Open Quantum Systems and Quantum Error Correction

Thesis

Submitted in partial fulfillment of the requirements of
BITS F422 Thesis

Soorya Rethinasamy

f2015350@pilani.bits-pilani.ac.in

Under the supervision of

Abhishek Mishra

Assistant Professor, Birla Institute of Technology and Science, Pilani

Birla Institute Of Technology And Science Pilani, Pilani Campus

June 2020

Acknowledgements

The author would like to thank Dr. Abhishek Mishra for his granting me the opportunity to work under him on this thesis. Despite the problem statement being out of this field of expertise, he was always willing to provide much needed guidance and support. He allowed a great deal of freedom to explore the terrain of the field. Furthermore, the author is grateful to the Computer Science Department, BITS Pilani and Birla Institute of Technology and Science, Pilani for this opportunity.

The quantum circuit diagrams in this manuscript are created using the quantikz package [Kay18].

Certificate

This is to certify that the thesis entitled, **Open Quantum Systems and Quantum Error Correction**, and submitted by **Soorya Rethinasamy**, ID No. **2015B5A70350P** in partial fulfillment of the requirement of BITS F422T Thesis, embodies the work done by him under my supervision.

Date :

Abhishek Mishra
Assistant Professor Professor
Birla Institute of Technology and Science, Pilani

Abstract

In this review, we look at several salient features of open quantum systems. We introduce various noise models, on multi-qubit and single-qubit systems. We look at several important examples of noisy channels. Classical error-correction is introduced as a precuror to quantum error-correction.

We then look at the three qubit bit flip channel and phase-flip channel. We explore the intricacies of the Shor code, capable of correcting arbitrary errors on a single qubit. In the next section, we begin the development of a theoretical framework for quantum error-correction. We look at the development of a class of quantum codes called CSS codes, firmly based on classical linear codes. Finally, we delve into the stabilizer formalism and create codes using this formalism.

Contents

Chapter 1

Introduction

Quantum computers work with qubits, unlike traditional classical computers that work with bits. It has been shown, both theoretically and practically, that, through precise control of quantum phenomena, quantum computers can outperform classical computers. Several notable examples include search algorithms, integer prime factorization, optimization, and quantum simulation.

Classical computers are built using transistors, that act as logical *ON/OFF* switches, and classical logic gates. Compared to the size of the qubit implementation, these transistors are a million times larger and are very easy to control. This control allows an almost complete removal of errors at the physical level. For quantum computers however, the qubit systems are much more error-prone and thus require a more active error correction. This requirement translates to a more involved algorithm design process, and the need for more sophisticated hardware. The field of quantum error correction evolved as an answer to the question, *What quantum operations can we reliably perform using unreliable underlying hardware?*

This thesis is divided into two important chapters. The first chapter will dive into open quantum systems, and how they can be modeled. The second chapter will talk about quantum error correction, several examples and a more robust theoretical base model.

Chapter 2

Open Quantum Systems

A introduction to the postulates of Quantum mechanics can be found in several texts and articles. However, the approach here is based on the approach of the excellent textbook by Nielsen and Chuang [NC11].

Postulate 3 of Quantum Mechanics tells us that a closed quantum system undergoes a unitary evolution. Closed systems have no interaction with the external world. However, no real system, besides the universe, is truly closed. Thus, any real system must interact with the environment and these interactions show up as noise. For example, if the state of a qubit is represented by two positions of a electron, then that electron will interact with other charged particles, which act as a source of uncontrolled noise affecting the state of the qubit. An open system is nothing more than one which has interactions with some other environment system, whose dynamics we wish to neglect, or average over.

We begin this chapter with the description of the quantum operations formalism. This formalism helps us exactly describe this quantum noise. Then, we shall look at the Kraus Operator Sum Representation. Lastly, we look at the open evolution of a qubit.

2.1 The System and the Environment

Consider a combined system made of 2 systems - a system of interest, called A, and the environment E. We assume that the combined system is closed and undergoes a unitary evolution, obeying Schrödinger equation [Lid19].

We pick two orthonormal bases for the two Hilbert spaces as

$$\begin{aligned} \mathcal{H}_A &= \text{span}\left(|i\rangle_A\right) \\ \mathcal{H}_E &= \text{span}\left(|\mu\rangle_E\right). \end{aligned} \tag{2.1}$$

By the second postulate, the Hilbert space of the two system combined is the tensor product of the individual spaces:

$$\begin{aligned} \mathcal{H} &= \mathcal{H}_A \otimes \mathcal{H}_E \\ &= \text{span}\left(|i\rangle_A \otimes |\mu\rangle_E\right). \end{aligned} \tag{2.2}$$

Thus, any state $|\psi\rangle \in \mathcal{H}$ can be expressed as a linear combination of the new basis vectors. Consider a pure state ensemble $\{|\psi_a\rangle, p_a\}$. Each state of the ensemble can be expressed as follows:

$$|\psi_a\rangle = \sum_{i,\mu} c_{a;i,\mu} |i\rangle_A \otimes |\mu\rangle_E . \tag{2.3}$$

Thus, the associated density matrix for this ensemble can be expressed as

$$\begin{aligned}\rho &= \sum_a p_a |\psi_a\rangle\langle\psi_a| \\ &= \sum_a p_a (\sum_{i,\mu} c_{a;i,\mu} |i\rangle_A \otimes |\mu\rangle_E)(\sum_{j,\nu} c^*_{a;j,\nu} \langle j|_A \otimes \langle\nu|_E) \\ &= \sum_{ij\mu\nu} \lambda_{ij\mu\nu} |i\rangle\langle j|_A \otimes |\mu\rangle\langle\nu|_E, \end{aligned} \tag{2.4}$$

where $\lambda_{ij\mu\nu} = \sum_a p_a c_{a;i,\mu} c^*_{a;j,\nu}$.

Since we are only interested in system A, we need to discard the environment E. We now define a new operation called partial trace, which effectively averages out of the components of B from the combined density matrix. The resultant density matrix then describes only A.

2.1.1 Partial Trace

The partial trace is a linear operator mapping operators from the total Hilbert space to the Hilbert space of A, i.e., $\mathcal{H} \mapsto \mathcal{H}_A$.

Consider an operator $O = X_A \otimes Y_E$. The partial trace with respect to system E is then defined as,

$$\begin{aligned}\mathrm{Tr}_E(X_A \otimes Y_E) &\equiv X_A \ \mathrm{Tr}(Y_E) \\ &= X_A \ \sum_\mu \langle\mu| Y_E |\mu\rangle \\ &= \sum_\mu \langle\mu| X_A \otimes Y_E |\mu\rangle . \end{aligned} \tag{2.5}$$

Using linearity, if $O = \sum_{ij} c_{ij} X^i_A \otimes Y^j_E$, then by linearity:

$$\begin{aligned}\mathrm{Tr}_E[O] &= \sum_{ij} c_{ij} \, \mathrm{Tr}_E(X^i_A \otimes Y^j_E) \\ &= \sum_{ij} c_{ij} X^i_A \sum_\mu \langle\mu| Y^j_E |\mu\rangle \\ &= \sum_\mu \sum_{ij} c_{ij} \langle\mu| X^i_A \otimes Y^j_E |\mu\rangle \\ &= \sum_\mu \langle\mu| O |\mu\rangle . \end{aligned} \tag{2.6}$$

When applied to the definiton of ρ from (2.4):

$$\mathrm{Tr}_E[|i\rangle\langle j|_A \otimes |\mu\rangle\langle\nu|_E] \equiv |i\rangle\langle j|_A \langle\mu|\nu\rangle_E. \tag{2.7}$$

By linearity,

$$\begin{aligned}\mathrm{Tr}_E[\sum_{ij\mu\nu}\lambda_{ij\mu\nu}|i\rangle\langle j|_A\otimes|\mu\rangle\langle\nu|_E] &= \sum_{ij\mu\nu}\lambda_{ij\mu\nu}|i\rangle\langle j|_A\langle\nu|\mu\rangle_E \\ &= \sum_{ij\mu}\lambda_{ij\mu\mu}|i\rangle\langle j|_A \\ &= \sum_{ij}\bar{\lambda}_{ij}|i\rangle\langle j|_A, \end{aligned} \tag{2.8}$$

where $\bar{\lambda}_{ij} = \sum_\mu \lambda_{ij\mu\mu}$. We note that the output looks like a state purely on subsystem A. It can be shown that the output satisfies all the required for a density matrix. Furthermore, it can be shown that the partial trace is the only operation that produces a **reduced density matrix** that obeys the measurement postulate for each subsystem. Here, without proof (proof can be found in [NC11]), the state of the subsystem A can be given as:

$$\rho_A = \mathrm{Tr}_E[\rho]. \tag{2.9}$$

2.2 Open System Dynamics

2.2.1 Kraus Operator Sum Representation

Consider a system S and environment E that are initially in a state $\rho(0)$. Since the combined system is closed, they undergo a unitary evolution $U(t) = e^{-iHt}$. Then, by Schrödinger's equation,

$$\rho(t) = U(t)\rho(0)U^\dagger(t). \tag{2.10}$$

The reduced density matrix for the environment can be expanded using an orthonormal basis as follows (Using the Spectral Decomposition Theorem [NC11]):

$$\rho_E(0) = \sum_\nu \lambda_\nu|\nu\rangle\langle\nu|, \tag{2.11}$$

where λ_ν are the eigenvalues (probabilities) and $\{|\nu\rangle\}$ are the corresponding orthonormal eigenvectors.

The state of the system is then found by performing a partial trace over the environment, i.e.,

$$\rho_S(t) = \mathrm{Tr}_E[\rho(t)]. \tag{2.12}$$

Since the trace is basis-independent, we pick the orthonormal eigenvectors of the environment, i.e.,

$$\begin{aligned}\rho_S(t) &= \mathrm{Tr}_E[U(t)\rho(0)U^\dagger(t)] \\ &= \sum_\mu \langle\mu|\, U(t)\rho(0)U^\dagger(t)\,|\mu\rangle\,. \end{aligned} \tag{2.13}$$

Here, we make a crucial assumption. Let the initial state be uncorrelated, i.e., exist as a product of density matrices.

$$\rho(0) = \rho_S(0) \otimes \rho_E(0). \tag{2.14}$$

Thus, using the definition of $\rho_E(0)$,

$$\begin{aligned}
\rho_S(t) &= \sum_{\mu} \langle\mu| \left[U(t)\rho_S(0) \otimes \sum_{\nu} \lambda_\nu |\nu\rangle\langle\nu| U^\dagger(t)\right] |\mu\rangle \\
&= \sum_{\mu\nu} \sqrt{\lambda_\nu} \langle\mu| U(t) |\nu\rangle_E \ \rho_S(0) \ \sqrt{\lambda_\nu} \langle\nu| U^\dagger(t) |\mu\rangle_E \\
&= \sum_{\mu\nu} K_{\mu\nu}(t)\rho_S(0)K_{\mu\nu}^\dagger(t),
\end{aligned} \tag{2.15}$$

where we have defined the system-only operators $\{K_{\mu\nu}\}$, called the **Kraus operators** and are given by

$$K_{\mu\nu}(t) = \sqrt{\lambda_\nu} \langle\mu| U(t) |\nu\rangle . \tag{2.16}$$

Note that the *partial* matrix element leaves an operator acting on the system alone. The equation defining the evolution of the system in terms of Kraus operator is called the **Kraus Operator Sum Representation (OSR)**

$$\rho_S(t) = \sum_{\mu\nu} K_{\mu\nu}(t)\rho_S(0)K_{\mu\nu}^\dagger(t). \tag{2.17}$$

We now look at several important properties that any Kraus OSR must satisfy.

2.2.2 Normalization

The system state should be normalized at all times, so we demand

$$\begin{aligned}
\mathrm{Tr}[\rho_S(t)] &= 1 \\
&= \mathrm{Tr}[\sum K_{\mu\nu}(t)\rho_S(0)K_{\mu\nu}^\dagger(t)] \\
&= \sum \mathrm{Tr}[K_{\mu\nu}(t)\rho_S(0)K_{\mu\nu}^\dagger(t)] \\
&= \sum \mathrm{Tr}[K_{\mu\nu}^\dagger(t)K_{\mu\nu}(t)\rho_S(0)] \\
&= \mathrm{Tr}[\sum K_{\mu\nu}^\dagger(t)K_{\mu\nu}(t)\rho_S(0)].
\end{aligned} \tag{2.18}$$

It is easy to check that the equation is satisfied if $\sum K_{\mu\nu}^\dagger(t)K_{\mu\nu}(t) = I$. However, this condition is not necessary. Thus the system state is guaranteed to be normalized provided the Kraus operators satisfy the following identity,

$$\sum_{\mu\nu} K_{\mu\nu}^\dagger(t)K_{\mu\nu}(t) = I. \tag{2.19}$$

This criterion can be verified for our definition of Kraus operators, given by Eq. (2.16).

$$\begin{aligned}\sum_{\mu\nu} K_{\mu\nu}^\dagger K_{\mu\nu} &= \sum_{\mu\nu} \lambda_\nu \langle\mu| U(t) |\nu\rangle \langle\nu| U^\dagger(t) |\mu\rangle \\ &= \sum_{\nu} \lambda_\nu \langle\nu| U^\dagger(t) \left(\sum_{\mu} |\mu\rangle\langle\mu|\right) U(t) |\nu\rangle \\ &= \sum_{\nu} \lambda_\nu \langle\nu|\nu\rangle \\ &= \sum \lambda_\nu = 1. \end{aligned} \tag{2.20}$$

Thus, such a set of Kraus operators preserves normalization.

