Quantum Error Correction Codes - From Qubit to Qudit

Xiaoyi Tang Paul McGuirk

December 7, 2005

1 Introduction

Quantum computation (QC), with inherent parallelism from the superposition principle of quantum mechanics, promises vast performance improvement over classical computing. QC has become the subject of increasing research interests since Shor showed an efficient algorithm for integer factorization in a quantum system providing exponential speed-up compared with any known classical algorithm.

As in any system, errors are inevitable in QC. Errors can arise from entanglement with the environment causing decoherence as a real quantum system cannot be completely isolated. Quantum gates, building blocks of QC, can also introduce errors. These are unitary transformations operating on a continuous parameter space, unlike classical digital computers. As a result, perfect accuracy is required for correct operation. For example, an X gate implemented in a spin resonance system requires perfect timing. Such strict accuracy requirements make quantum gates prone to errors. Errors, if left uncorrected, propagate through the computation path and render QC unreliable.

One solution to the problem is to apply quantum error correction codes (QECC), similar to classical error correction in communication and storage systems. Various types of QECC have been studied. The focus of our project is on *stabilizer* codes [1, 2].

While a qubit represents a two dimensional quantum system, a qudit is generalization to higher a d dimensional quantum system. Aside from theoretical interests, higher dimensions allow more efficient systems. The second part of the project is to find out how easily QECC for qubits can be extended to qudits.

2 Quantum Error Correction Codes

Like classical error correction codes, QECC encode k qubits into n qubits where $n > k$ so that the extra $n - k$ qubits serve as redundancy to fight errors. One

of the simplest codes is the 3 qubit bit flip code defined as

$$\begin{aligned} |\,0\rangle \rightarrow |\,0_L\rangle &\equiv |\,000\rangle \\ |\,1\rangle \rightarrow |\,1_L\rangle &\equiv |\,111\rangle \end{aligned} \tag{1}$$

It's similar to classical repetition codes. However, only basis states are cloned because of the no-cloning theorem. Therefore, an arbitrary qubit state $|\phi\rangle = \alpha|0\rangle + \beta|1\rangle$ is encoded into $\alpha|0_L\rangle + \beta|1_L\rangle$. The code can correct up to 1 bit flip error, i.e., Pauli X error operator on 1 qubit. Decoding is done by measuring Z_1Z_2 and Z_2Z_3 where Z_i is Pauli Z operator on the i^{th} qubit. Measurement generally destroys quantum state if it is not an eigen state. In this case, no quantum information is destroyed because the encoded state after an X error operator remains as an eigen state of the two measurements. For example, if the error is X_1, then the state becomes $\alpha|100\rangle + \beta|011\rangle$, which is an eigen state of Z_1Z_2 with eigen value -1 and an eigen state of Z_2Z_3 with eigen value 1. The measurement results identify possible 1 qubit bit flip error and required unitary operation to correct the error as listed in Table 1.

Table 1: Decoding for 3 qubit bit flip code [1].

Z_1Z_2	Z_2Z_3	Error type	Action
+1	+1	no error	no action
+1	-1	bit 3 flipped	X_3
-1	+1	bit 1 flipped	X_1
-1	-1	bit 2 flipped	X_2

A phase flip error (Pauli Z) takes state $\alpha|0\rangle + \beta|1\rangle$ to $\alpha|0\rangle - \beta|1\rangle$. It turns out to be a bit flip error in the $|+\rangle$ and $|-\rangle$ basis because it transforms $\frac{1}{\sqrt{2}}(\alpha+\beta)|+\rangle + \frac{1}{\sqrt{2}}(\alpha-\beta)|-\rangle$ to $\frac{1}{\sqrt{2}}(\alpha+\beta)|-\rangle + \frac{1}{\sqrt{2}}(\alpha-\beta)|+\rangle$. The relationship can be also understood by noting $HXH = Z$. Therefore the same 3 qubit bit flip code can be used to correct 1 phase flip error after changing basis by a Hadamard gate.

