Quantum Error Recovery on Permutation Invariant Spin Codes

By

Daniel Madden

A thesis submitted to Macquarie University

for the degree of Master of Research

Department of School of Mathematical and Physical Sciences

July 2022

Examiner's Copy

Typeset in LaTeX 2_ε.

Except where acknowledged in the customary manner, the material presented in this thesis is, to the best of my knowledge, original and has not been submitted in whole or part for a degree in any university.

Daniel Madden

Acknowledgements

The opportunity to undertake a research degree has been a dream of mine since before I can remember. I had always dreamed of understanding the complexities of the universe, beholding a truth hidden to the world that I could call my own. But of course dreams do not just fulfil themselves, it was with the help and provision of an extraordinarily many people that I have been able to stand where I am today and to these people I wish to give my thanks.

Without guidance one easily becomes lost in the realms of research, as such I would have been helpless without a mentor to guide me through my research, and for this I gives the deepest thanks to Gavin Brennen for his continual support and guidance. Through lockdowns, breakdowns and shutdowns Gavin has continued to provide guidance in my research and it is to this that I owe the completion of this Masters degree. Simple thanks feels insufficient for the continual work he has put in and it fills me with pride to be able to call myself his student.

To cultivate research and discovery an environment of learning and support is necessary, and for this I would like to express my deepest gratitude to the research team and support staff at Macquarie University who have enabled me to study to the fullest of my capabilities. My thanks go out to all the Research staff in the Department of Physics who have helped me through many problems trivial and pertinent. Further I would like to specifically thank my MRes advisor Joanne Dawson. Joanne has supported me tremendously during my thesis study, ensuring that nothing slows the pace of my research, and for their continued support I give my thanks.

Additionally I would like to thank my friends and family who have been there at every step of the way to encourage my successes and comfort my failures. I have been extremely fortunate to be blessed a large support network around me and without it I feel it would have been impossible to raise myself after all my falls. For the continual support and good times

that kept me moving I thank you all.

Finally I wish to thank the single person in my life who has supported more than anyone, with countless late night lifts home, preparing food for me when I was too busy to do it myself, and never letting me stop and give up. Thanks mum, luv ya, and I guess you too dad.

Abstract

In this thesis we examine the concept of error recovery as an alternative to error correction in quantum computation and metrology on permutation invariant spin states. We investigate the encoding of Dicke states using Geometric Phase Gates and discuss the current applicability of these systems. We then discuss error correction in quantum systems and examine two recent relevant error correcting codes for multiple-spin systems and compare their performance to two known error correcting codes for multi-level qudit systems. By introducing the concept of error recovery we then analyse these codes performance under multiple simulations, discussing the relative performance of each code along with the advantages and disadvantages of the simulated recovery methods. Within the discussion of error recovery methods we touch on computational requirements for error recovery, observing that not all available methods are likely to be realistically implementable. Further we discuss the models of noise for quantum systems, again touching on the computational requirement for each, as this turns out to be a key limiting factor. With the culmination of our analysis methods we hope that our simulations can be helpful for testing and comparing codes in the future and that our analysis on the applicability of different recovery methods will aid the development of quantum systems.

Contents

List of Figures

List of Tables

The best that most of us can hope
to achieve in physics is simply to
misunderstand at a deeper level.

Wolfgang Pauli

1 Introduction

With the current rapid development of quantum computing, it is inevitable that some time in the near future quantum computers will play an integral role in everyday life. Quantum computers have been shown to be able to surpass classical computers in specific tasks [2], hence garnering a lot of attention. These improvements could revolutionise computation and may overhaul many current algorithms [44, 26]. But quantum computers face a major flaw with current systems, environmental and system noise, which generate errors rendering them useless [30]. This stage in development of quantum computing mirrors the early development of classical computing technology where fragile vacuum tubes were used, which would break regularly making calculations fail [16].