Note that when there is just a single Kraus operator, the normalization condition (2.19) forces it to be unitary, which is the case of a closed system.

2.2.3 The Schrödinger equation as a special case

Assume that $U = U_S \otimes U_E$. In this special case the Kraus operators become

$$K_{\mu\nu} = U_S \sqrt{\lambda_\nu} \langle\mu| U_E |\nu\rangle \equiv c_{\mu\nu} U_S. \tag{2.21}$$

It's easy to see that the sum rule normalization condition implies

$$\sum_{\mu\nu} c_{\mu\nu}^* c_{\mu\nu} = 1. \tag{2.22}$$

Thus,

$$\begin{aligned}\rho_S(t) &= \sum_{\mu\nu} c_{\mu\nu} U_S(t) \rho_S(0) c_{\mu\nu}^* U_S^\dagger(t) \\ &= U_S(t) \rho_S(0) U_S^\dagger(t), \end{aligned} \tag{2.23}$$

which is unitary, Schrödinger-like dynamics. Thus the Kraus operator formalism is a generalization of Schrödinger dynamics.

2.2.4 Quantum Maps

The Kraus OSR can be interpreted as a *map* (or synonymously a *process* or *channel*) Φ from the initial to the final system state, i.e.,

$$\rho(t) = \Phi[\rho(0)] \qquad \leftrightarrow \qquad \Phi : \rho(0) \mapsto \rho(t) \ , \tag{2.24}$$

where $\Phi[X] \equiv \sum_\alpha K_\alpha X K_\alpha^\dagger$. Note that Φ is called a superoperator, as it acts on operators. There are some key properties that any such map must possess.

1. Trace Preserving:

$$\begin{aligned}\mathrm{Tr}[\Phi(\rho)] &= \sum_\alpha \mathrm{Tr}(K_\alpha \rho K_\alpha^\dagger) \\ &= \sum_\alpha \mathrm{Tr}(K_\alpha^\dagger K_\alpha \rho) \\ &= \mathrm{Tr}(\sum_\alpha K_\alpha^\dagger K_\alpha \rho) \\ &= \mathrm{Tr}(\rho), \end{aligned} \tag{2.25}$$

where we substitute $\sum_\alpha K_\alpha^\dagger K_\alpha = I$. Thus the map Φ is trace-preserving.

2. Linear: By direct substitution we find:

$$\begin{aligned}\Phi(a\rho_1 + b\rho_2) &= \sum_\alpha \mathrm{Tr}(K_\alpha a\rho_1 K_\alpha^\dagger) + \sum_\alpha \mathrm{Tr}(K_\alpha b\rho_2 K_\alpha^\dagger) \\ &= a\sum_\alpha \mathrm{Tr}(K_\alpha \rho_1 K_\alpha^\dagger) + b\sum_\alpha \mathrm{Tr}(K_\alpha \rho_2 K_\alpha^\dagger) \\ &= a\Phi(\rho_1) + b\Phi(\rho_2) \end{aligned} \tag{2.26}$$

for any scalars a and b. Thus the map Φ is linear.

3a. Positivity:

This property means that Φ maps positive operators to positive operators. Assume the operator $A > 0$, i.e., it has only non-negative eigenvalues, not all zero. Note that any density matrix ρ must be positive, and we can write $A = \sum_i \lambda_i |i\rangle\langle i|$ where all $\lambda_i \geq 0$ using the Spectral Decomposition Theorem.

In order to demonstrate that $\Phi(A) > 0$ it is sufficient show that $\langle\nu|\,\Phi(A)\,|\nu\rangle \geq 0$ for all $|\nu\rangle \in \mathcal{H}_S$, since this means in particular that the eigenvalues of $\Phi(A)$ are all non-negative. Let $|w_a\rangle = K_\alpha^\dagger |\nu\rangle$. Then:

$$\begin{aligned}\langle\nu|\,\Phi(A)\,|\nu\rangle &= \sum_\alpha \langle\nu|\,K_\alpha A K_\alpha^\dagger\,|\nu\rangle \\ &= \sum_\alpha \langle w_a|\,A\,|w_a\rangle \\ &= \sum_{ai} \lambda_i |\langle w_a|i\rangle|^2 \,. \end{aligned} \tag{2.27}$$

Since each term in the sum is positive, $\Phi(A) > 0$, and Φ itself is a positive map.

The Kraus OSR satisfies these three properties, but does every map that satisfy the same properties have a Kraus OSR? It turns out that we must modify and strengthen the positivity property into "complete positivity".

Complete Positivity

The map Φ is a *completely positive* (CP) map if it maps positive operators to positive operators (is "positivity preserving"), and moreover, $\Phi \otimes \mathcal{I}_E^{(k)}$ is positive for all k, where k is the dimension of an ancillary Hilbert space $\mathcal{H}_E$.

3b. Complete Positivity

If $(\Phi \otimes \mathcal{I}_E^{(k)})(A) > 0\ \forall k$, then Φ is called a completely positive (CP) map.

It turns out that conditions $1, 2, 3\text{b}$ are necessary and sufficient for the Kraus OSR. That is:

Theorem 1. *A quantum map Φ has a Kraus operator sum representation [i.e., $\Phi(X) = \sum_\alpha K_\alpha X K_\alpha^\dagger$ with $\sum_\alpha K_\alpha^\dagger K_\alpha = I$] iff it is trace preserving, linear, and completely positive. [NC11]*

Note that there exist positive, but not completely positive maps. Several other properties and equivalences have been found.

2.3 Quantum Maps on a qubit

In this section, by focusing on the case of one qubit, we will develop a geometric picture of the action of quantum maps. The main tool that will allow us to do this is the Bloch sphere representation.
A density matrix of a qubit may be written as (Ch:2 [NC11])

$$\rho = \frac{1}{2}(I + \vec{v} \cdot \vec{\sigma}) \tag{2.28}$$

where $\vec{\sigma} = (\sigma_x, \sigma_y, \sigma_z)$ is a triplet of the Pauli matrices and $\vec{v} = (v_x, v_y, v_z) \in \mathbb{R}^3$ is the Bloch vector. Thus any single qubit density matrix can be thought of as a point in or on the Bloch Sphere. States with $\|\vec{v}\| = 1$ lie on the surface of the sphere and correspond to pure states of the form $\rho = |\psi\rangle\langle\psi|$. Points on the interior of the sphere correspond to mixed states with purity $P = \mathrm{Tr}[\rho^2] < 1$.

2.3.1 Transformation of the Bloch Vector

What happens when a quantum map acts on a single qubit? When the density matrix gets mapped using $\Phi : \rho \mapsto \rho'$, both must be expressible as a Bloch vector. Thus, $\rho' = \frac{1}{2}(I + \vec{v}' \cdot \vec{\sigma})$. We shall show (without proof) that $\rho \mapsto \rho'$ is equivalent to mapping the Bloch vector

$$\vec{v} \mapsto \vec{v}' = M\vec{v} + \vec{c} \tag{2.29}$$

for some real 3×3 matrix M and a vector $\vec{c} \in \mathbb{R}^3$ (Refer [Lid19] for the proof).

Expanding (2.29), we can show that the components of M and c are as follows:

$$\begin{aligned} M_{ij} &= \frac{1}{2}\sum_{\alpha} \mathrm{Tr}\left(\sigma_i K_\alpha \sigma_j K_\alpha^\dagger\right) \\ c_i &= \frac{1}{2}\sum_{\alpha} \mathrm{Tr}\left(\sigma_i K_\alpha K_\alpha^\dagger\right) \end{aligned} \tag{2.30}$$

We can decompose M in a way that will reveal more of the geometric aspects of the transformation. Recall the polar decomposition, which allows us to write any square matrix A as $A = U|A|$, where U is a unitary matrix and $|A| \equiv \sqrt{A^\dagger A}$ is Hermitian (since clearly its eigenvalues are real), a generalization of the polar representation of a complex number $z = e^{i\theta}|z|$. If A is a real matrix, U becomes real-unitary, i.e., orthogonal, and $|A|$ becomes real-Hermitian, i.e., symmetric. So, for our 3×3 real matrix M we can write $M = OS$, for orthogonal O and symmetric $S = \sqrt{M^\dagger M}$. S causes deformation by scaling along the directions of the eigenvectors by a factor of the corresponding eigenvalues. O is a rotation matrix. Now we may interpret the action of a quantum map on a qubit state as mapping the Bloch vector according to

$$\vec{v} \mapsto \vec{v}' = OS\vec{v} + \vec{c}, \tag{2.31}$$

as a shift by $\vec{c}$, a deformation by S and a rotation by O. Because the Bloch sphere represents the set of possible Bloch vectors, we may view the Kraus map acting on a qubit as a transformation of the Bloch sphere that displaces its center by $\vec{c}$ and turns the sphere into an angled ellipsoid.

We now look at a representative example, the phase damping channel.

2.3.2 Phase Damping Channel

The phase damping channel, upon acting on a qubit, outputs the same qubit with probability p and applies the Z operation with probability $1-p$. More succinctly,

$$\rho' = p\rho + (1-p)Z\rho Z, \tag{2.32}$$

where $Z = \sigma_z$. Thus, using (2.24), we can express the Kraus operators as $K_0 = \sqrt{p}I$ and $K_1 = \sqrt{1-p}Z$. Using (2.30), we see that

$$\begin{aligned} c_i &= \frac{1}{2}\sum_\alpha \mathrm{Tr}(\sigma_i K_\alpha K_\alpha^\dagger) \\ &= \frac{1}{2}[p\,\mathrm{Tr}(\sigma_i) + (1-p)\,\mathrm{Tr}(\sigma_i)] \\ &= 0. \end{aligned} \tag{2.33}$$

Similarly,

$$\begin{aligned} M_{ij} &= \frac{1}{2}\sum_\alpha \mathrm{Tr}(\sigma_i K_\alpha \sigma_j K_\alpha^\dagger) \\ &= \frac{1}{2}[p\,\mathrm{Tr}(\sigma_i\sigma_j) + (1-p)\,\mathrm{Tr}(\sigma_i Z\sigma_j Z)] \\ &= p\delta_{ij} + \frac{1}{2}(1-p)J_{ij}\ , \end{aligned} \tag{2.34}$$

where $J_{ij} \equiv \mathrm{Tr}(\sigma_i Z\sigma_j Z)$. Written explicitly the matrix J is:

$$\begin{aligned} J &= \begin{pmatrix} \mathrm{Tr}(XZXZ) & \mathrm{Tr}(XZYZ) & \mathrm{Tr}(XZZZ) \\ \mathrm{Tr}(YZXZ) & \mathrm{Tr}(YZYZ) & \mathrm{Tr}(YZZZ) \\ \mathrm{Tr}(ZZXZ) & \mathrm{Tr}(ZZYZ) & \mathrm{Tr}(ZZZZ) \end{pmatrix} \\ &= \begin{pmatrix} \mathrm{Tr}(-I) & \mathrm{Tr}(\sigma) & \mathrm{Tr}(\sigma) \\ \mathrm{Tr}(\sigma) & \mathrm{Tr}(-I) & \mathrm{Tr}(\sigma) \\ \mathrm{Tr}(\sigma) & \mathrm{Tr}(\sigma) & \mathrm{Tr}(I) \end{pmatrix} \\ &= \mathrm{diag}(-2,-2,2). \end{aligned} \tag{2.35}$$

Thus,

$$M = \begin{pmatrix} 2p-1 & 0 & 0 \\ 0 & 2p-1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{2.36}$$

and

$$\vec{v}' = M\vec{v} = [(2p-1)v_x, (2p-1)v_y, v_z]^t\ . \tag{2.37}$$

The corresponding transformation of the Bloch sphere is shown in Fig. 2.1. There is no shift of the Bloch sphere, while there is a rescaling along the v_x and v_y directions by a factor of $(2p-1)$, and all points on the v_z axis are fixed. The map has two fixed pure states, the north and south poles of the Bloch sphere, $|0\rangle\langle 0|$ and $|1\rangle\langle 1|$. For $p = 1$, the Bloch sphere remains unchanged.

