QUANTUM ERROR-CORRECTION USING CODES WITH LOW-DENSITY GENERATOR MATRIX

Hanqing Lou and Javier Garcia-Frias

Department of Electrical and Computer Engineering
University of Delaware
Newark, DE 19716
{lou, jgarcia}@ee.udel.edu

ABSTRACT

We propose the use of linear codes with low-density generator matrix in the context of quantum error correction. The proposed codes allow greater flexibility and are easier to design than existing sparse-graph quantum codes, while leading to better performance.

1. INTRODUCTION

Quantum error-correcting codes have drawn much interest during the last years. The first quantum code, introduced by Shor in 1995 [1], encoded one qubit into nine qubits and could correct both bit flip errors and phase flip errors. Shortly after, it was shown that quantum error-correcting codes can be constructed based on classical block codes [2]. This led to the development of an important class of quantum error-correcting codes by Calderbank, Shor and Steane [3, 4], known as CSS codes.

The construction of quantum error-correcting codes from classical ones makes it possible the application of turbo-like codes in quantum error correction. Decoding is performed by using iterative techniques, properly modified to take into account the quantum nature of the information. This was shown in [5], where practical quantum sparse-graph codes with iterative decoding were introduced for the first time. The codes in [5] are based on classical LDPC codes [6, 7], and easily outperform previous quantum codes, since large block lengths can be constructed. The quantum rate can be easily adjusted, but the special (dual-containing) quantum error-correcting structure used in [5] introduces some restrictions on the matrices of the LDPC codes, which leads to some performance degradation.

Motivated by [5], in this paper we propose quantum low-density generator matrix (LDGM) codes [8]. LDGM codes are a special class of LDPC codes with a systematic sparse generator matrix, and, therefore, they can be decoded with the same complexity as standard LDPC codes. The key idea is that in LDGM codes both the generator matrix and the parity check matrix can be easily generated, which provides much flexibility in the construction of quantum codes. Moreover, standard analysis techniques utilized in classical codes, such as density evolution and EXIT charts, can be adapted to this context.

This work was supported by the University of Delaware Research Foundation.

2. RELATIONSHIP BETWEEN QUANTUM ERROR-CORRECTING CODES AND CLASSICAL CODES

In this section, which is taken from [5], we provide a quick review of the notation utilized in this paper. A detailed explanation of these concepts can be found in [9].

2.1. Basic Concepts

Quantum error-correcting codes are based on qubits. A qubit represents a quantum state and is denoted as

$$|\psi >= \alpha_0|0 > +\alpha_1|1 >,$$

where α_0 and α_1 are complex numbers satisfying $|\alpha_0|^2 + |\alpha_1|^2 = 1$. A quantum state of n qubits has the form

$$\sum_{i=0}^{2^n-1} \alpha_i|i >,$$

where $\alpha_i \in \mathbb{C}$ and $\sum_i |\alpha_i|^2 = 1$.

Quantum error-correcting codes protect quantum states from error. An important challenge in quantum codes is that the quantum error can be continuous. Specifically, we can consider a quantum error as an arbitrary unitary linear operator that transfers a quantum state to a corrupted state. For one-qubit systems, the quantum error operator is a 2×2 complex unitary matrix. In this paper we consider three types of quantum errors: The bit flip error represented by matrix X, the phase flip error represented by matrix Z, and

the combination of bit and phase flips $Y = -iZX$. Together with the identity matrix, I, X, Y, and Z are the well known Pauli matrices:

$$I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

$$Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.$$

Any unitary operator acting on one qubit can be expressed as a linear combination of the Pauli matrices. Therefore, a code capable of correcting these three types of errors is able to correct any errors generated as linear combinations of these matrices. A Pauli operator on N qubits can be expressed as a sequence of N operators, each one of them being a Pauli matrix and acting on a different qubit.

