit is f, and the probability of experiencing no error is $(1-f)$.

We simulate a QC-LDPC code which has a codeword length of $N=180$ qubits, a code rate of 1/2, and a stabilizer generator (in its binary representation)

$$\mathbf{A}_1=\left[\mathbf{I}_0+\mathbf{I}_{61}\,\middle|\,\mathbf{I}_8+\mathbf{I}_{79}\right],\quad \mathbf{A}_2=\left[\mathbf{I}_1+\mathbf{I}_{60}\,\middle|\,\mathbf{I}_6+\mathbf{I}_{81}\right], \quad (16)$$

, where the identity matrix has a dimensionality of 90×90.

The sub-matrices $\mathbf{A_1}$ and $\mathbf{A_2}$ both have row weight 4 and column weight 2, and neither has length-4 cycles. The combined parity check matrix in GF(4) is regular, and has row weight 8 and column weight 4.

We measure word error rate and qubit error rate as a function of the total flip probability f. Figure 3 presents the simulation performance, and compares it with the quantum LDPC code presented in [8]. The quantum LDPC code in [8] is also a regular one, and has rate 1/2 and row weight 8, but its column weight is 8 and the code length is 8736 qubits. It is encouraging to see that our code performs remarkably better than the code in [8], even though our codeword length is significantly shorter (180 qubits versus 8736 qubits).

Although the sub-matrices in (16) do not have any length-4 cycle, concatenated together they produce length-4 cycles. It appears that length-4 cycles are unavoidable in stabilizer LDPC codes [8]. To mitigate the negative impact due to length-4 cycles, we suggest irregular constructions, or converting the quantum QC-LDPC codes to quantum LDPC-convolutional codes. Due to the space limitation, the simulations for these two improvements are not provided in this paper, but can be found in [16].

VI. Conclusion

We have developed simple and systematic ways to construct two rich classes of quantum stabilizer codes: quasi-cyclic LDPC codes and LDPC-convolutional codes. These codes, designed from the (general) symplectic inner product condition rather than the special CSS case, greatly enrich the family of stabilizer codes. With rates $R=(\tau-1)/\tau$ for $\tau=2,3,\cdots$, they produce a large set of rates, as well as rates much higher than those reported in literature.

We suggest building good quantum QC-LDPC codes based on good classical QC-LDPC codes. Making quantum QC-LDPC codes irregular can usually improve the performance over their regular counterparts. Transforming QC-LDPC codes to LDPC-convolutional codes further reduces short cycles and boosts the decoding efficiency. Simulation results confirm the strong error correction capability of the proposed codes.

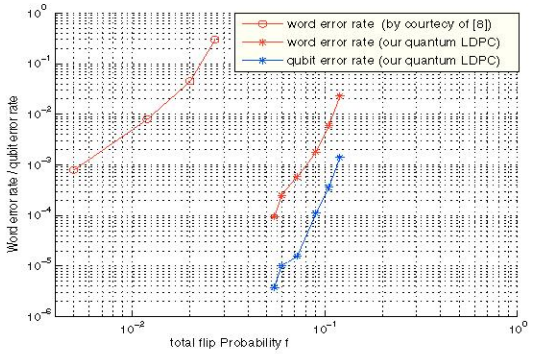

Fig. 3. Word error rate and qubit error rate of the quantum QC-LDPC code with length $N=180$ qubits, compared with a regular quantum LDPC code reported in [8].

References

[1] P. W. Shor, "Scheme for reducing decoherence in quantum computer memory,"*Phys. Rev. A*, vol. 52, pp. 2493-2496, 1995.

[2] A. R. Calderbank and P. W. Shor, "Good quantum error-correcting codes exist"; *Phys. Rev. A*, vol. 54, pp. 1098-1106, 1996.

[3] A. Steane, "Multiple particle interference and quantum error correction," *Proc. Roy. Soc. Lond. A*, 452, 2551, 1996

[4] A. R. Calderbank, E. M. Rains, P. W. Shor and N. J. A. Sloane; "Quantum error correction and orthogonal geometry," *Phys. rev. Lett.*, vol. 78, pp.405-408, 1997.

[5] M. S. Postol, "A proposed quantum low density parity check code," quant-ph/0108131.

[6] D. J. C. MacKay, G. Mitchison, and P. McFadden, "Sparse-graph codes for quantum error-correction", *IEEE Trans. on Inf. Theory*, vol.50, pp. 2315 - 2330, Oct. 2004.

[7] H. Lou and J. Garcia-Frias; "On the Application of Error-Correcting Codes with Low-Density Generator Matrix over Different Quantum Channels", *Proc. Int. Symp. Turbo Codes*, April 2006

[8] T. Camara, H. Ollivier and J.-P. Tillich, "Constructions and performance of classes of quantum LDPC codes," arXiv quant-ph/0502086.

[9] A. C. A. de Almeida and R. Palazzo, Jr., "A concatenated [(4, 1, 3)] quantum convolutional code", *Proc. IEEE Inf. Theory Workshop*, San Antonio, TX, Oct. 2004.

[10] G. D. Forney, Jr., M. Grassl, and S. Guha, "Convolutional and tail-biting quantum error-correcting codes," ArXiv quant-ph 0511016v1, Nov. 2005.

[11] M. Grassl and T. Beth, "Quantum BCH codes," arXiv quant-ph/9910060.

[12] M. Grassl and T. Beth, "Quantum Reed-Solomon Codes," arXiv quant-ph/9910059.

[13] A. Ashikhmin and E. Knill, "Non-binary quantum stabilizer codes," *IEEE Trans. Inf. Theory*, 2000

[14] A. J. Feltstrom and K. S. Zigangirov, "Time-varying periodic convolutional codes with low-density parity-check matrix," *IEEE Trans. Inf. Theory*, vol. 45, pp. 2181-2191, Sept. 1999.

[15] M. Shin, J. Kim, and H. Song, "Minimum distance bounds of irregular QC-LDPC codes and their applications," *IEEE ISIT*, Chicago, June 2004.

[16] P. Tan, and J. Li, "New classes of efficient quantum stabilizer codes," *in preparation*.