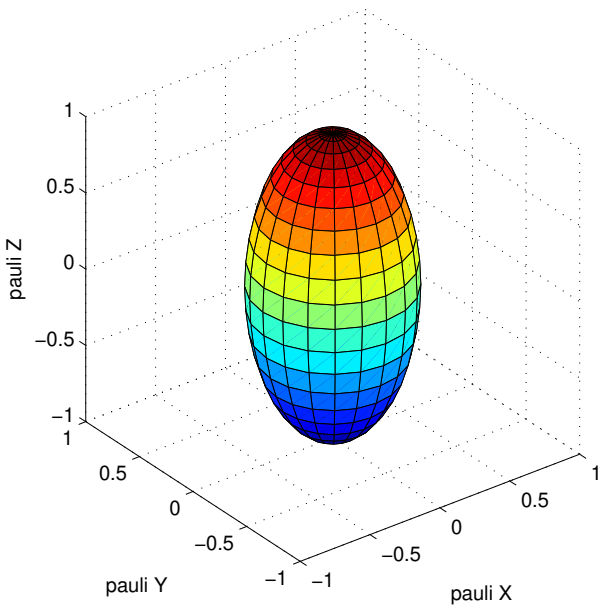

Figure 2.1: The Bloch sphere become an ellipsoid after transformation by the phase damping channel. The invariant states are those on the σ_z axis. The major axis has length 2, the minor axis has length $2(2p-1)$. Figure taken from [Lid19]

The purity of the transformed state is

$$\begin{aligned} P' &= \mathrm{Tr}[(\rho')^2] \\ &= \frac{1}{2}(1 + \|\vec{v}'\|^2) \\ &= \frac{1}{2}[1 + (2p-1)^2(v_x^2 + v_y^2) + v_z^2] \\ &\leq P \ . \end{aligned} \tag{2.38}$$

Thus the purity always decreases under the phase damping channel, except for the states on the v_z axis (with $v_x = v_y = 0$), whose purity is invariant.

The Bit Flip Channel has the same properties as the Phase Damping Channel with all Z replaced with X. Thus, states on the on the v_x axis (with $v_y = v_z = 0$), are invariant.

Chapter 3

Quantum Error Correction

The field of quantum error correction evolved as an answer to the question, *What quantum operations can we reliably perform using unreliable underlying hardware?* We begin with understanding the basics of classical error correction, and the challenges that need to be tackled in quantum error correction. Then, we pick a specific procedure to perform quantum error correction. The next section develops a theoretical framework for all quantum error-correcting codes. Next, we talk about constructing quantum CSS codes from classical linear codes. Finally, we talk about the stabilizer formulation and the associated quantum codes. The development of the chapter follows the work by Neilsen and Chuang [NC11].

3.1 Classical Error Correction

Over the years, classical systems have become incredibly reliable - an error rate of 1 in 10^{17} operations. However, before the advent of reliable hardware, a systems methodology was required to protect against noise.

Let us take the example of a noisy classical communication channel. The channel flips the input bit with probability $p > 0$ and doesn't with probability $1 - p$. This channel is called the *binary symmetric channel*. The basic idea is to *encode* the input by adding redundancy. For example, one way to protect against noise is to replicate every bit thrice i.e.,

$$\begin{aligned} 0 &\longrightarrow 000 \\ 1 &\longrightarrow 111. \end{aligned} \tag{3.1}$$

The strings 000 and 111 are referred to as the **logical 0** and **logical 1** and this type of code is called the repetition code. Suppose the output of the channel is 001. Provided p is low enough, it is highly likely that the input was 0. This is called majority decoding, as we check the majority of the bits in the output.

We notice that majority decoding fails is two or more bits have been flipped. Thus, denoting p_e as the probability of error,

$$\begin{aligned} p_e(\text{repetition code}) &= 3p^2(1-p) + p^3 \\ p_e(\text{without code}) &= p. \end{aligned} \tag{3.2}$$

Thus, the transmission is more reliable when $3p^2(1-p) + p^3 < p$ or $p < 0.5$.

There are several salient differences between classical and quantum information which makes "adding redundancy" not as straightforward. The three major differences are:

- No Cloning: The No Cloning theorem of quantum mechanics prevent arbitrary states $|\psi\rangle$ from being cloned. So, a simple redundancy addition is impossible.
- Continuous Errors: The errors that can affect a quantum state are not discrete, like the classical case. Discerning the type of error would require infinite precision.
- Measurement: In quantum mechanics, measurement of a state collapses the superposition to a basis state, and thus, information is lost.

3.2 Three qubit bit-flip code

These errors, while formidable, are surmountable. To see the effects in action, we take the quantum analogue of the binary symmetric channel. The channel leaves qubits untouched with probability $1-p$ and with probability p, the state $|\psi\rangle$ is taken to $X|\psi\rangle$. Suppose the state $a|0\rangle + b|1\rangle$ is encoded as $a|000\rangle + b|111\rangle$. In other words, the encoding scheme we use is:

$$\begin{aligned} |0\rangle &\longrightarrow |0_L\rangle \equiv |000\rangle \\ |1\rangle &\longrightarrow |1_L\rangle \equiv |111\rangle . \end{aligned} \tag{3.3}$$

The quantum circuit that performs this encoding is:

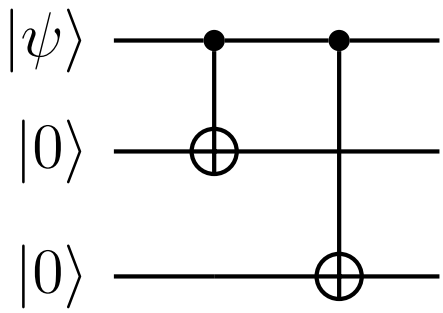

Figure 3.1: Encoding circuit for the three qubit bit flip code

Each of these qubits are now sent through the bit flip channel and we assume that a bit flip occurred on only one or fewer qubits. The error-correction strategy is made of two parts:

- **Error-detection or Syndrome Diagnosis:** We perform a special measurement that which indicates where the error occurred. However, the measurement operators for this measurement are chosen such that they reveal no information about the

initial state. There are four error syndromes, corresponding to four projection operators:

$$\begin{aligned} P_0 &\equiv |000\rangle\langle 000| + |111\rangle\langle 111| \text{ no error} \\ P_1 &\equiv |100\rangle\langle 100| + |011\rangle\langle 011| \text{ bit flip on qubit 1} \\ P_2 &\equiv |010\rangle\langle 010| + |101\rangle\langle 101| \text{ bit flip on qubit 2} \\ P_3 &\equiv |001\rangle\langle 001| + |110\rangle\langle 110| \text{ bit flip on qubit 3.} \end{aligned} \tag{3.4}$$

Notice that these projectors are orthogonal. Suppose a bit flip occurred on qubit one. Thus, the corrupted state is $a|100\rangle + b|011\rangle$. We see that, in this case, $\langle\psi| P_1 |\psi\rangle = 1$ and the outcome is definitely 1. Furthermore, the state after the measurement is undisturbed! As mentioned earlier, the measurement only tells us about which error occurred and nothing about a or b.

- **Recovery:** There are four possible measurement outcomes and each has it's own recovery procedure: 0 - do nothing; 1 - flip 1st qubit; 2 - flip second qubit; 3 - flip 3rd qubit.

Note that the error analysis is similar the classical case and we conclude that the procedure increases the reliability if $p < 0.5$.

Another equivalent way of performing the syndrome can be generalized easily. Instead of the four projectors, assume we measure the two observables Z_1Z_2 ($Z \otimes Z \otimes I$) and Z_2Z_3. Each of these have eigenvalues ± 1 and thus contain one bit of information each. Z_1Z_2 can be thought of as comparing the first and second qubit. Similarly for Z_2Z_3. With both bits, we can conclude which qubit had the error.

3.3 Three qubit phase-flip code

The bit-flip code was very analogous to the classical bit flip channel. However, there are other errors only applicable in the quantum regime - eg. phase errors. The phase-flip channel leaves qubits untouched with probability $1 - p$ and with probability p, the state $|\psi\rangle$ is taken to $Z|\psi\rangle$, i.e., $a|0\rangle + b|1\rangle$ is taken to $a|0\rangle - b|1\rangle$.

While this channel seems to be a totally different channel, with no classical equivalent, the phase-flip and bit-flip are actually similar. Suppose we work in a new basis $|+\rangle \equiv (|0\rangle + |1\rangle)/\sqrt{2}$ and $|-\rangle \equiv (|0\rangle - |1\rangle)/\sqrt{2}$. We notice that in this new basis, $Z|+\rangle = |-\rangle$ and vice versa. In other words, Z flips the labels, acting like a bit-flip error in this basis! Thus, the encoding is done as follows:

$$\begin{aligned} |0\rangle &\longrightarrow |0_L\rangle \equiv |+++\rangle \\ |1\rangle &\longrightarrow |1_L\rangle \equiv |---\rangle . \end{aligned} \tag{3.5}$$

The quantum circuit that performs this encoding is:

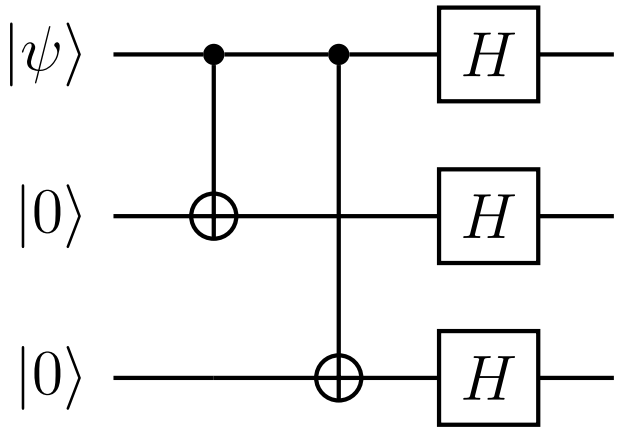

Figure 3.2: Encoding circuit for the three qubit phase flip code

The other steps remain the same, showing that the protocols are equivalent. Instead of measuring Z_1Z_2 and Z_2Z_3, we measure X_1X_2 and X_2X_3. These observables compare the phase of the qubits, and thus, can be used to find out which qubit has an error.

We say that these channels are *unitarily equivalent*, since there is an unitary operator U, such that the action of one channel is the same as the other is preceeded by U and succeeded by $U^\dagger$.

3.4 The Shor Code

So far, we have seen how to error-correct against bit-flip and phase-flip errors. But, as we have seen earlier, the errors on quantum states are continuous. However, there is a simple quantum code that can protect against the effects of arbitrary errors! The code, called the Shor code, is a combination of the bit-flip and the phase-flip codes.

We concatenate the two encoding procedures, i.e., we first encode using the phase-flip encoding and then encode each of them using the bit-flip encoding. In other words, first we encode $|0\rangle \longrightarrow |+++\rangle$ and $|1\rangle \longrightarrow |---\rangle$. Then, $|+\rangle \longrightarrow (|000\rangle + |111\rangle)/\sqrt{2}$ and $|-\rangle \longrightarrow (|000\rangle - |111\rangle)/\sqrt{2}$ (which is the encoding output is we encode $|+\rangle$ and $|-\rangle$ using the bit-flip procedure). The result is a nine-qubit code with logical qubits:

$$\begin{aligned} |0\rangle \longrightarrow |0_L\rangle &\equiv \frac{(|000\rangle + |111\rangle)(|000\rangle + |111\rangle)(|000\rangle + |111\rangle)}{2\sqrt{2}} \\ |1\rangle \longrightarrow |1_L\rangle &\equiv \frac{(|000\rangle - |111\rangle)(|000\rangle - |111\rangle)(|000\rangle - |111\rangle)}{2\sqrt{2}}. \end{aligned} \tag{3.6}$$

The quantum circuit that performs this encoding is:

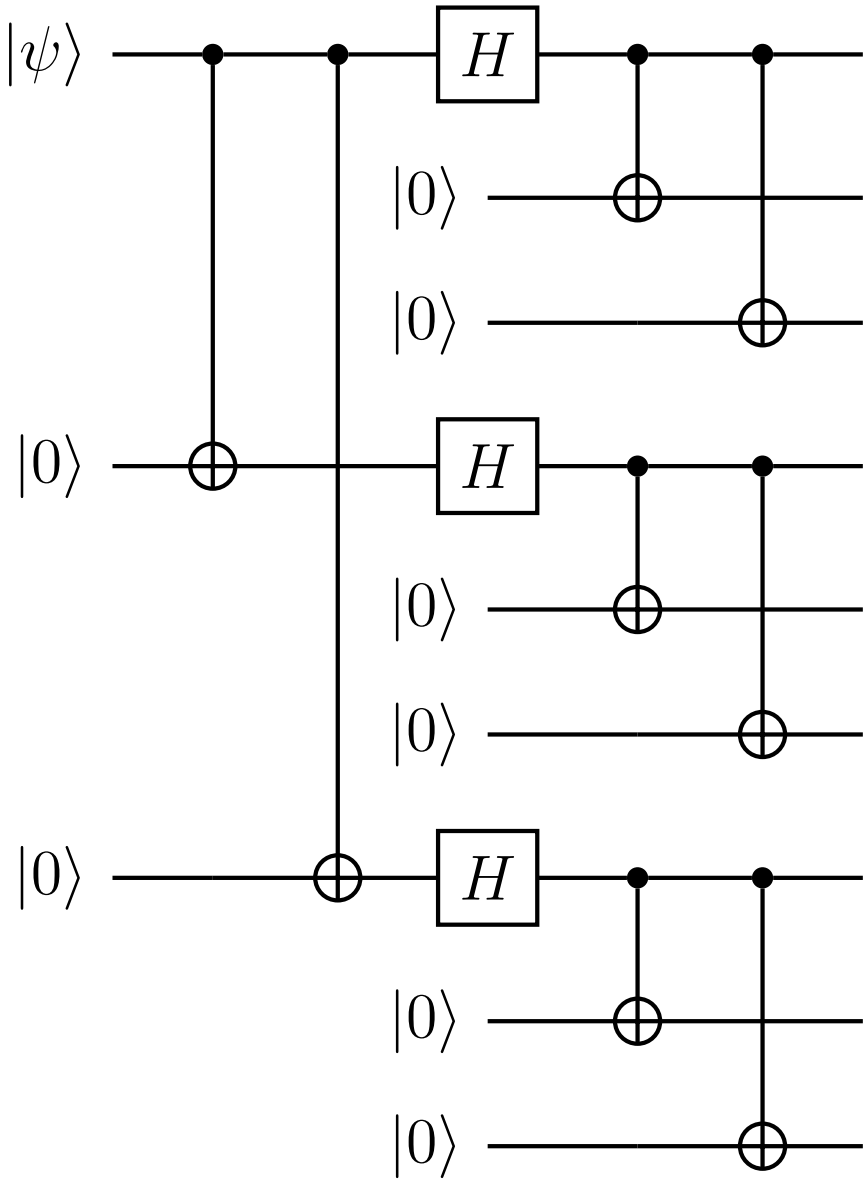

Figure 3.3: Encoding circuit for the Shor nine qubit code

We now show that the Shor code can protect against phase flip and bit flips errors on a single qubit. Assume that a bit flip occurred on qubit 5. Thus, $Z_1Z_2 = 1$, $Z_2Z_3 = 1$, $Z_4Z_5 = -1$, $Z_5Z_6 = -1$, $Z_7Z_8 = 1$, $Z_8Z_9 = 1$, proving that the bit flip occurred on qubit 5. Thus, we can apply X on the fifth qubit and restore the original state. We see that, using the similar procedure as the three qubit bit-flip code, we can fix any single bit-flip error.