2.2. Stabilizer Codes

A stabilizer group $\mathcal{S}$ is a set of Pauli operators on N qubits, so that the set is closed under multiplication and any two operators in the set commute (which occurs when disregarding the position where one of them is equal to I, they differ in an even number of positions). Obviously, it is enough to check the commutation property on a set of generators. Given the set of stabilizer generators $\{S_i\}$, a quantum codeword is defined as a state $|\psi>$ that is a $+1$ eigenstate of all the stabilizers (i.e., $S_i|\psi>=|\psi>$ for all i.)

The set of error operators $\{E_\alpha\}$ is a set of Pauli operators taking a quantum state $|\psi>$ to the corrupted state $E_\alpha|\psi>$. Since all of them are Pauli operators, a given error operator E_α either commutes or anticommutes with each stabilizer generator S_i. Therefore, $E_\alpha|\psi>$ is an eigenstate of S_i for all i. The syndrome is defined as the "commutation status" (either commute or non-commute) of $E_\alpha|\psi>$ with respect to all the stabilizers, and is completely determined by the commutation properties of E_α with the stabilizers and independent of the quantum state $|\psi>$.

Given any Pauli operator on N qubits, we can write it uniquely as a product of an X-containing operator (i.e., using only matrices X and I), a Z-containing operator, and a phase factor $(+1, -1, i \text{ or } -i)$. Then, we can express the X- (Z-) containing operator as a binary string of length N, with '1' standing for X (Z) and '0' for I. For instance,

$$\begin{aligned} XIYZYI &= -(XIXIXI) \times (IIZZZI) \\ &= (101010|001110). \end{aligned}$$

In this way, we can represent each stabilizer as a binary vector, and write the set of generators of $\mathcal{S}$ as a binary matrix $A = (H|G)$, where row i of H corresponds to the X-containing operator of stabilizer generator i, and row i of G is the binary representation of the Z-containing operator of stabilizer generator i. With this binary representation, the commutativity of stabilizers appears as orthogonality of the rows of G and H with respect to a twisted (or symplectic) product. In matrix representation, the twisted product property can be expressed as

$$GH^T + HG^T = 0. \quad (1)$$

A Pauli error operator E_α can be interpreted as a binary string e_α of length $2N$. By reversing the order of the X and Z strings in the error operator, the ordinary dot product (mod 2) of e_α with a row of the matrix A is 0 if E_α and the stabilizer represented by that row commute, and 1 otherwise. Thus, the quantum syndrome for the error operator E_α is exactly the classical syndrome Ae_α, where matrix $A = (H|G)$, called quantum parity check matrix, acts as the standard parity check matrix and e_α as binary error pattern. Therefore, we can conclude that from any binary matrix $(H|G)$ of size $M_Q \times 2N$ satisfying (1), it is possible to construct an equivalent quantum code that encodes $N - M_Q$ qubits in N qubits.

2.3. CSS Codes

An important class of stabilizer codes is that of Calderbank-Shor-Steane codes (CSS) [3, 4], which has the form

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

where $C(G)$ and $C(H)$ are classical linear codes such that $C(H)^\perp \subset C(G)$. Here, the notation $C(T)$, which we will use in the sequel, represents the classical code with parity check matrix T. Notice that this assumption guarantees that the commutativity condition defined by (1) is satisfied. The proposed quantum LDGM codes are based on this CSS structure.

3. QUANTUM LOW-DENSITY GENERATOR MATRIX CODES

3.1. Low-Density Generator Matrix Codes for Classical Channels

We focus on systematic LDGM codes, which are linear codes with sparse generator matrix, $[I\ P]$, with $P = [p_{lm}]$. The information message to be transmitted, $u = [u_1, \cdots, u_L]^T$, together with the coded (parity) bits, $c = [c_1, \cdots, c_M]^T$, generated as $c = Pu$, are transmitted through the channel. The corrupted sequence at the decoder is denoted as $[(u')\ (c')]$, where $c'_m = c_m + e^1_m$ and $u'_l = u_l + e^2_l$, with e^1_m and e^2_l being the error pattern (or noise) introduced by the channel. Notice that the code above is an $\frac{L}{L+M}$ rate systematic code. We will use the notation (X, Y) LDGM code

to indicate that the degrees of the systematic bit nodes and the parity nodes are X and Y, respectively.