In order to overcome the barriers of errors in current systems researchers have mirrored the methods of early classical computing and developed error correcting codes such as the 9-qubit Shor [45] code and the 7-qubit Steane code [46]. Since the introduction of these codes quantum computing has advanced significantly, with far more developed codes [9, 31], and far more stable qubits [23].

This thesis will focus on the application of error recovery to permutation invariant spin states, i.e. states where it is not necessary to distinguish between individual qubits. This specification will introduce a number of benefits and drawbacks, that will be discussed at length, but they overall introduce an interesting system for computation. We will also investigate the possibility of error recovery in place of complete error correction, and utilise computational simulations to motivate our conclusions. Expanding on this we also discuss a number of error models, representing noise in quantum systems.

1.1 Spin States

Our work uses the Dicke formalism to mathematically represent the behaviour of quantum states. Dicke states, introduced in 1973 by Klaus Hepp and Elliot Lieb [28], was named after pioneering physicist Robert Dicke, who first predicted the superradiant behaviour of the then un-named Dicke states, a result still widely studied to this day [13].

A commonly discussed modern use for these Dicke states is in quantum sensing, where the high entanglement possible from Dicke states makes them highly valuable [17]. This has become apparent through the derivation of the quantum Cramér-Rao bound [11], which has shown that by using highly entangled states the classical limit $\left(\frac{1}{\sqrt{N}}\right)$ can be overcome, limited instead by the Heisenberg limit $\left(\frac{1}{N}\right)$. This offers a broad range of improvements in technology and has driven recent research in the field.

But despite the known behaviour of Dicke states, their complex nature and the difficulty of preparing quantum states meant that until as recent as 2009 [38] even preparing a 6-qubit Dicke state yielded an unusable fidelity and was only performed as a proof of concept. Since then the understanding around Dicke states and the system that prepare them have improved dramatically with the IonQ project claiming to have up to 22 algorithmic qubits able to be prepared.

Dicke states have been experimentally realised in a variety of experimental setups such as trapped ions [32], atoms [47] and photons [39], thus representing a quantifiably applicable system to investigate. Additionally recent work by Johnsson et al. has developed a "tool box" for quantum control on Dicke states [29], introducing a method to prepare states on N qubits with a number of control gates scaling as $O(N)$, a vast improvement on the previously best known method with scaling $O(N(\frac{N}{2}+M))$ [7]. Moreover, the former method does not require

addressability of spins.

Due to their permutation symmetry there have been several proposals to encode quantum information in Dicke subspaces [41]. From this field has risen error correction codes, which will be discussed in detail in section 1.2. The first paper to actively discuss codes for Dicke states was by Ouyang in [35], with a focus on a code that could be varied to fit the exact noise model of a system. The code was developed with exact correction of a known number of qubits in mind and has been shown to be able to correct t errors for a system with $(2t + 1)^2$ qubits. In a 2021 paper by Gross [25], a new and highly efficient quantum code was found, which focused not on the principal of exact error correction but error recovery, which will be discussed in section 1.3. This code was derived from the binary octahedral group (2O), with the conceptual drive that by representing the binary octahedral group with Clifford gates the derived code would be naturally implementable. This development has further improved the ability of spin codes as the Gross code encodes 17 physical qubits, which is a realistically preparable amount.

1.2 Error Correction

The concept of error correction in computing dates back to the advent of the classical computer. As systems were not as reliable as modern computers a solution was needed to identify and remove errors from codes. Initial concepts as to the methods to avoid errors mainly revolved around spotting when an error had occurred, not fixing them. A simple example of this is the parity bit, where a single bit in a block of bits is set to represent the parity of the entire block. Using this method a large number of bits could be encoded with little redundancy, but if an error was detected there was no way to correct it. The first major breakthrough in correction of these errors was Hamming's code [27], a parity based correction code with a redundancy of $n + 1$ for a block of length $N = 2^n$. The code utilised n parity bits which would each represent the parity of a section of bits that would make up half the code. This way by testing the parity of each redundant bit we halve the number of possible locations for an error to be. By using $\log_2(N)$ parity bits we can guarantee the location of a single error in the code. Shortly after the publication of the Hamming code, Shannon published his landmark paper [43], forming the defined mathematical representations that would be used in the field for decades to come. Other encoding formats were also discovered

such as the Reed-Solomon Codes [40] and the more recent Turbo Codes [8].