Suppose that a single phase flip occurred in qubit 3. This changes the phase of the first set of qubits from $|000\rangle + |111\rangle$ to $|000\rangle - |111\rangle$. Furthermore, if the phase error had occurred in any of the first three qubits, we would have the same effect. Here instead of checking the phase value of individual qubits, we check the phase value of the blocks of qubits. More concretely, we check whether the first and second block of qubits have the

same sign. Then, we check if the second and third block of qubits have the same sign. The syndrome measurement corresponding to this is $X_1X_2X_3X_4X_5X_6$ and $X_4X_5X_6X_7X_8X_9$. In our case, the first measurement gives us -1 and the second measurement gives 1. Thus, we can deduce that the error occurred in one of the first three qubits. This can be accomplished by performing $Z_1Z_2Z_3$. Thus, we can fix any single phase-flip error.

Notice that if both bit and phase flip errors occurred on any qubit, the procedure will first detect the bit flip, correct it, and then detect the phase error and correct it. Thus, Shor code also enables the correction of the combined error ZX.

What is incredible is that, despite only correcting a discrete set of errors, the Shor code can already correct the entire continuum of quantum errors! These errors can be tiny - a rotation about the z axis of the Bloch sphere by $\pi/263$ radians - or it can be a complete error, i.e., replace the qubit with garbage. This discretization is at the core of quantum error correction.

To simplify, we assume that the error occurs only on the first qubit. We describe this noise by a trace-preserving quantum map $\mathcal{E}$. From Theorem 1, we know that we can expand this operation using the Kraus elements $\{E_i\}$. If the state before the error was originally $\alpha\,|0_L\rangle + \beta\,|1_L\rangle$, then the final state will be

$$\mathcal{E}(|\psi\rangle\langle\psi|) = \sum_i E_i|\psi\rangle\langle\psi|E_i^\dagger. \tag{3.7}$$

Let us focus on just one term, say $E_j|\psi\rangle\langle\psi|E_j^\dagger$. Since E_j acts only on qubit 1, it can be expanded as a linear combination of the identity I, bit flip X_1, phase flip Z_1 and the combined bit-phase flip X_1Z_1:

$$E_j = e_{j0}I + e_{j1}X_1 + e_{j2}Z_1 + e_{j3}X_1Z_1. \tag{3.8}$$

Thus, in turn, implies that $E_j\,|\psi\rangle$ can be written as a linear combination of $|\psi\rangle$, $X\,|\psi\rangle$, $Z\,|\psi\rangle$, and $XZ\,|\psi\rangle$. Measuring the error syndrome will collapse the state to one of these, and the appropriate correction can be made. This is a deep fact about quantum error-correction - by just correcting the bit-flip, phase-flip and combined error, any code can correct arbitrary errors!

We now construct a more general theory of quantum error correcting codes.

3.5 Theory of Quantum Error Correction

We delve into creating a working set of conditions that allow the existence of quantum error-correction codes. The framework we develop doesn't guarantee the existence of good codes. The next section will talk about the actual construction of codes.

The salient features and terminology required can be explained using the Shor code. The input states need to be encoded first. The encoding is done using a unitary into a **quantum error-correcting code**. A quantum error-correcting code is defined as a subspace C of a Hilbert space. Here, we introduce the projector onto the code space P. For example, in the three qubit bit-flip code, the logical codewords were $|000\rangle$ and $|111\rangle$.

Thus, $P = |000\rangle\langle 000| + |111\rangle\langle 111|$.

Next, the encoded state is subject to noise. We then perform the syndrome measurement to find which error occurred so that recovery can be attempted. One key requirement is that different syndromes correspond to undeformed and orthogonal subspaces. The subspaces must be orthogonal so that we can distinguish which space the state is in, and in turn, diagnose the error that brought it to that space. The non-deformity condition is required because an error should map orthogonal keywords to orthogonal states. For example, in the bit-flip code, $\alpha|000\rangle + \beta|111\rangle$, upon any single error, gets mapped to a different subspace, all orthogonal to each other and within each subspace, $|000\rangle$ and $|111\rangle$ get mapped to two orthogonal states.

For developing a broader theory, we make two assumptions. Firstly, the noise is described by a quantum operation $\mathcal{E}$ and the error-correction procedure is described by another quantum operation $\mathcal{R}$. Thus, we require that for any state ρ that belongs to the code C,

$$(\mathcal{R} \circ \mathcal{E})\rho = \rho. \tag{3.9}$$

The **quantum error-correction conditions** are a set of condition to whether a quantum error-correcting code protects against a noise $\mathcal{E}$.

Theorem 2 (Quantum error-correction conditions)**.** *Let C be a quantum code, and let P be the projector onto C. Suppose $\mathcal{E}$ is a quantum operation with operation elements $\{E_i\}$. A necessary and sufficient condition for the existence of an error-correction operation $\mathcal{R}$ correcting $\mathcal{E}$ on C is that*

$$PE_i^\dagger E_j P = \alpha_{ij} P, \tag{3.10}$$

for some Hermitian matrix α of complex numbers.

Proof. We first prove sufficiency by constructing an explicit correction operation $\mathcal{R}$. Suppose $\{E_i\}$ is a set of operation elements that satisfy conditions (3.10). Since α is a Hermitian matrix, it can be diagonalized $d = u^\dagger \alpha u$, where u is a unitary and d is diagonal. Define operators $F_k \equiv u_{ik} E_i$. Since $\{F_k\}$ are linear combinations of $\{E_i\}$, $\{F_k\}$ is also a set of operation elements of $\mathcal{E}$. Thus,

$$\begin{aligned} PF_k^\dagger F_l P &= \sum_{ij} u_{ki}^\dagger u_{jl} P E_i^\dagger E_j P \\ &= \sum_{ij} u_{ki}^\dagger \alpha_{ij} u_{jl} P, \end{aligned} \tag{3.11}$$

where the 2nd equality is from (3.10). Since $d = u^\dagger \alpha u$, we see that

$$PF_k^\dagger F_l P = d_{kl} P, \tag{3.12}$$

which is just the diagonal version of the original conditions (3.10). From the polar decomposition theorem, we see that

$$F_k P = U_k \sqrt{P F_k^\dagger F_k P} = \sqrt{d_{kk}} U_k P, \tag{3.13}$$

for some unitary U_k. Thus, the effect of F_k is to rotate the coding subspace into the subspace defined by the projector $P_k \equiv U_k P U_k^\dagger = F_k P U_k^\dagger / \sqrt{d_{kk}}$. Furthermore, the subspaces are orthogonal, which can be shown since when $k \neq l$,

$$P_l P_k = P_l^\dagger P_k = \frac{U_l P F_l^\dagger F_k P u_k^\dagger}{\sqrt{d_{ll} d_{kk}}} = 0. \tag{3.14}$$

Thus, the syndrome measurement is defined by the projectors P_k, with an additional projector if necessary to satisfy the completeness relation. Recovery is accomplished by simply applying $U_k^\dagger$. Thus, the combined detection-recovery step corresponds to the quantum operation $\mathcal{R}(\sigma) = \sum_k U_k^\dagger P_k \sigma P_k U_k$.

Thus, for states ρ in the code, we see that

$$\begin{aligned} U_k^\dagger P_k F_l \sqrt{\rho} &= U_k^\dagger P_k^\dagger F_l P \sqrt{\rho} \\ &= \frac{U_k^\dagger U_k P F_k^\dagger F_l P \sqrt{\rho}}{\sqrt{d_{kk}}} \\ &= \delta_{kl} \sqrt{d_{kk}} P \sqrt{\rho} \\ &= \delta_{kl} \sqrt{d_{kk}} \sqrt{\rho}. \end{aligned} \tag{3.15}$$

Thus, we finally see that,

$$\begin{aligned} \mathcal{R}(\mathcal{E}(\rho)) &= \sum_{kl} U_k^\dagger P_k F_l \rho F_l^\dagger P_k U_k \\ &= \sum_{kl} \delta_{kl} d_{kk} \rho \\ &= \rho. \end{aligned} \tag{3.16}$$

To prove necessity of the conditions, suppose $\{E_i\}$ is the set of errors which are correctable by a operation $\mathcal{R}$ with operation elements $\{R_j\}$. Define a quantum operation $\mathcal{E}_C(\rho) \equiv \mathcal{E}(P\rho P)$. Since $P\rho P$ is in the code for all ρ, it follows that, for all states ρ,

$$\mathcal{R}(\mathcal{E}_C(\rho)) \propto P\rho P. \tag{3.17}$$

Furthermore, the proportionality constant must be a constant c, independent of the state ρ to conserve linearity. Expanding the operations,

$$\sum_{ij} R_j E_i P \rho P E_i^\dagger R_j^\dagger = c P \rho P. \tag{3.18}$$

Since this equation holds for all ρ, the quantum operation with operation elements $\{R_j E_i P\}$ is identical to the quantum operation with a single operation element $\sqrt{c}P$. Thus, we know that there exists complex numbers C_{ki} such that

$$R_k E_i P = c_{ki} P. \tag{3.19}$$

Thus, $P E_i^\dagger R_k^\dagger R_k E_i P = c *_{ki} c_{kj} P$. Since $\mathcal{R}$ is a trace preserving operation $\sum_k R_k^\dagger R_k = I$. Thus, summing over k, we see that

$$P E_i^\dagger E_j P = \alpha_{ij} P, \tag{3.20}$$

where $\alpha_{ij} \equiv \sum_k c *_{ki} c_{kj}$ is a Hermitian matrix of complex numbers. Hence, proved. □

3.5.1 Discretization of errors

In the previous section, we discussed the correction against a specific noise process $\mathcal{E}$. We now that a specific code C and error-correction procedure $\mathcal{R}$ can be used to protect against an entire class of operations.

Theorem 3. *Suppose C is a quantum code and $\mathcal{R}$ is the error-correction operation to recover from the noise-operation $\mathcal{E}$ with elements $\{E_i\}$. Suppose $\mathcal{F}$ is a quantum operation with elements $\{F_j\}$, which are linear combinations of the $\{E_i\}$, i.e., $F_j = \sum_i m_{ji} E_i$ for some matrix m of complex numbers. Then, $\mathcal{R}$ corrects the effects of the noise $\mathcal{F}$.*

Proof. From Theorem 2, we see that $PE_iE_j^\dagger P = \alpha_{ij}P$. We can perform a similar transformation to the set $\{E_i\}$ such that α_{ij} can be made diagonal d_{ij}. The error-correction operation $\mathcal{R}$ has operational elements $U_k^\dagger P_k$, where U_k and P_k are chosen such that for any ρ in the code space:

$$U_k^\dagger P_k E_i \sqrt{\rho} = \delta_{ki}\sqrt{d_{kk}}\sqrt{\rho}. \tag{3.21}$$

Substituting $F_j = \sum_i m_{ji}E_i$, we see that

$$\begin{aligned} U_k^\dagger P_k F_j \sqrt{\rho} &= \sum_i m_{ji}\delta_{ki}\sqrt{d_{kk}}\sqrt{\rho} \\ &= m_{jk}\sqrt{d_{kk}}\sqrt{\rho}\ , \end{aligned} \tag{3.22}$$

and thus,

$$\begin{aligned} \mathcal{R}(\mathcal{F}(\rho)) &= \sum_{kj} U_k^\dagger P_k F_j \rho F_j^\dagger P_k U_k \\ &= \sum_{jk} |m_{jk}|^2 d_{kk}\rho \\ &= \rho. \end{aligned} \tag{3.23}$$

□

Thus, instead of talking about error processes $\mathcal{E}$ correctable by a code C and error-operation $\mathcal{R}$, we can talk about a set of error operators $\{E_i\}$ which are correctable (using equation (3.10)). This is a very powerful result, and as seen before, a code than can correct X, Z and XZ errors on a single qubit, can correct arbitrary errors on a single qubit.