One of the earliest QECC developed is the Shore code. It concatenates the bit flip code and the phase flip code to form a 9 qubit code and can correct an arbitrary error on a single qubit. The codewords are

$$\begin{aligned} |\,0\rangle \rightarrow |\,0_L\rangle &\equiv \frac{(|000\rangle + |111\rangle)(|000\rangle + |111\rangle)(|000\rangle + |111\rangle)}{2\sqrt{2}} \\ |\,1\rangle \rightarrow |\,1_L\rangle &\equiv \frac{(|000\rangle - |111\rangle)(|000\rangle - |111\rangle)(|000\rangle - |111\rangle)}{2\sqrt{2}} \end{aligned} \tag{2}$$

2.1 Stabilizer Codes

Stabilizer formalism provides a group theoretical framework to analyze quantum error correction. The group under study is called *Pauli Group* P_n for n qubits.

It's a n-fold tensor of P_1 which is defined as

$$P_1 = \{\pm I, \pm iI, \pm X, \pm iX, \pm Y, \pm iY, \pm Z, \pm iZ\}$$

Because the set of operators I, X, Y, Z either commute or anti-commute, any two operators in P_n either commute or anti-commute as well. A state $|\phi>$ is said to be stabilized by an operator U if $U|\phi>= |\phi>$, i.e., $|\phi>$ is an eigen state of U with eigen value 1. Let S be a commutative (Abelian) subgroup of P_n generated by $n-k$ independent generators $g_1, g_2, \ldots, g_{n-k}$. Let V_S be the subspace of states stabilized by every element in S. It can be proved that V_S has dimension 2^k. The stabilizer code $C(S)$ is simply the basis of V_S.

Error is detected and corrected by first measuring $g_1, g_2, \ldots, g_{n-k}$. An error operator E from P_n still keeps the corrupted codeword an eigen state of all the generators. To see this, let $|\phi>$ be a codeword. Because it's stabilized by S,

$$g_i|\phi>= |\phi> \quad i = 1, \ldots, n-k$$

It follows

$$g_i E|\phi>= \pm E g_i|\phi>= \pm E|\phi> \quad i = 1, \ldots, n-k$$

Therefore, measurement of the generators does not collapse the quantum state. Any E in P_n that anti-commutes with at least one generator can be detected because measurement result gives -1 while +1 is expected if there is no error. If E commutes with all generators and does not belong to S, then the error cannot be detected. Minimum distance d of a stabilizer code is defined as the minimum weight of such un-detectable operators, where the weight is the number of non-I operations. Similar to classical error correction, the number of qubit errors t a code can fix is $\lfloor \frac{d-1}{2} \rfloor$.

Examples of stabilizer codes are abundant. The 3 qubit bit flip code has stabilizer $\{Z_1Z_2, Z_2Z_3\}$. The 5 qubit code [5,1,3] has an S generated by

$$\{XZZXI, IXZZX, XIXZZ, ZXIXZ\}$$

3 Qudits

Quantum computation is usually formulated in terms of a qubit, a quantum state in a two dimensional Hilbert space. Several authors have considered the advantages of a higher dimensional Hilbert space.

3.1 d-Nary Arithmetic

Nearly all of classical computation uses a binary (base 2) representation of numbers. A non-negative integer n can be written as a sum of powers of two:

$$n = \sum_{j=0}^{\infty} b_j 2^j \tag{3}$$

where $\forall j\ b_j \in \{0,1\}$. The binary representation of n, $[[n]]_2$, is the string $b_{N_2} b_{N_2-1} \dots b_0$ where $N_2 = \lfloor \log_2(n) \rfloor$, the greatest integer less than $\log_2(n)$ (since n is finite, $b_j = 0\ \forall j > N$). Each member of the string is called a bit. The basic operations that we can perform on each bit are addition and multiplication modulo 2, and from these we can construct basic arithmetic operations on n. Binary strings can also be used to represent negative integers, rational numbers, and approximations to real numbers, allowing for more general computation.