For the case of LDPC codes, the decoder goal is to solve equation $s = He$, where s represents the syndrome calculated from the received sequence, H is the parity check matrix of the code, and e is the error pattern that we are interested in calculating. Since LDGM codes are a particular class of LDPC codes, they can be decoded in exactly the same way. However, we can look at LDGM decoding as a method to solve equation $c = Pu$, where c is the vector of parity bits generated at the encoder (i.e., before they are corrupted by the channel noise), P is the non-systematic part of the generator matrix, and u is the information message that we want to calculate. Then, the decoding algorithm for LDGM codes can be derived by applying belief propagation [10] (or factor graph decoding [11]) over the graph associated with equation $c = Pu$.

It is well known that LDGM codes are asymptotically bad. This was corroborated in [8, 12], where it was shown that the use of single LDGM codes always leads to error floors. However, a considerable error floor reduction can be achieved by using either a simple serial concatenation of two LDGM codes [8, 12] or a parallel scheme [13]. The resulting performance for concatenated LDGM codes over BSC and AWGN channels is, as shown in [8, 12, 13], comparable to that of irregular LDPC and turbo codes, with a very low encoding/decoding complexity.

3.2. Quantum LDGM Codes

In the dual-containing low-density parity-check codes (DC-LDPC) utilized in [5], the quantum parity check matrix has the form

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

with the following constraints:

- every row has weight k and every column has weight j.
- every pair of rows in H has an even overlap, and every row has even weight, which guarantees that the twisted product property is satisfied.

In order to satisfy these constraints, H has to be carefully designed, which constrains the matrix structure and leads to some performance degradation with respect to the H matrices utilized in standard LDPC codes.

The idea in this paper is to allow more degrees of freedom for the quantum parity check matrix A. In order to do so, we focus on systematic LDGM codes, since both the parity check matrix $\tilde{H}$ and the generator matrix $\tilde{G}$ of an LDGM code[1] are sparse and easy to build. More importantly, $\tilde{H}$ and $\tilde{G}$ are orthogonal by definition (i.e., $\tilde{G}\tilde{H}^{\perp} = \tilde{H}\tilde{G}^{\perp} = 0$).

The first intuition would be to directly use $\tilde{H}$ and $\tilde{G}$ in the CSS structure, which would result in

$$\tilde{A} = \left(\begin{array}{c|c} \tilde{H} & 0 \\ 0 & \tilde{G} \end{array} \right).$$

However, $\tilde{A}$ has size $N \times 2N$, and, therefore, it can not be used for encoding purposes, since the resulting quantum rate is 0. In order to get a valid quantum code, we need to reduce the number of rows, while keeping the CSS condition ($C(H)^{\perp} \subset C(G)$). The simplest way to do this is to puncture several rows of matrix $\tilde{G}$. However, the systematic structure of $\tilde{G}$ makes it impossible to decode successfully in this case, since puncturing $\tilde{G} = [I\ P]$ leads to a new matrix where some columns have degree 0.

The technique proposed in this paper to construct quantum codes from matrix $\tilde{A}$ is to perform linear row operations on both $\tilde{G}$ and $\tilde{H}$ to reduce the number of rows. In the sequel, we show that the matrices H and G resulting from the row operations satisfy the CSS condition ($C(H)^{\perp} \subset C(G)$), and, therefore, they can be used to build a quantum code with parity check matrix

$$A = \left(\begin{array}{c|c} \mathrm{H} & 0 \\ 0 & \mathrm{G} \end{array} \right).$$

The following theorem details this result.