3.6 Constructing quantum codes

With the framework laid down, we now begin constructing quantum codes. We first describe classical correction codes and using them define quantum CSS codes.

3.6.1 Classical Linear Codes

A linear code C encoding k bits of information into n bits is specified by a n by k generator matrix G whose entries are all elements of $\mathbb{Z}_2$, i.e., made of zeros and ones. Thus, an input message x is encoded as Gx, where x is treated as a column vector. Note that

all operations are performed modulo 2.

For example, the repetition code maps single bits to three copies of itself and, therefore must have a 3 by 1 generator matrix,

$$G = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \tag{3.24}$$

since $G[0] = (0, 0, 0)$ and $G[1] = (1, 1, 1)$. A code that encodes k bits into n bits is called a $[n, k]$ code. We notice that each column of the G matrix is n bits long and the encode message is the linear combination of the columns. More concretely, if the $G = [g^1, g^2, \ldots, g^k]$ (where g^i is an n by 1 vector) and the message is $x = [x_1, x_2, \ldots, x_k]$, then the encoded message is given by $Gx = x_1 g^1 + x_2 g^2 + \ldots x_k g^k$. Thus, for all messages to be uniquely encoded, the only constraint is that the columns of G are **linearly independent**.

Property 1. *Adding one column of G to another results in a generator matrix generates the same code. The columns are linearly independent and the code is all possible linear combinations of the columns. Thus, adding a column to another doesn't change the code.*

The greatest advantage of the linear codes is the compact specification. A general $[n, k]$ code needs n bits for each of the 2^k codewords, needing a total of $n2^k$ bits. As we have seen, a linear code needs nk bits.

Another alternate, but equivalent, formulation of linear codes is using the parity matrix. This formulation makes the error-correction aspect more transparent. An $[n, k]$ code is the set of all n-bit vectors over $\mathbb{Z}_2$ such that

$$Hx = 0, \tag{3.25}$$

where H is a $n - k$ by n matrix called the parity check matrix. We similarly note that the rows of the parity check matrix must be linearly independent.

Property 2. *Adding one row of H to another results in a parity check matrix for the same code.*

We now connect the two formulations. To go from the parity check matrix to the generator matrix, pick k linearly independent vectors $y_1, \ldots, l_k$ that span the kernel of H ($Hy_i = 0$). Set these as the columns of G. For the reverse direction, pick $n - k$ linearly independent vectors $x_1, \ldots, x_{n-k}$ orthogonal to the columns of G and set the rows of H to be $y_1^T, \ldots, y_{n-k}^T$.

Property 3. *For any linear code, $HG = 0$. Since, for all x, Gx belongs to the code. Thus, $HGx = 0$, by the definition of the parity check matrix. Since this equation is true for all x, HG must be identically zero.*

The parity check makes error-correction easy. For an input message x, we encode it using the generator matrix G such that $y = Gx$. An error e occurs that makes the encoded message $y' = y + e$. Thus, $Hy' = H(y + e) = He$ is called the error syndrome. In the cases where no errors or just one error occurred, we can compute He_j and compare it with He. This figures out the error bit j which can be added to y' to restore y.

We now introduce the concept of distance. For two n bit words x and y, the **Hamming distance** $d(x, y)$ is defined as the number of places that x and y differ. The **Hamming weight** of a word x is it's distance from the zero string $\mathrm{wt}(x) \equiv d(x, 0)$. Note that $\mathrm{wt}(x + y) \equiv d(x, y)$. Using our previous encoding example, if the probability of a error is less than 0.5, then the most likely y for a received y' is the one that minimizes $d(y, y') = wt(e)$.

We now define a global property of a code. The distance of a code is the minimum distance between any two codewords,

$$d(C) \equiv \min_{x,y \in C, x \neq y} d(x, y). \tag{3.26}$$

Since $d(x, y) = \mathrm{wt}(x, y)$, and for a linear code, $x + y$ is also a code word,

$$d(C) = \min_{x \in C, x \neq 0} \mathrm{wt}(x). \tag{3.27}$$

Thus, a code is specified by three parameters $[n, k, d]$. A code with a distance of $2t+1$ can correct up to t errors.

We also look at an important construction called the **dual construction**. Consider a code $[n, k]$ code C with a generator matrix G and parity check matrix H. We define a new code, the dual code of C, $C^\perp$ with generator matrix H^T and parity check matrix G^T. A code is weakly self-dual if $C \subseteq C^\perp$ and strictly self-dual if $C = C^\perp$.

Property 4. *A code with generator matrix G is weakly self-dual iff $G^T G = 0$.*

Proof. Consider a weakly self-dual code C with generator matrix G. Thus, $C \subseteq C^\perp$. For all x, $Gx \in C$ and thus $Gx \in C^\perp$. Thus, from the definition of the parity check matrix, $G^T G x = 0$. Since this is true for all x, $G^T G = 0$. For the reverse direction, consider a code and its dual such that $G^T G = 0$. Thus, for all x, $G^T(Gx) = 0$ and in turn implies that $Gx \in C^\perp$. Thus, $C \subseteq C^\perp$. □

Property 5. *For a linear code C, if $x \in C^\perp$, then $\sum_{y \in C} (-1)^{x.y} = |C|$ and if $x \notin C^\perp$, then $\sum_{y \in C} (-1)^{x.y} = 0$.*

Proof. Consider an $x \in C^\perp$. Thus, $x = H^T z$ for some z. Thus,

$$\begin{aligned} \sum_{y \in C} (-1)^{x^T y} &= \sum_{w \in \{0,1\}^k} (-1)^{z^T H G w} \\ &= \sum_{w \in \{0,1\}^k} (-1)^0 \\ &= |C|. \end{aligned} \tag{3.28}$$

Similarly, we pick $x \notin C^\perp$. Thus, $G^T x \neq 0$ or $x^T G \neq 0$.

$$\begin{aligned} \sum_{y \in C} (-1)^{x^T y} &= \sum_{w \in \{0,1\}^k} (-1)^{x^T G w} \\ &= \sum_{w \in \{0,1\}^k} (-1)^{v^T w}, \end{aligned} \tag{3.29}$$

where $v = x^T G \neq 0$. Let v have $n > 0$ entries 1. Thus, $v^T w$ is the sum of entries of w whose corresponding v entries are not zero. Define w' such that $w'_i = w_i$ when v_i is not zero, and 0 elsewhere. Thus $v^T w = \text{rank}\,(w')$. Thus,

$$\begin{aligned}\sum_{w\in\{0,1\}^k} (-1)^{v^T w} &= \sum_{w\in\{0,1\}^k} (-1)^{\text{rank}(w')} \\ &= \sum_{w\in\{0,1\}^k,\text{rank}(w')\text{ even}} 1 - \sum_{w\in\{0,1\}^k,\text{rank}(w')\text{ odd}} 1 \\ &= 0. \end{aligned} \tag{3.30}$$

This is because, independent of n, half of the w' have even rank and other half are odd. □

3.6.2 Calderbank-Shor-Steane Codes

We now look at a large class of quantum codes called CSS codes. Suppose C_1 and C_2 are $[n, k_1]$ and $[n, k_2]$ classical linear codes such that $C_2 \subset C_1$ and C_1 and $C_2^\perp$ both correct t errors. We now define a $[n, k_1 - k_2]$ quantum code, called CSS(C_1, C_2) which can correct t errors. This is called the CSS code of C_1 over C_2.

Before we begin, we define a relation using C_1 and C_2 called R. We say that $(u, v) \in R$ for $u, v \in C_1$ iff there exists a $w \in C_2$ such that $u = v + w$.

Property 6. *Relation R, defined above, is a equivalence relation and splits C_1 into equivalence classes or cosets.*

Proof. We show that R is reflexive, symmetric and transitive.

- Reflexive. For every $u \in C_1$, $(u, u) \in R$ iff there exists a $w \in C_2$ such that $u = u+w$ or $w = 0$. We know that $w = 0 \in C_2$ because $H_2 0 = 0$, where H_2 is the parity check matrix for code C_2.
- Symmetric. For every $u, v \in C_1$, $(u, v) \in R$ iff there exists a $w \in C_2$ such that $u = v + w$. Adding w, we see that $u + w = v$. Thus, $(v, u) \in R$.
- Transitive. For every $u, v, w \in C_1$, $(u, v), (v, w) \in R$ iff there exists a $\alpha, \beta \in C_2$ such that $u = v + \alpha$ and $v = w + \beta$. Adding the equations, we see that $u = w + \alpha + \beta$. Since $\alpha, \beta \in C_2$ and C_2 is a linear code, $\gamma = \alpha + \beta \in C_2$. Thus, $u = w + \gamma$ and thus, $(u, w) \in R$.

□

Suppose $x \in C_1$ is any codeword in C_1. We define the quantum state $|x + C_2\rangle$ by

$$|x + C_2\rangle \equiv \frac{1}{\sqrt{|C_2|}} \sum_{y\in C_2} |x + y\rangle\,, \tag{3.31}$$

where the addition is bitwise addition modulo 2. Suppose x' is an element of C_1 such that $x - x' \in C_2$. We see that $(x, x') \in R$. Furthermore, we notice that $|x + C_2\rangle = |x' + C_2\rangle$. Thus, every coset requires just one representative as every other member of the coset results in the same final state. Additionally, for x, x' belonging to different cosets, there

are no y, y' such that $x + y = x' + y'$. Thus, $|x + C_2\rangle$ is orthogonal to $|x' + C_2\rangle$. The number of cosets are $|C_1|/|C_2| = 2^{k_1 - k_2}$.

To sum up, the subcode C_2 and defines $2^{k_1-k_2}$ cosets in C_1, where each coset has a representative element, and elements from different cosets are orthogonal to each other.

The quantum code CSS(C_1, C_2) is defined as the vector space spanned by the states $|x + C_2\rangle$ for all $x \in C_1$. Let us now see how this code corrects errors.

Let the bit flip errors be described by an n bit vector e_1 where 1s where bit flip occurred. Similarly, let the phase flip errors be described by an n bit vector e_2 where 1s where phase flip occurred. Thus, the initial state $|x + C_2\rangle$ now becomes,

$$\frac{1}{\sqrt{|C_2|}} \sum_{y \in C_2} (-1)^{(x+y).e_2} |x + y + e_1\rangle . \tag{3.32}$$

We now perform an operation, with sufficient ancillary qubits as required initially in the $|0\rangle$, to store the syndrome of the code C_1. In other words, we perform the following operation

$$|x + y + e_1\rangle |0\rangle \to |x + y + e_1\rangle |H_1(x + y + e_1)\rangle = |x + y + e_1\rangle |H_1 e_1\rangle , \tag{3.33}$$

since $(x + y) \in C_1$. This operation can be performed using only CNOT gates as is shown in the property below. Thus, the effect of this procedure is to create the following state,

$$\frac{1}{\sqrt{|C_2|}} \sum_{y \in C_2} (-1)^{(x+y).e_2} |x + y + e_1\rangle |H_1 e_1\rangle . \tag{3.34}$$

Now, the ancillary qubits are measured and discarded. Using $H_1 e_1$, we can infer e_1 since C_1 can correct upto t errors. Thus, applying NOT gates are the correct positions using e_1, we get the state

$$\frac{1}{\sqrt{|C_2|}} \sum_{y \in C_2} (-1)^{(x+y).e_2} |x + y\rangle . \tag{3.35}$$

Now to detect phase flip errors, we use the similar trick as before. Applying Hadamard gates to all the qubits, we get

$$\frac{1}{\sqrt{|C_2|2^n}} \sum_{z'} \sum_{y \in C_2} (-1)^{(x+y).(e_2+z')} |z'\rangle , \tag{3.36}$$

where the sum is over all possible n bit z'. Setting $z \equiv z' + e_2$,

$$\frac{1}{\sqrt{|C_2|2^n}} \sum_{z} \sum_{y \in C_2} (-1)^{(x+y).z} |z + e_2\rangle . \tag{3.37}$$

Now, using property 5, we see that if $z \in C_2^\perp$ then $\sum_{y \in C_2} (-1)^{y.z} = |C_2|$ and if $z \notin C_2^\perp$ then $\sum_{y \in C_2} (-1)^{y.z} = 0$. Thus the state is,

$$\frac{1}{\sqrt{2^n/|C_2|}} \sum_{z \in C_2^\perp} (-1)^{x.z} |z + e_2\rangle . \tag{3.38}$$

This state looks like a bit error described by e_2. So, we introduce the necessary ancilla qubits, apply the parity check matrix H_2 for $C_2^{\perp}$, measure to get $H_2 e_2$ and correct the e_2 errors. Thus, the state is,

$$\frac{1}{\sqrt{2^n/|C_2|}} \sum_{z \in C_2^{\perp}} (-1)^{x.z} |z\rangle . \tag{3.39}$$

Now applying Hadamard to each qubit again, noticing that the Hadamard is the inverse of itself, and we are applying the inverse operation for the special case $e_2 = 0$, we see that the state is

$$\frac{1}{\sqrt{|C_2|}} \sum_{y \in C_2} |x + y\rangle , \tag{3.40}$$

which is the original state. Thus, upto t bit-flip and phase-flip errors have been corrected.