The next ground-breaking development in the field of error correction was Shor's introduction of a quantum error correcting code [44]. This code was a simple proof of concept to encourage the field of quantum computing, as correcting a quantum state is far more complex than a classical state. This is due to two main reasons, firstly the no-cloning theorem [36] states that it is impossible to create an exact replica of a quantum state by measurement. This prevents obvious encoding schemes such as a repetition code, where you send three qubits and take a majority vote on the result. And secondly, the measurement of a quantum state changes that state, meaning that you can not perform error correction after measurement. With these two restrictions in mind it was the understanding of entanglement, a property unique to quantum states, that enabled the production of error correcting codes for quantum systems.

These error correcting codes were given a rigorous mathematical model in the landmark paper by Knill and Laflamme [30], which developed the tests for ensuring that a code was capable of error correction. We discuss this in further detail in section 3.1.

1.3 Optimal Recovery

With a modern understanding of Quantum Error Correcting Codes it is possible to identify the ability of a code to correct a known number of errors [30], and to compare the efficiency of qubit encryptions for different codes. But these methods are developed for the purpose of a general recovery, without knowledge of the specific error channel, logically then we may be able to find some improvement by catering to the specific error map at hand.

From this thought arises the concept of error recovery; not trying to perfectly correct a set number of errors, but instead recover your state as closely as possible to the goal state. The mathematically optimal method was first proposed in [3], where the authors suggest the use of a semidefinite program optimisation to perform an optimisation on completely positive maps. This approach manipulates the optimisation problem of maximising the entanglement fidelity under some noise with a recovery operation into a semidefinite program, a known mathematical optimisation procedure [10]. Along with this method other methods for analysis of the optimal recovery and near optimal recovery are known and will be investigated in this thesis.

1.4 Thesis Structure

We start by introducing the physics and mathematical representation of Dicke states in Chapter 2. This will lay the groundwork for understanding the notation and representations used throughout this thesis. Chapter 2 builds up from the bare bones beginnings, defining the behaviour of a single spin, comparing it to two spins and extrapolating to multiple spins, and defining the operators for a multiple spin system. Additionally in Chapter 2 we investigate the application of Geometric Phase Gates for controlling a spin system and discuss recent developments in the realisation of quantum spin systems.

In Chapter 3 we discuss the requirements for quantum error correction and introduce the error codes that will be examined in this thesis. We also define some of the different measures of code performance that can be used t characterise quantum error recovery.

This follows into Chapter 4, where we discuss the error models studied in this thesis. These models are compared in terms of their simplicity, applicability, and ability to be simulated. The latter of which turned out to be a large restriction within the scope of this project. Completing our theory in Chapter5 we discuss the different methods of error recovery and derive the optimal recovery, in the process introducing a powerful measure of code performance where ancillary systems are used. These methods define the procedure for simulating error recovery used throughout the rest of the thesis.

In Chapter 6 we simulate the recovery of the error codes in the python package QuTip and discuss and compare the codes performance and the performance of the different recovery simulations. Finally in Chapter 7 we summarise our findings, outlining key conclusions and explore the possible directions this work could take in the future.

The important thing is not to stop questioning.