Theorem. *Let $\tilde{H} = [P^T\ I]$ of size $n_1 \times N$ and $\tilde{G} = [I\ P]$ of size $n_2 \times N$ be the parity check matrix and the generator matrix for a given classical linear block code C, respectively (i.e., $n_1 + n_2 = N$). Define $H = M_1\tilde{H}$ and $G = M_2\tilde{G}$, where M_1 has size $m_1 \times n_1$ such that $m_1 < n_1$, and the size of M_2 is $m_2 \times n_2$ such that $m_2 < n_2$. The matrix*

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

is the parity check matrix of a quantum CSS code Q with rate $\frac{N-m_1-m_2}{N}$.

Proof: We only need to show that $C(H)^{\perp} \subset C(G)$. We make the proof in three stages[2]:

1. $C(H)^{\perp} \subset C(\tilde{H})^{\perp}$, since $\forall c \in C(H)^{\perp}, c = \sum_i a_i h_i = \sum_j b_j \tilde{h}_j \in C(\tilde{H})^{\perp}$. In the previous equality, h_i is the i^{th} row of matrix H and $\tilde{h}_j$ is the j^{th} row of matrix $\tilde{H}$. The equality holds because $H = M_1\tilde{H}$, and, therefore, any h_i is a linear combination of $\{\tilde{h}_j\}$. Since $m_1 < n_1$, $C(H)^{\perp} \subset C(\tilde{H})^{\perp}$.
2. $C(\tilde{H})^{\perp} = C(\tilde{G})$, since $\tilde{H}$ and $\tilde{G}$ are the parity check and generator matrices of a linear block code.

[1] In this paper, we always use $\tilde{H}$ and $\tilde{G}$ to denote the parity check and generator matrices of LDGM codes.

[2] As explained before, $C(T)$ represents the classical code defined by the parity check matrix T.

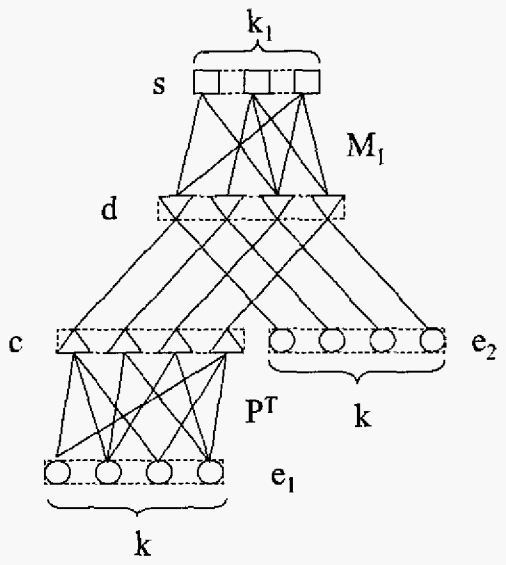

Fig. 1. Decoding graph for matrix H corresponding to the X-containing operators.

3. $C(\tilde{G}) \subset C(G)$. Let g_i ($\tilde{g}_i$) denote the i^{th} row of matrix G ($\tilde{G}$). Assume $c \in C(\tilde{G})$. Then, $c \cdot \tilde{g}_j = 0 \, \forall j$. Since $\forall i$, $g_i = \sum_j a_{ij} \tilde{g}_j$ (for some $a_{ij} \in \mathbb{Z}\{0,1\}$), $c \cdot g_i = c \cdot \sum_j a_{ij} \tilde{g}_j = 0 \, \forall i$, which implies $c \in C(G)$. Since $m_2 < n_2$, $C(\tilde{G}) \subset C(G)$.