Property 7. *Given a parity check matrix H, the transformation $|x\rangle |0\rangle \rightarrow |x\rangle |Hx\rangle$ can be computed using only CNOT gates.*

Proof. $|x\rangle$ is an n dimensional vector and $|0\rangle$ is an $n-k$ dimensional vector. This is because Hx will result in an $n-k$ bit vector. Also note that the CNOT gate maps $|x\rangle |y\rangle$ to $|x\rangle |x \oplus y\rangle$. Consider an example computation of Hx,

$$\begin{bmatrix} a & b & c \\ d & e & f \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} ax \oplus by \oplus cz \\ dx \oplus ey \oplus fz \end{bmatrix} . \tag{3.41}$$

If $a = 0$, x doesn't contribute to the sum, and if $a = 1$, x is added (modulo 2) to the sum. Thus, we conclude that the following rule can be used. If the (i, j) entry of the parity check matrix H H_{ij}, is 1, add a CNOT gate between the i^{th} qubit and j^{th} ancillary qubit. This set of CNOT operations will result in the ancillary qubits in the state $|Hx\rangle$. As an example, consider,

$$|x\rangle = \begin{bmatrix} x \\ y \\ z \end{bmatrix} , \tag{3.42}$$

and

$$H = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} . \tag{3.43}$$

Then, the circuit made of CNOTs that performs this operation is,

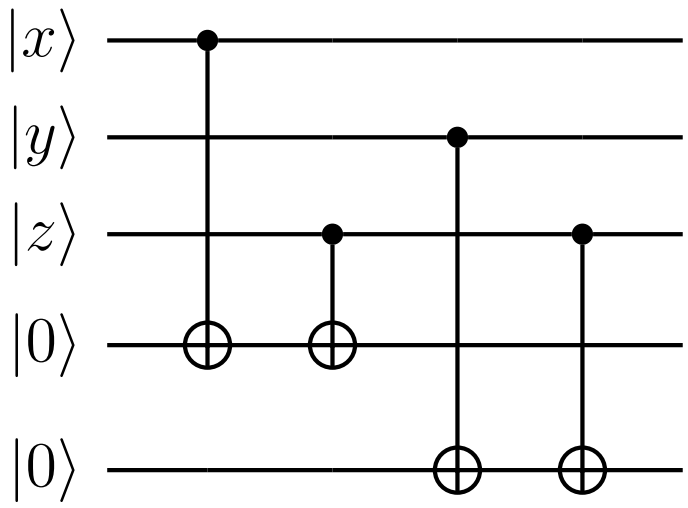

Figure 3.4: Circuit to perform Hx for the given H

□

3.6.3 Steane Code

The Steane Code is an example of a CSS code that is constructed using the $[7, 4, 3]$ Hamming code, which we call C, whose parity check matrix is

$$H[C] = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}. \tag{3.44}$$

Using the procedure given in the previous section, we can deduce that the generator matrix for C is

$$G[C] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{bmatrix}. \tag{3.45}$$

We now define $C_1 \equiv C$ and $C_2 \equiv C^{\perp}$. Thus, $G[C_1] = G[C]$, $H[C_1] = H[C]$, $G[C_2] = H[C_1]^T$ and $H[C_2] = G[C_1]^T$. We see that the span of rows of $H[C_2]$ strictly contain that of $H[C_1]$, and since the corresponding codes are the kernels of the parity check matrices, $C_2 \subset C_1$. Furthermore, $C_2^{\perp} = C$, and thus both C_1 and C_2 are distance 3 codes which can correct errors on a single qubit. Thus, CSS(C_1, C_2) is $[7, 1]$ quantum code.

We know explicitly calculate the codewords for both codes and show the logical codewords. The codewords for the codes are calculated as Gx for all x.

x	$G[C_1]x$
0000	0000000
0001	0001111
0010	0010110
0011	0011001
0100	0100101
0101	0101010
0110	0110011
0111	0111100
1000	1000011
1001	1001100
1010	1010101
1011	1011010
1100	1100110
1101	1101001
1110	1110000
1111	1111111

Table 3.1: Codewords for C_1

x	$G[C_2]x$
000	0000000
001	1010101
010	0110011
011	1100110
100	0001111
101	1011010
110	0111100
111	1101001

Table 3.2: Codewords for C_2

To confirm, we explicitly see that $C_2 \subset C_1$. Thus, the number of cosets is $|C_1|/|C_2| = 2$. So we need to pick two elements x, x' from C_1 such there exists no $y \in C_2$ such that $x - x' = y$. We pick $x = 0000000$ and $x' = 1111111$. These two elements become the logical 0_L and logical 1_L for the encoded qubit. Thus,

$$|0_L + C_2\rangle = \frac{1}{\sqrt{8}}[|0000000\rangle + |1010101\rangle + |0110011\rangle + |1100110\rangle + |0001111\rangle + |1011010\rangle + |0111100\rangle + |1101001\rangle] \quad (3.46)$$

$$|1_L + C_2\rangle = \frac{1}{\sqrt{8}}[|0000000\rangle + |1010101\rangle + |0110011\rangle + |1100110\rangle + |0001111\rangle + |1011010\rangle + |0111100\rangle + |1101001\rangle] \quad (3.47)$$

Note that, as proven, $|0_L + C_2\rangle$ is orthogonal to $|1_L + C_2\rangle$.

3.7 Stabilizer Codes

Stabilizer codes, also called additive quantum codes, are another important class of quantum codes. We first delve into the stabilizer formalism of quantum mechanics, and then come discuss the error-correction aspect.

3.7.1 Stabilizer Formalism

Consider the Bell state

$$|\psi\rangle = \frac{|00\rangle + |11\rangle}{\sqrt{2}} \tag{3.48}$$

We see that $X_1X_2 |\psi\rangle = |\psi\rangle$ and $Z_1Z_2 |\psi\rangle = |\psi\rangle$. We say this state is **stabilized** by the operators X_1X_2 and Z_1Z_2. The basic idea of the stabilizer formalism is that we can describe the system in question by the operators that stabilize them. The key to this reformulation is group theory. The group in question is the Pauli group G_n on n qubits, with multiplicative factors. For the single qubit case,

$$G_1 \equiv \{\pm I, \pm iI, \pm X, \pm iX, \pm Y, \pm iY, \pm Z, \pm iZ\} \tag{3.49}$$

This set of matrices forms a group under the matrix multiplication operation. G_n over n qubits is made of all n-fold tensor products of these matrices.

Suppose S is a subgroup of G_n and define V_S to the set of n qubit states which are fixed (eigenvalue is one) by every element of S. V_S is the vector space stabilized by S, and S is the stabilizer of the space V_S. Thus, V_S is the intersection of the subspaces fixed by each operator in S (eigenvalue one eigenspaces of elements of S).

We now look at an example of the stabilizer formulation for $n = 3$ and $S = \{I, Z_1Z_2, Z_2Z_3, Z_1Z_3\}$. The subspace fixed by Z_1Z_2 is spanned by $|000\rangle, |001\rangle, |110\rangle$ and $|111\rangle$. The subspace fixed by Z_2Z_3 is spanned by $|000\rangle, |100\rangle, |011\rangle$ and $|111\rangle$. Thus, the overlapping states are $|000\rangle$ and $|111\rangle$. Furthermore, these two states are fixed by Z_1Z_3 and I as well. Thus, V_S is spanned by these these two states.

We determined V_S by looking at the subspace of two operators only. This is another property from group theory - the description of a group by its **generators**. A set of elements $g_1, \ldots, g_l$ in a group G is said to generate the group G if every element of G can be written as a product list of the generators. We then represent this as $G = \langle g_1, \ldots, g_l \rangle$. For our example, $S = \langle Z_1Z_2, Z_2Z_3 \rangle$ as $Z_1Z_3 = (Z_1Z_2)(Z_2Z_3)$ and $I = (Z_1Z_2)^2$. To check if a state is stabilized by a set S, we just need to check if it's stabilized by its generators. The advantage arises in its compactness. If the size of a group is N, the size of its generator set is at most $\log(N)$.

Some subgroups of S can only stabilize a trivial subspace. For example, any subgroup with $-I$, as $-I |\psi\rangle = |\psi\rangle$ is only possible for $|\psi\rangle = 0$. What are the conditions for a non-trivial subspace?

- the elements of S commute.
- $-I$ is not an element of S.

The seven qubit Steane code can be written in the stabilizer formulation. Six generators generate the stabilizer for the code space of the Steane code.

Name	Operator
g_1	$IIIXXXX$
g_2	$IXXIIXX$
g_3	$XIXIXIX$
g_4	$IIIZZZZ$
g_5	$IZZIIZZ$
g_6	$ZIZIZIZ$

Table 3.3: Stabilizer generators for the seven qubit Steane code.

Note the similarity in structure between the generators and the parity check matrices for C_1 and C_2 described in the previous section. The first three generators have Xs in the locations corresponding to 1s in the rows of the parity check matrix $H[C_1]$. Similarly, the last three generators have Zs in the locations corresponding to 1s in the rows of the parity check matrix $H[C_2]$. This proves that, there is nothing special about Steane code's status as a quantum code - it is merely a subspace of a Hilbert space which happens to have a description using stabilizers.

In practice, we want our generator set to be independent. In other words, removing any generator g_i from the group makes the generated group smaller,

$$\langle g_1, \ldots, g_{i-1}, g_{i+1}, \ldots, g_l \rangle \neq \langle g_1, \ldots, g_l \rangle. \tag{3.50}$$

Checking if a set of generators in independent is done using a check matrix, so-named because it plays a role analogous to the parity check matrix.

Suppose $S = \langle g_1, \ldots, g_l \rangle$ for a n qubit space. The check matrix is a $l \times 2n$ matrix whose rows correspond to the generators; the left half of the matrix contains 1s where the generator has an X, and the right side of the matrix contains 1s where the generator has a Z. Lastly, if the generator has a Y, both left and right side entries are 1. More concretely, ith row is constructed as follows. If g_i has an I on the jth qubit, then the jth and $n+j$th column entries are 0; if it contains an X on the jth qubit, then the jth column entry is 1 and $n+j$th column entry is 0; if it contains an Z on the jth qubit, then the jth column entry is 0 and $n+j$th column entry is 1; if it contains an Y on the jth qubit, then the jth and $n+j$th column entries are 1.

As an example, we now show the check matrix for the seven qubit Steane code.

$$\left[\begin{array}{ccccccc|ccccccc} 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{array}\right]. \tag{3.51}$$

Note that the check matrix doesn't contain information about the multiplicative factors. However, we can glean other information. Define $r(g)$ to denote the $2n$ dimensional

row vector representation for a generator g. Furthermore, define a $2n \times 2n$ matrix Λ

$$\Lambda = \begin{bmatrix} 0 & I \\ I & 0 \end{bmatrix}, \tag{3.52}$$

where the off-diagonal I are both $n \times n$.

Property 8. *Two generators g and g' commute iff $r(g)\Lambda r(g')^T = 0$, where the arithmetic is done modulo 2.*

Proof. Let us split $r(g)$ as $[a, b]$ and $r(g')$ as $[a', b']$, where a, b, a', b' are row vectors with n entries. Thus, $r(g)\Lambda r(g')^T$ can be expressed as $a.b'^T + b.a'^T$. In other words, $r(g)\Lambda r(g')^T$ represents a twisted inner product; the dot product of the first half of the $r(g)$ and the second half of $r(g')$ plus the dot product of the second half of the $r(g)$ and the first half of $r(g')$ (modulo 2).
We now show that $r(g)\Lambda r(g')^T$ is the number of anti-commuting elements (modulo 2). Consider the following cases.

- g has I in the jth position. Thus, $r(g)$ has 0 in the jth and the $(n+j)$th position. Thus, g' can have any entry in the jth position as $r(g)\Lambda r(g')^T$ will have a contribution of 0 from the jth entry. This makes sense, as I commutes with any operator. Thus, g having I in any position adds nothing to the number of anti-commuting elements. Similarly, we can draw the same conclusion if g' has an I in any position.

- g has X in the jth position. Thus, a has 1 in the jth and b has 0 in the jth position.

  - $a.b'^T + b.a'^T$ has a contribution of 0 from the jth entry iff b' has 0 in the jth entry and a' has either 0 or 1 in the jth entry. Thus, a contribution of 0 occurs iff g' has operator I or X in the jth position. This makes sense X commutes with either X or I.
  - $a.b'^T + b.a'^T$ has a contribution of 1 from the jth entry iff b' has 1 in the jth entry and a' has either 0 or 1 in the jth entry. Thus, a contribution of 1 occurs iff g' has operator Z or Y in the jth position. This makes sense X anti-commutes with either Z or Y.

  We can draw similar conclusions for g having Z or Y.