We can generalize this construction for any natural number d greater than 1:

$$n = \sum_{j=0}^{\infty} p_j d^j \tag{4}$$

where $\forall\ j\ b_j \in \mathbb{Z}_d = \{0, 1, \cdots, d-1\}$. The d-nary representation of n is defined in the same manner as the binary represnetation, $[[n]]_d = p_{N_d} p_{N_d-1} \dots p_0$ where $N_d = \lfloor \log_d(n) \rfloor$. Each member of the string is called a *dit* (the special names trit and digit are given for $d = 3$ and $d = 10$ respectively). The basic ditwise operations are, as is the case for $d = 2$, addition modulo d and multiplication. Algorithms for more complex arithmetic can be built from these operations.

In a theoretical sense, d-nary representations of numbers are more efficient than binary representations. If $d > 2$, then fewer dits are required to represent a number n. For a string of N dits, there are d^N possible numbers that can be represented. Since this is the number of available states, this can be considered the number of degrees of freedom. The relative increase in the number of degrees of freedom when compared to $d = 2$ is

$$\Delta_c(N, d) = \frac{d^N - 2^N}{2^N} \sim \left(\frac{d}{2}\right)^N \tag{5}$$

There are some advantages and some difficulties associated with the implementation a d-nary computer [3]

3.2 Qudit Gates

A *qudit* is any quantum state in a d-dimensional Hilbert space. For a particular Hilbert space, we choose a particular orthonormal basis called the computational basis

$$\mathcal{B}_d = \{|0\rangle, |1\rangle, \dots, |d-1\rangle\} \tag{6}$$

An arbitrary qudit in this system is given by

$$|\psi\rangle = \sum_{i=0}^{d-1} \alpha_i |i\rangle \tag{7}$$

where $\alpha_i \in \mathbb{C}$. In terms of matrix representations, $|\psi\rangle$ is a vector in $\mathbb{C}^d$. If $d = 3$, the qudit is called a qutrit. Examples of qutrit systems are similiar to qubits.

One might construct a qutrit system by considering different energy levels of a particular atom, or the orientation of a massive spin-1 boson in a magnetic field. Much recent work has focused on the prospect trapped-ion implementations [4], [5]

The Pauli group can be generalized for a qudit system. There are several methods of doing this [6], [7]. Following the perscription of Gottesman [8] we define the operators X and Z by their action on the computational basis

$$\begin{aligned} X\left|j\right\rangle &= \left|j+1\right\rangle \\ Z\left|j\right\rangle &= \omega^{j}\left|j\right\rangle \end{aligned} \tag{8}$$

where $\omega = e^{2\pi i/d}$ and addition is modulo d. For $d = 2$, these reduce to the Pauli matricies σ_x and σ_z respectively. For $d = 3$, for example, the operators have the matrix representation (in the computational basis)

$$\begin{aligned} X &= \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \\ Z &= \begin{pmatrix} 1 & 0 & 0 \\ 0 & e^{2\pi i/3} & 0 \\ 0 & 0 & e^{4\pi i/3} \end{pmatrix} \end{aligned} \tag{9}$$

These operators are generators of the Pauli group for a d-dimensional Hilbert space. We define

$$U_{r,s}^{d} = X^{r}Z^{s} \tag{10}$$

The Pauli group is defined by

$$\mathcal{P}_d = \left\{U_{r,s}^{d} | r, s \in \mathbb{Z}_d\right\} \tag{11}$$

$\mathcal{P}_d$ forms a basis for the group of unitary $d \times d$ matricies $U(d)$ [6]. The set is orthonormal in the sense that

$$\mathrm{tr}\left(\left(U_{r,s}^{d}\right)^{\dagger} U_{r',s'}^{d}\right) = d\delta_{r,r'}\delta_{s,s'} \tag{12}$$

Since $\mathcal{P}_d$ spans $U(d)$, any singe qudit quantum gate can be expressed in terms of these operators. The construction of multiple qudti gates are discussed in [9] and [10].

The use of qudits presents several advantages over the use of qudits because of the richer possibilities for entanglement of multi qudit systems (the basic principles for qutrit entanglement are described in[11]). This allows, for example, for more secure quantum communication [12]. Qubit systems also have the advantage of being much more robust against noise and other errors [13].