3.3. Building Quantum LDGM Codes

Decoding is performed by applying the belief propagation algorithm on the graph defined by matrix A. Notice that in A, decoding for the H and G matrices can be performed independently. Fig. 1 represents the decoding graph for matrix H (X-containing operators), but a similar process is performed for matrix G (Z-containing operators). We denote the syndrome as s, which is related to the error pattern by $s = He = M_1 \tilde{H} e = M_1 [P^T \; I] e$. In order to construct the graph in Fig. 1, we split the error pattern e in two parts e_1 and e_2, and relate it to the syndrome in a two step process:

$$d = [P^T \; I] e = [P^T \; I] \begin{pmatrix} e_1 \\ e_2 \end{pmatrix} = P^T e_1 + e_2 \quad (2)$$

$$s = M_1 d. \quad (3)$$

We apply the belief propagation algorithm over the graph defined in Fig. 1. Notice that the syndrome s is known exactly from the quantum decoding circuit, and the *a priori* probability of the error pattern e can be calculated according to the quantum channel. However, there is no information about the middle level of the graph. In other words, we do not have any *a priori* information about variables c or d, which introduces some difficulties in the first decoding iterations. In order to overcome this problem, we utilize the *doping* [14] technique in matrix M_1. That is, we introduce some degree-1 syndrome nodes, which propagate correct information to some nodes of d, and push the iterative decoding in the right direction. Notice that matrix M_1 is irregular, although in the simulation results shown in the sequel we choose the degree of the other nodes of M_1 and matrix P in a regular way. Further performance gains can be expected if the degrees are optimized utilizing density evolution or EXIT chart techniques.

It is important to mention that decoding based on the graph presented in Fig. 1 is different from decoding based on the final matrix H. The reason is that the product of M_1 and $\tilde{H}$ eliminates some edges and introduces more cycles. Notice that the decoding structure based on Fig. 1 is somehow similar to the serial concatenated LDGM scheme, which reduces the error floor of a single LDGM code. Simulation results will show that the proposed quantum scheme has also very low error floors.

4. SIMULATIONS

In order to simplify the code design, we utilize $M_1 = M_2 = M$. For comparison purposes, we consider quantum codes with the same rate and block length as [5]: $R_Q \cong 1/4$ and block length $N = 19014$, which encode $k_1 = 7131$ qubits in N qubits. Matrix P corresponds to a $1/2$ LDGM code and has size 9507×9507, so that both $\tilde{G}$ and $\tilde{H}$ have size (9507×19014). The matrix M, with size (7131×9507), is used to transform $\tilde{H}$ and $\tilde{G}$ into H and G. The channel model used in the simulations corresponds to two binary symmetric channels, so that the errors in the binary representation of the X- and Z-containing operators are i.i.d. with a flip probability f_m.

Fig. 2 shows the results obtained utilizing matrices P and M with different degrees. The *y-axis* represents the block error rate, while the *x-axis* is the flip probability f_m. As explained before, matrix M is designed using the doping technique [14]. Specifically, we fixed the degree of the $k = 9507$ nodes d to 3, and, for the $k_1 = 7131$ syndrome nodes, we chose p syndrome nodes with degree 1 and the rest $k_1 - p$ nodes with degree x, so that equation $(k_1 - p)x + p = 3k$ holds. In our simulations we tried $x \in \{6, 8, 10\}$, and denote the degrees of matrix M as $(3; 1, x)$. Notice that if the number of syndrome nodes, p, with degree 1 increases, we propagate exact information from those p nodes to nodes d in the first iteration. However, since the number of edges is a constant, we also increase the degree of the other $k_1 - p$ syndrome nodes, which means that information coming from these nodes is less reliable. The choice of x is a tradeoff between these two effects. Fig. 2

shows that for the matrices P utilized in the simulation, the matrix M $(3;1,8)$ behaves better than $(3;1,6)$. However, further increases to $(3;1,10)$ lead to a performance degradation (not shown in the figure).

We also investigated the effect of matrix P when M is kept fixed. For simplicity, we just consider regular matrices P with size $k \times k$ and degrees (y, y). Fig. 2 shows a performance improvement when y increases from 10 to 14. However, a slight performance degradation is observed when y is increased to 15.

The results presented here are not exhaustive, and further performance gains are expected by carefully choosing the parameters of the proposed scheme. It is interesting, however, to note that, even without careful optimization, the resulting performance compares favorably with that of [5].