In other words, $r(g)\Lambda r(g')^T$ gets a contribution of 0 from the jth entry if the jth operator of g and g' commute and a contribution of 1 if if the jth operator of g and g' anti-commute. Since, the arithmetic is done modulo 2, $r(g)\Lambda r(g')^T$ is the number of anti-commuting elements (modulo 2). Thus,

$$\begin{aligned} [g, g'] = 0 &\iff \text{Number of anti-commuting elements is even} \\ &\iff \text{Number of anti-commuting elements is 0(modulo 2)} \\ &\iff r(g)\Lambda r(g') = 0. \end{aligned} \tag{3.53}$$

□

Property 9. *Let S be a subgroup of G_n such that $-I$ is not an element of S. Then $g^2 = I$ for all $g \in S$, and thus $g = g^\dagger$.*

Proof. Every member g of G_n can be written in the form

$$g = (i)^k a_1 \otimes a_2 \otimes \ldots \otimes a_n, \tag{3.54}$$

where each a_i can be either X_i, Y_i, Z_i and k can be any integer from 0 to 3. Thus,

$$\begin{aligned} g^2 &= (i)^{2k} a_1^2 \otimes a_2^2 \otimes \ldots \otimes a_n^2 \\ &= \pm I, \end{aligned} \tag{3.55}$$

as $(i)^{2k}$ is either 1 or -1 and $a_i^2 = I$ for X_i, Y_i, Z_i. Given that $-I \notin S$, $g^2 = I$ for all $g \in S$. Furthermore, multiplying by $g^\dagger$ and since g is unitary, $g^\dagger g = I$, we see that $g = g^\dagger$. □

We now show the independence of generators can be inferred from the check-matrix.

Property 10. *Let $S = \langle g_1, \ldots, g_l \rangle$ be such that $-I \notin S$. The generators g_1 through g_l are independent if and only if the rows of the corresponding check-matrix are linearly independent.*

Proof. We prove the contrapositive. First, notice that $g_i^2 = I$ for all i. Furthermore, observe that $r(g)+r(g') = r(gg')$, so addition in the row vector representation corresponds to multiplication of group elements. Thus, the rows of the check matrix are linearly dependent $\sum_i a_i r(g_i) = 0$ and $a_j \neq 0$ for some j, if and only if $\prod_i g_i^{a_i}$ is equal to the identity, upto an overall multiplicative factor. Since $-i \notin S$, the multiplicative factor must be 1. Thus, this corresponds to $g_j = g_j^{-1} = \prod_{i \neq j} g_i^{a_i}$ and thus, are dependent generators. □

Property 11. *Let $S = \langle g_1, \ldots, g_l \rangle$ be generated by l independent elements from G_n such that $-I \notin S$. Fix i in the range $1, \ldots, l$. Then, there exists $g \in G_n$ such that $gg_ig^\dagger = -g_i$ and $gg_jg^\dagger = g_j$ for all $j \neq i$.*

Proof. Let G be the check matrix associated with the generators. The rows are linearly independent using Property 10. Thus, there exists a $2n$-dimensional vector x such that $G\Lambda x = e_i$, where e_i is an l-dimensional vector with 1 in the ith position and 0 elsewhere. Let g be such that $r(g) = x^T$. Thus, by the definition of x, we have $r(g_j)\Lambda r(g)^T = 0$ for $j \neq i$ and $r(g_i)\Lambda r(g) = 1$. Thus, we know that, using Property 8,

- For all $j \neq i$, $[g_j, g] = 0$. Thus, $g_j g = g g_j$ and thus, $g_j = g g_j g^\dagger$.
- $\{g_i, g\} = 0$. Thus, $g_i g = -g g_i$ and thus, $g_i = -g g_i g^\dagger$.

□

Property 12. *Let $S = \langle g_1, \ldots, g_{n-k} \rangle$ be generated by $n-k$ independent elements from G_n such that $-I \notin S$. Then, V_S is a 2^k dimensional space.*

Proof. Let $x = (x_1, \ldots, x_{n-k})$ be a vector of $n-k$ elements of $\mathbb{Z}_2$. Define,

$$P_S^x \equiv \frac{\prod_{j=1}^{n-k} (I + (-1)^{x_j} g_j)}{2^{n-k}}. \tag{3.56}$$

Because $(I + g_j)/2$ is the projector onto the $+1$ eigenspace of g_j, we see that, by the definition of V_S, $P_S^{(0,\ldots,0)}$ must be the projector onto V_S. By Property 11, for each x, there

exists a g_x in G_n such that $g_x P_S^{(0,\ldots,0)} g_x^\dagger = P_S^x$, and thus, the dimension of P_S^x is the same as V_S. Furthermore, for distinct x, P_S^x are orthogonal. We also see that

$$I = \sum_x P_S^x. \tag{3.57}$$

The left hand side is a projector onto a 2^n-dimensional space, while the right hand side is a sum over 2^{n-k} orthogonal projectors of the same size as V_S, which means V_S must be of size 2^k. □

With this, we conclude our look at the basic elements of the stabilizer formalism. We have used the stabilizer formalism to describe vector spaces, so far.

3.7.2 Unitary gates

We now expand the formalism to talk about the dynamics of a system as well. Suppose we apply a unitary operation U to a vector space V_S stabilized by the group S. Let $|\psi\rangle$ be any element of V_S. Then, for any element g of S,

$$U|\psi\rangle = Ug|\psi\rangle = UgU^\dagger U|\psi\rangle, \tag{3.58}$$

and thus, $U|\psi\rangle$ is stabilized by $UgU^\dagger$. In other words, the vector space UV_S is stabilized by the group $USU^\dagger \equiv \{UgU^\dagger | g \in S\}$. Furthermore, if $g_1, \ldots, g_l$ generate S, then $Ug_1U^\dagger, \ldots, Ug_lU^\dagger$ generate $USU^\dagger$. Thus, to compute the change in stabilizer, we only need to compute how it affects the generators!

As a concrete example, we look at the the action of the Hadamard gate. We see that,

$$HZH^\dagger = X. \tag{3.59}$$

Thus, the state stabilized by Z, which we know to be $|0\rangle$, upon acting of the Hadamard gate becomes stabilized by X, which we know to be $|+\rangle$. We now look at the CNOT gate, which along with the Hadamard gate, can create entanglement.

Denoting U to be CNOT gate (with qubit 1 as control and 2 as target), we see that

$$\begin{aligned} UX_1U^\dagger &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \\ &= X_1X_2. \end{aligned} \tag{3.60}$$

Similarly, we see that $UX_2U^\dagger = X_2$, $UZ_1U^\dagger = Z_1$ and $UZ_2U^\dagger = Z_1Z_2$. Now, we can calculate how U conjugates other two qubit operators. For example, $UX_1X_2U^\dagger = UX_1U^\dagger UX_2U^\dagger = (X_1X_2)X_2 = X_1$. Similarly, $UY_2U^\dagger = iUX_2Z_2U^\dagger = iUX_2U^\dagger UZ_2U^\dagger = iX_2(Z_1Z_2) = Z_1Y_2$.

It turns out that any unitary operation taking elements of G_n to elements of G_n under conjugation can be composed from Hadamard, phase and CNOT gates. This set is called the normalizer of G_n, i.e., the set of U such that $UG_nU^\dagger = G_n$.

3.7.3 Measurement in the stabilizer formalism

Beyond the vector spaces and unitary dynamics, measurements can be conveniently described as well. Imagine we take a measurement of $G \in G_n$. We further assume that g is a product of Pauli matrices with no multiplicative factors of -1 or $\pm i$. The system is in the state $|\psi\rangle$, stabilized by $\langle g_1, \ldots, g_n\rangle$. Upon measurement, there are two possibilites,

- g commutes all the generators.
- g anti-commutes with one or more generators. Suppose g anti-commutes with g_1. We can assume that g commutes with all other generators, since if it doesn't commute with one of them (say g_2), we replace g_2 with $g_1 g_2$, with which it commutes now.

In the first case, either g or $-g$ is an element of the stabilizer by the following argument. Since $g_j g |\psi\rangle = g g_j |\psi\rangle = g |\psi\rangle$ for each generator g_j, $g |\psi\rangle$ is in V_S and is therefore a multiple of $|\psi\rangle$. Furthermore, since $g^2 = I$, $g |\psi\rangle = \pm |\psi\rangle$, whence g or $-g$ must be in the stabilizer. We assume g is in the stabilizer, with $-g$ processing analogously. In this instance, $g |\psi\rangle = |\psi\rangle$ and thus a measurement of g yield $+1$ with probability 1 and no disturbance to the system. Thus, the stabilizer is invariant.

In the second case, g anti-commutes with g_1. g has eigenvalues ± 1 and thus, the projectors for the outcomes are $(I \pm g)/2$ respectively, and the probabilities are given by $p(+1) = \mathrm{Tr}\left(\frac{I+g}{2}|\psi\rangle\langle\psi|\right)$ and $p(-1) = \mathrm{Tr}\left(\frac{I-g}{2}|\psi\rangle\langle\psi|\right)$. Using the fact that $g_1 |\psi\rangle = |\psi\rangle$ and $g g_1 = -g_1 g$,

$$\begin{aligned} p(+1) &= \mathrm{Tr}\left(\frac{I+g}{2}|\psi\rangle\langle\psi|\right) \\ &= \mathrm{Tr}\left(g_1 \frac{I-g}{2}|\psi\rangle\langle\psi|\right) \\ &= \mathrm{Tr}\left(\frac{I-g}{2}|\psi\rangle\langle\psi|\right) \\ &= p(-1). \end{aligned} \tag{3.61}$$

Thus, $p(+1) = p(-1) = 1/2$. Suppose $+1$ occurs. The new state is given by $|\psi^+\rangle = (I+g)|\psi\rangle / \sqrt{2}$ which has stabilizer $\langle g, g_2, \ldots, g_n\rangle$. Similarly, if the result -1 occurs, the state will then be stabilized by $\langle -g, g_2, \ldots, g_n\rangle$.

3.7.4 Stabilizer Code Constructions

We now use the stabilizer formalism to describe quantum codes. The basic idea is thus - an $[n, k]$ stabilizer code is defined to be the vector space V_S stabilized by a subgroup S of G_n such that $-I \notin S$ and S has $n-k$ independent and commuting generators, $S = \langle g_1, \ldots, g_{n-k}\rangle$. This code is denoted by $C(S)$.

Given this code $C(S)$, which is 2^k-dimensional, we can choose any 2^k orthonormal vectors in the code $C(S)$ to act as the logical computational basis states. However, more systematic methods exist. Here, we outline one important method.

We first choose operators $\bar{Z}_1, \ldots \bar{Z}_k \in G_n$ such that $g_1, \ldots, g_{n-k}, \bar{Z}_1, \ldots, \bar{Z}_k$ form an independent and commuting set (The construction is shown in the following section). The $\bar{Z}_j$ operator plays the role of a logical Pauli sigma Z operator on logical qubit number j. In other words, the logical computational basis state $|x_1, \ldots, x_k\rangle_L$ is defined to the state with stabilizer

$$\langle g_1, \ldots, g_{n-k}, (-1)^{x_1}\bar{Z}_1, \ldots, (-1)^{x_k}\bar{Z}_k\rangle. \tag{3.62}$$

Similarly, we define $\bar{X}_j$ to be that product of Pauli matrices, such that

$$\begin{aligned} \bar{X}_j\bar{Z}_j\bar{X}_j^\dagger &= -\bar{Z}_j \\ \bar{X}_j\bar{Z}_i\bar{X}_j^\dagger &= \bar{Z}_i \text{ for } i \neq j \\ \bar{X}_j g_k \bar{X}_j^\dagger &= g_k \text{ for all } k. \end{aligned} \tag{3.63}$$

Thus, we see that $\bar{X}_j$ has the effect of a quantum NOT gate acting on the jth encoded qubit. The operator $\bar{X}_j$ commutes with all the generators of the code, commutes with all $\bar{Z}_k$ expect $\bar{Z}_j$ with which it anti-commutes.

We now look at the error-correcting properties of a stabilizer code. Suppose $C(S)$ is a stabilizer code and an error $E \in G_n$ occurs. In the case that E anti-commutes with an element of the stabilizer, the code space $C(S)$ is taken to an orthogonal subspace, and the error can be detected and corrected. If $E \in S$, the error E doesn't affect the state. Thus, the real danger occurs for errors E that commute with all elements of S but is not in S, i.e., $Eg = gE$ for all $g \in S$. The set of $E \in G_n$ such that $Eg = gE$ for all $g \in S$ is known as the centralizer of S in G_n and is denoted by $Z(S)$. Another subset of interest, called the normalizer of S, denoted by $N(S)$, consists of all elements E of G_n such that $EgE^\dagger \in S$ for all $g \in S$. Note that $Z(S) \subseteq N(S)$. However, for any S such that $-I \notin S$, $Z(S) = N(S)$.

Theorem 4 (Error-correction conditions for stabilizer codes)**.** *Let S be the stabilizer for a stabilizer code $C(S)$. Suppose $\{E_j\}$ is a set of operators in G_n such that $E_j^\dagger E_k \notin N(S) - S$ for all j and k. Then $\{E_j\}$ is a correctable set of errors for the code $C(S)$.*

Proof. Without loss of generality, we restrict ourselves to errors where $E_j^\dagger = E_j$. We can split G_n into 3 subsets : S, $N(S) - S$ and $G_n - N(S)$. Since we know that $E_j^\dagger E_k \notin N(S) - S$, there are two possibilities: either $E_j^\dagger E_k$ in S or $E_j^\dagger E_k$ in $G_n - N(S)$. Let P be the projector onto the code space $C(S)$.