Albert Einstein

2

Dicke States

2.1 Single Spin Systems

In order to consider a group of spins we will first consider the simplest size, the mechanics of a single two-level system. These systems are widely used in introductory examples of quantum systems [42, 24] as they can describe simple systems that can be solved without the need for computers or approximations. Further the two-level system is used as a quantum analogue of a bit, known as a qubit [42]. Thus while being a simple system, it is important not to overlook its importance in the fundamental structure of quantum mechanics. For the purpose of simplicity, in this thesis we will consider each qubit to be a two level system with angular momentum always considered about the z-axis. Our notation will give the angular momentum -1/2 as $|0\rangle$ or referred to as pointing down($\downarrow$) and the angular momentum +1/2 as $|1\rangle$ or as pointing up ($\uparrow$). As a matrix representation these states are given as $|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$

$|1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$. These Dirac representation values of 0 & 1 are given to mimic the classical concept of a bit, and hence we require operations to apply to these bits. The basic operations on a single qubit are given by the Pauli matrices:

$$\sigma_x = |0\rangle\langle 1| + |1\rangle\langle 0|, \sigma_y = i|1\rangle\langle 0| - i|0\rangle\langle 1|, \sigma_z = |0\rangle\langle 0| - |1\rangle\langle 1|, \sigma_0 = |0\rangle\langle 0| + |1\rangle\langle 1| = I \tag{2.1}$$

The Pauli matrix σ_x is an operation common with classical computing, known as a bit flip takes a 1 to a 0 and visa versa, but the similarities stop there with the other Pauli matrices introducing new operations to form the single-qubit quantum gates [24]. Further to the concept of error correction in quantum systems these quantum gates make up the set of basic errors that we aim to correct.

2.2 Systems of Many Spins

As with many models of standard qubits this representation scales in size as dimension 2^n where n is the number of qubits in the system. This poses a problem for optimisation as computational efforts will rapidly begin to require more memory and hence more processing per a qubit. In order to perform optimisation on a larger system we reduce our representation by taking the spins to be permutation invariant, i.e. we cannot tell apart the individual spins and only consider the total spin of the system. We also introduce the operator $\hat{J}^2$ such that:

$$\hat{J}^2 = \sum_{j,k} \hat{J}_j \cdot \hat{J}_k \tag{2.2}$$

Where $\hat{J}_k = \frac{1}{2}(\sigma_x^k \hat{x} + \sigma_y^k \hat{y} + \sigma_z^k \hat{z})$ with eigenvalues $J(J+1)$ for $0 \leq J \leq \frac{N}{2}$. Using this measurable value we can represent the states as a $|J, M\rangle$ where J is the total angular momentum and M is the total spin, in this model each downward spin ($|0\rangle$) adds $-1/2$ to the value M and each upwards spin ($|1\rangle$) adds $+1/2$ to the value M, with no regard for the order they are added. This gives the three symmetric states, where J takes the maximum value of $N/2$ of the 2 spin-1/2 systems as $|1,1\rangle = |11\rangle, |1,0\rangle = \frac{1}{\sqrt{2}}(|10\rangle + |01\rangle), |1,-1\rangle = |00\rangle$. The antisymmetric part of the system given by $J \neq \frac{N}{2}$ for N being the number of qubits, given as$|0,0\rangle = \frac{1}{\sqrt{2}}(|10\rangle - |01\rangle)$.

For a system with multiple spins we begin to see a clear advantage in the size of our representation, as the matrix dimension of our representation can be calculated as a block matrix of size $O(N^2)$ [4]. Further for our study we focus in on the symmetric space, with $J = \frac{N}{2}$, which has a representation of dimension $O(N)$. The symmetric states form a qudit-like system with a group of spins represented as a single $N + 1$ level system. This behaviour results in a highly entangled group of spins which assume permutation invariance for the spins in a system, assuming that the individual spins are indistinguishable from each other. The Dicke state is given by the total spin value of the system M, written as:

$$|J, M\rangle = \sqrt{\binom{2J}{J+M}^{-1}} \sum_{\pi} |\pi(\underbrace{0, ..., 0}_{\text{(J-M)}}, \underbrace{1, ..., 1}_{\text{(J+M)}})\rangle \tag{2.3}$$