3.3 Qudit Stabilizer Codes

The formalism used to produce stabilizer codes can be generalized to qudits. For prime d, this process is relatively straightforward [8]. The stabilizer S of a code for qudits of dimension d is an Abelian subgroup of $\mathcal{P}_d$. If we choose the subgroup such that there are $n-k$ generators, then, for prime d, the codespace has dimension d^{n-k}. This allows us to encode k qudits in a block of n qudits. (If d is not prime, then it is also necessary that the elements of S be of order d.)

As with the qubit case, we need to find a set of operators that commute with all of the members of S but are not in S. As discussed in [8], the set of operators that commute with S n members. We choose from $U(d)$ k operators that commute with S and call them $\overline{Z}_1, \cdots \overline{Z}_k$. These are the action of Z on the encoded qudits. We then choose k more operators $\overline{X}_1, \cdots \overline{X}_k$ such that

$$\begin{aligned} &\overline{X}_i \overline{Z}_j = \overline{Z}_j \overline{X}_i \, (i \neq j) \\ &\overline{X}_i \overline{Z}_i = \omega^{-1} \overline{Z}_i \overline{X}_i \\ &\overline{X}_i M = M \overline{X}_i \, (\forall M \in \mathcal{P}) \end{aligned} \tag{13}$$

The group generated by the generators of S and the encoded X and Z is the subgroup of $\mathcal{P}_d$ that S. As is the case for qubits, the members of this set that are not in S perform the encoded operations on the data.

When d is prime, the codes often take the same form as they do for qubits. The 3 qubit flip code, for example, corresponds to the stabilizer

$$S = \left\{ Z_1 \left(Z_2\right)^{-1}, Z_2 \left(Z_3\right)^{-1} \right\} \tag{14}$$

The set $\{|jjj\rangle\}$ is stabilized by this code. Similarly, the $[[5,1,3]]$ code is based on the set

$$S = \{XZZXI, IXZZX, XIXZZ, ZXIXZ\} \tag{15}$$

Both of these examples have the same form as the corresponding qubit codes (for $d = 2$, each member of the Pauli group has order 2, and so is its own inverse).

4 Conclusion

In this project, we study the use of QECC in quantum systems, from 2 dimensions to higher dimensions. In particular, we look at the stabilizer framework for QECC analysis. Extension of stabilizer codes from qubits to qudits is easy when d is prime.

Xiaoyi Tang wrote Sections 1 and 2. Paul McGuirk wrote Section 3.

References

[1] M. Nielsen and I. Chuang, *Quantum Computation and Quantum Information* (2000)

[2] J. Preskill, *Lecture Notes Chapter 7*

[3] Brousentsov N. P., et al. Development of ternary computers at Moscow State University (http://www.computer-museum.ru/english/setun.htm). Retrivied December 6, 2005 (Brousentsov et al constructed a ternary computer at Moscow State University in the 1950's and 60's. Most of the relevant literature is in Russian, but the authors have provided a summary of their work in English at this location).

[4] Mc Hugh, D, and Tawmley, J. Trapped-ion qutrit spin molecule quantum computer. *New Journal of Physics* **7** (2005) 174

[5] Klimv, A.B., et al. Qutrit quantum computer with trapped ions. *Phys. Rev. A* **67** (2003) 062313

[6] Knill, E. Non-binary Unitary Bases and Quantum Codes. quant-ph/9608048

[7] Knill, E. Group Representations, Error Bases, and Quantum Codes. quant-ph/9608049

[8] Gottesman, D. Fault-Tolerant Quantum Computation with Higher-Dimensional Systems. quant-ph/9802007

[9] Muthukrishnan, A., and Stroud, C.R., Jr. Multivalued logic gates for quantum computation. *Phys. Rev. A.* **62** (2000) 052309

[10] Brennen, G.K., O'Leary, D.P., and Bullock, Stephen S. Criteria for Exact Qudit Universality. *Phys. Rev. A.* **71** (2005) 052318

[11] Caves, C.M., and Milburn, G.J. Qutrit Entanglement. *Opt. Commun.* **179** (2000) 439

[12] Cerf, N.J., et al. Security of Quantum Key Distribution Using d-Level Systems. *Phys. Rev. Lett* **88** (2002) 127902

[13] Aharonov, D., and Ben-Or, M. Fault-Tolerant Quantum Computation with Constant Error Rate quant-pu/9906129