In the first case, $PE_j^\dagger E_k P = P$ since P is invariant under multiplication by elements of S.

In the second case, $E_j^\dagger E_k$ must anticommute with some element g_1 of S. Let $g_1, \ldots, g_{n-k}$ be the set of generators of S, so that

$$P = \frac{\prod_{l=1}^{n-k}(I + g_l)}{2^{n-k}} \tag{3.64}$$

Using anti-commutativity gives

$$E_j^\dagger E_k P = (I - g_1)E_j^\dagger E_k \frac{\prod_{l=2}^{n-k}(I + g_l)}{2^{n-k}}. \tag{3.65}$$

But $P(I - g_1) = 0$ since $(I + g_1)(I - g_1) = 0$ and therefore $PE_j^\dagger E_k P = 0$ whenever $E_j^\dagger E_k P \in G_n - N(S)$. Thus, using Theorem 2, we see that the set of errors $\{E_j\}$ satisfies the quantum error-correction conditions, and thus forms a correctable set of errors. □

However, the theorem doesn't specify how to perform the error-correction explicitly. To see the procedure, we take a set of generators $g_1, \ldots, g_{n-k}$ of an $[n, k]$ stabilizer code and $\{E_j\}$ is a set of correctable errors for the code. Error-detection is performed by measuring the generators of the stabilizer g_1 through g_{n-k} in turn, to obtain the error syndrome, which consists of the results of the measurement β_1 through β_{n-k}. If the error E_j occurred, then the error syndrome is such that $E_j g_l E_j^\dagger = \beta_l g_l$. In the case when E_j is the unique error operator having this syndrome, recovery may be simply achieved by applying $E_j^\dagger$. In the case when there are two distinct errors E_j and $E_{j'}$ giving the same error syndrome, it follows that $E_j P E_j^\dagger = E_{j'} P E_{j'}^\dagger$. Thus, $E_j^\dagger E_{j'} P E_{j'}^\dagger E_j = P$, implying that $E_j^\dagger E_{j'} \in S$ and thus, applying $E_j^\dagger$ after the error $E_{j'}$ occurred results in successful recovery. Thus, for each possible error syndrome, pick a single error E_j with that syndrome, and apply $E_j^\dagger$.

We now look at the previous examples, like the bit-flip code and Shor's code, but using the stabilizer formalism.

3.7.5 Three qubit bit-flip code

The three qubit bit-flip code is spanned by $|000\rangle$ and $|111\rangle$, with a stabilizer generated by $Z_1 Z_2$ and $Z_2 Z_3$. By inspection, we see that every possible product of two elements from the error set $\{I, X_1, X_2, X_3\}$ – $I, X_1, X_2, X_3, X_1X_2, X_2X_3, X_1X_3$ – anti-commutes with atleast one of the generators of the stabilizer (expect I which is in S). Thus, from Theorem 4, the set $\{I, X_1, X_2, X_3\}$ forms a correctable set of errors for the three qubit bit-flip code.

Error-detection is effected by measuring the stabilizer generators $Z_1 Z_2$ and $Z_2 Z_3$. For example, if error X_1 occurred, then the stabilizer is transformed to $\langle -Z_1 Z_2, Z_2 Z_3 \rangle$, so the syndrome measurement gives the result -1 and $+1$. This error, and the other possible errors, can thus, be corrected.

3.7.6 Nine qubit Shor code

The stabilier for the Shor code has eight generators as shown below.

Name	*Operator*
g_1	$ZZIIIIIII$
g_2	$IZZIIIIII$
g_3	$IIIZZIIII$
g_4	$IIIIZZIII$
g_5	$IIIIIIZZI$
g_6	$IIIIIIIZZ$
g_7	$XXXXXXIII$
g_8	$IIIXXXXXX$

Table 3.4: Generators for the nine qubit Shor code

The conditions of Theorem 4 can be verified for the error set consisting for I and all single qubit errors. For example, consider single qubit errors like X_1 and Y_4. The product X_1Y_4 anti-commutes with Z_1Z_2, and thus, is not in $N(S)$. Similarly, all other products of two errors are either in S or else anti-commute with atleast one element of S, implying that the Shor code can be used to correct an arbitrary single qubit error.

From direct computation, we see that $\bar{Z} = XXXXXXXXX$ and $\bar{X} = ZZZZZZZZZ$.

3.7.7 Standard form for a stabilizer code

In this section, we show how the construct the logical Z and logical X operations. The construction is much easier to understand if we put the code into a standard form. Consider the check matrix ofr an $[n, k]$ stabilizer code C,

$$G = \left[\mathrm{G}_1 \mid \mathrm{G}_2 \right]. \tag{3.66}$$

This matrix has $n - k$ rows. Swapping rows of this matrix corresponds to relabeling generators and swapping columns on both sides of the matrix corresponds to relabeling qubits. Furthermore, adding two rows corresponds to multiplying generators; it is easy to see that we may always replace a generator g_i by g_ig_j when $i \neq j$. Thus, there is an equivalent code with a different set of generators whose check matrix corresponds to the matrix G where Gaussian elimination has been done on G_1, swapping qubits if needed:

$$\begin{array}{r} \\ r\{ \\ n-k-r\{ \end{array} \left[\begin{array}{cc|cc} \overbrace{I}^{r} & \overbrace{A}^{n-r} & \overbrace{B}^{r} & \overbrace{C}^{n-r} \\ 0 & 0 & D & E \end{array} \right]$$

where r is the rank of G_1. Next, we perform Gaussian elimination on E to obtain

$$\begin{array}{r} \\ r\{ \\ n-k-r-s\{ \\ s\{ \end{array} \left[\begin{array}{ccc|ccc} \overbrace{I}^{r} & \overbrace{A_1}^{n-k-r-s} & \overbrace{A_2}^{k+s} & \overbrace{B}^{r} & \overbrace{C_1}^{n-k-r-s} & \overbrace{C_2}^{k+s} \\ 0 & 0 & 0 & D_1 & I & E_2 \\ 0 & 0 & 0 & D_2 & 0 & 0 \end{array} \right]$$

The last s generators cannot commute with the first r generators unless $D_2 = 0$, and thus, we can assume $s = 0$. Furthermore, we can also set $C_1 = 0$ by taking appropriate linear combinations of rows, so our final check matrix has the form:

$$\begin{array}{r} \\ r\{ \\ n-k-r\{ \end{array} \left[\begin{array}{ccc|ccc} \overbrace{I}^{r} & \overbrace{A_1}^{n-k-r} & \overbrace{A_2}^{k} & \overbrace{B}^{r} & \overbrace{0}^{n-k-r} & \overbrace{C}^{k} \\ 0 & 0 & 0 & D & I & E \end{array} \right]$$

where we have relabeled E_2 as E and D_1 as D. While not unique, any code will have a check matrix of the form above.

Given the standard code of a quantum code, we can define the logical Z operations with ease. Recall that the logical Z operations are k operators independent of the generators and of each other, yet commuting with one another and the generators. We now create a check matrix for these encoded Z operations G_z in the form $G_z = [F_1F_2F_3|F_4F_5F_6]$, where all the matrices have k rows, and column sizes $r, n-k-r, k, r, n-k-r$ and k respectively. We chose these matrices such that $G_z = [000|A_2^T 0 I]$. We now show the that these encoded Z operations satisfy all the required properties.

- Commutativity with the generators. We know that commutativity between two generators g, g' is given by $r(g)\Lambda r(g') = 0$. Thus, commutativity between the generators and the encoded Z operations is given by $G\Lambda G_z^T = 0$. Expanding, we see that this condition is met as $I \times A_2 + A_2 = 0$.

- Commutativity within encoded Z set. We see that the logical Z operations commute with themselves as they're made up entirely of Z and I operations.

- Independence from the generators. The encoded Z operations are independent of the first r generators as no X terms appear in the encoded Z operations. The independence from the remaining $n-k-r$ generators follows from the fact that $(n-k-r) \times (n-k-r)$ identity matrix appearing in the generator check matrix and the lack of the corresponding terms for the encoded Z operations.

- Independence within encoded Z set. The identity matrix in the definition of G_z ensure that the rows are linearly independent of each other, thus, showing that the encoded Z operations are independent of each other.

Similarly, we pick the encoded X operators, with $k \times 2n$ check matrix $[0E^T I|C^T 00]$. The encoded X operations can be shown to be independent of one another and of the generators, commute with the generators, with each other, and $\bar{X}_j$ commutes with all the $\bar{Z}_k$ except $\bar{Z}_j$, with which it anti-commutes.

For example, we calculate the encoded X and Z operations for the Steane code, with check matrix (3.51). We have $n = 7$ and $k = 1$, we see that the $r = 3$ by inspection. The matrix can be brought into standard form by swapping qubits 1 and 4, 3 and 4, and 6 and 7. Then by adding row 6 to row 4, then row 6 to row 5, and finally adding rows 4 and 5 to row 6. The resulting standard for is:

$$\left[\begin{array}{ccccccc|ccccccc} 1&0&0&0&1&1&1&0&0&0&0&0&0&0\\ 0&1&0&1&0&1&1&0&0&0&0&0&0&0\\ 0&0&1&1&1&1&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&1&0&1&1&0&0&1\\ 0&0&0&0&0&0&0&0&1&1&0&1&0&1\\ 0&0&0&0&0&0&0&1&1&1&0&0&1&0 \end{array}\right]. \tag{3.67}$$

Thus, we can read $A_2 = (1, 1, 0)$ and thus, the encoded Z has check matrix $[0000000|1100001]$ which corresponds to $\bar{Z} = Z_1Z_2Z_7$. Since qubits 1 and 4, 3 and 4, 6 and 7 were swapped, this corresponds to an encoded $\bar{Z} = Z_2Z_4Z_6$ in the original code. Previously, we mentioned that $\bar{Z} = Z_1Z_2Z_3Z_4Z_5Z_6Z_7$. This is because $Z_1Z_3Z_5Z_7$ which is an element of the

stabilizer of the Steane code, and thus have the same effect on Steane code states.

Furthermore, we see that $C = (0, 0, 0)$ and $E = (1, 1, 0)$. Thus, the encoded X has check matrix [0001101|0000000] which corresponds to $X_4X_5X_7$, which after swapping, corresponds to $X_1X_5X_6$.

3.7.8 Quantum circuits for encoding, decoding and correction

One important feature of the stabilizer formalism is that their structure enables systematic construction of procedures for encoding, decoding and error-correction. We look at at $[n, k]$ stabilizer code with generators $g_1, \ldots, g_{n-k}$ and logical Z operators $\bar{Z}_1, \ldots, \bar{Z}_k$.

Preparing an encoded $|0\rangle_L^{\otimes k}$, is quite simple. We begin with the state $|0\rangle^{\otimes n}$ and measure each of the observables $g_1, \ldots, g_{n-k}, \bar{Z}_1, \ldots, \bar{Z}_k$. Depending on the measurement outcomes, the resulting quantum state will have the stabilizer $\langle \pm g_1, \ldots, \pm g_{n-k}, \pm\bar{Z}_1, \ldots, \pm\bar{Z}_k \rangle$, with the signs being determined by the measurement outcomes. The signs can now we fixed up by applying products of Pauli operators resulting in the state with a stabilizer $\langle g_1, \ldots, g_{n-k}, \bar{Z}_1, \ldots, \bar{Z}_k \rangle$.

Once $|0\rangle_L^{\otimes k}$ has been prepared, we can prepare an arbitrary computational state $|x_1, \ldots, x_k\rangle_L$ by applying the appropriate $\bar{X}_1, \ldots, \bar{X}_k$.

Decoding is simple as well, but it can be shown that a full decoding is often not necessary. The techniques of fault-tolerant quantum computation can be used to perform logical operations directly on encoded data.

The error-correction procedure has already been described: measure the generators, obtain the error syndrome and perform the required recovery operation. Performing measurement can be done in a systematic way, as shown in [NC11].

Chapter 4

Conclusion

In this work, we looked at several features and characteristics of open quantum systems and quantum error-correction. We explored the system-bath approach and derived the Kraus OSR representation for a channel. We look at the action of several characteristic channels on a single qubit.

We took a deep look at several aspects of quantum error-correction. We began with the problems with quantum error-correction systems, and the three-qubit bit- and phase-flip codes. Combining these codes, we look at the Shor code, capable of correcting arbitrary errors on a single qubit, by discretizing the errors. We saw the construction of quantum codes using classical linear codes, most notably being the Steane code. We finally look at the stabilizer formalism. We saw how vector spaces, dynamics and measurements are described in the stabilizer formalism. Lastly, we constructed several codes using the stabilizer formalism and looked at circuits for encoding, decoding and correction.

Bibliography

[Kay18] Alastair Kay. Tutorial on the quantikz package, 2018.

[Lid19] Daniel A. Lidar. Lecture notes on the theory of open quantum systems, 2019.

[NC11] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, USA, 10th edition, 2011.