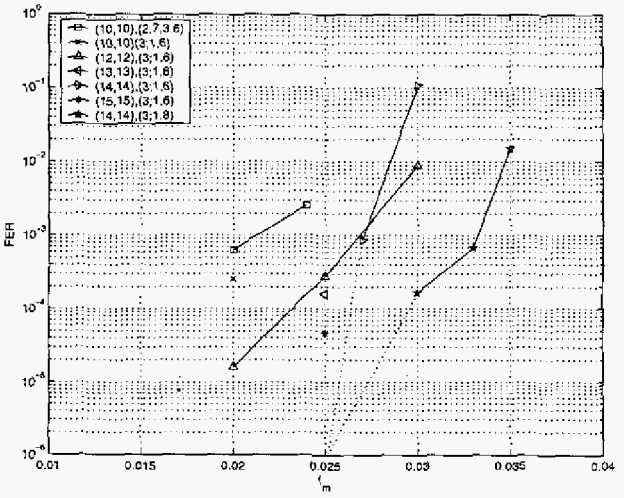

Fig. 2. Simulation results for the family of quantum codes defined in Section 4. The first and second tuple indicate the degrees of the P and M matrices, respectively. After simulating more than $60,000$ blocks, no errors where observed at $f_m = .025$ for codes $[(14,14),(3;1,6)]$ and $[(14,14),(3;1,8)]$.

5. CONCLUSION

We introduce quantum codes based on LDGM codes. Similar to quantum (dual-containing) LDPC codes, there is a high degree of flexibility in the choice of rate and block length. However, the design of the quantum parity check matrix presents less constraints than in the dual-containing LDPC case, which leads to a better performance even without exhaustive optimization of the system parameters.

6. REFERENCES

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

[2] Q.R. Calderbank, E.M. Rains, P.W. Shor and N.J.A. Sloane "Quantum error correction via codes over GF(4)," *IEEE Trans. on Information Theory*, vol. 44, no. 4, pp.1369-1387, July 1998.

[3] A.R. Calderbank and P.W. Shor, "Good quantum error-correcting codes exist," *Phys. Rev. A*, vol. 54, no. 2, pp.1098-1105, August 1996.

[4] A. Steane, "Multiple particle interference and quantum error correction," *Proc. Roy. Soc. Lond. A*, vol. 452, pp. 2551-2577, 1996.

[5] D.J.C. MacKay, G. Mitchison and P.L. McFadden, "Sparse-graph codes for quantum error-correction," *IEEE Trans. on Information Theory*, vol. 50, no. 10, pp.2315-2330, October 2004.

[6] R. G. Gallager, "Low-density parity-check codes," *IEEE Trans. on Information Theory*, vol.8, no.1, pp. 21-28, January 1962.

[7] D. J. C. MacKay, "Good error-correcting codes based on very sparse matrices," *IEEE Trans. on Information Theory*, vol. 45, no.2, pp. 399-431, March 1999.

[8] J. Garcia-Frias and W. Zhong, "Approaching near shannon performance by iterative decoding of linear codes with low-density generator matrix," *IEEE Communications Letters*, vol. 7, no.6, pp. 266-268, June 2003.

[9] M.A. Nielsen and I.L. Chuang, "Quantum computation and quantum information," Cambridge University Press, 2000.

[10] J. Pearl, "Probabilistic reasoning in intelligent systems: networks of plausible inference," Morgan Kaufmann, 1988.

[11] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. on Information Theory*, vol. 47, no.2, pp. 498-519, February 2001.

[12] J. Garcia-Frias, W. Zhong and Y. Zhao, "Iterative decoding schemes for source and channel coding of correlated sources," *Asilomar Conference on Signals, Systems, and Computers*, November 2002.

[13] H. Chai, W. Zhong, J. Garcia-Frias, "Parallel concatenation of LDGM codes to approach capacity limits," *Proc. CISS'05*, March 2005.

[14] S. ten Brink, "Code doping for triggering iterative decoding convergence," *Proc. ISIT'01*, June 2001.