Where $-J \leq M \leq J$ and π is the sum is over all the distinct permutations of the given number of 0's and 1's, and this is normalised by the combinatorial function (also known as the binomial coefficient). To describe this system without moving out of the symmetric space we must act on all spins at once identically, such that the system evolves simultaneously, hence we define spin operators:

$$\begin{aligned} \hat{J}_{\pm} &= \sum_{i}^{N} \sigma_i^{\pm} \\ \hat{J}_z &= \sum_{i}^{N} \frac{\sigma_i^z}{2} \end{aligned} \tag{2.4}$$

With $\sigma_i^+ = |1\rangle\langle 0|_i, \sigma_i^- = |0\rangle\langle 1|_i$ and σ_i^z are the operators acting on the ith spin. From this we can define the spin operators $\hat{J}_x$ and $\hat{J}_y$ such that $\hat{J}_x = \frac{\hat{J}_+ + \hat{J}_-}{2}$, $\hat{J}_y = \frac{\hat{J}_+ - \hat{J}_-}{2i}$. With this we have the well-known commutation relations for the algebra su(2):

$$\begin{aligned} &[\hat{J}_+, \hat{J}_-] = 2\hat{J}_z, \quad [\hat{J}_z, \hat{J}_{\pm}] = \pm\hat{J}_{\pm} \\ &[\hat{J}_j, \hat{J}_k] = i\epsilon_{jkl}\hat{J}_l \quad \text{for} j, k, l \in \{x, y, z\} \end{aligned} \tag{2.5}$$

Where ϵ is the Levi-Civita symbol in three dimensions. We also note that the collective operators act on our Dicke states as:

$$\begin{aligned} \hat{J}_{\pm} |J, M\rangle &= \sqrt{(J \pm M + 1)(J \mp M)}\, |J, M \pm 1\rangle \\ \hat{J}_z |J, M\rangle &= M |J, M\rangle \end{aligned} \tag{2.6}$$

In the symmetric space we represent our Dicke states as a vector with dimension $2J + 1$.

2.3 Unitary and State Synthesis in Dicke Subspace

In order to develop a working system implementing codes on the Dicke space it is important to have a suitable method for preparing and manipulating these states. To do this we will examine the geometric phase gates (GPG) method by Johnsson et al. in their recent paper [29]. The method is designed for use with spins systems that have a coupling to a bosonic mode, such as trapped ion qubits coupled to a motional mode or nitrogen-vacancy centre qubits coupled to a microwave cavity mode. The spins are prepared using a series of incremental "phase gates" performed as a closed loop in phase space. These phase gates perform a series of iterative steps, traversing phase space using the displacement operator $D(\alpha) = e^{\alpha\hat{a}^\dagger - \alpha\hat{a}}$. We assume a dispersive coupling between the spins and the bosonic mode of the form $V = g\hat{a}^\dagger\hat{a}J_z$, which allows for the use of the rotation operator $R(\theta J_z) = e^{i\theta\hat{a}^\dagger\hat{a}J_z}$. These operators are used to perform a closed loop as:

$$\begin{aligned} U_{GPG}(\theta,\phi,\chi) &= D(-\beta)R(\theta J_z)D(-\alpha)R(-\theta J_z)D(\beta)R(\theta J_z)D(\alpha)R(-\theta J_z) \\ &= e^{-i2\chi\sin(\theta J_z+\phi)} \end{aligned} \tag{2.7}$$

Where $\theta = arg(\alpha) - arg(\beta)$ and $\chi = |\alpha\beta|$. The operator $R(-\theta J_z) = e^{-i\pi J_x}R(\theta J_z)e^{i\pi J_x}$ is advantageous as it allows to dynamically decouple environmental interactions giving rise to phase error during the GPG.

By then performing a series of these geometric phase gates, $W(\ell) = \prod_{k=1}^{N/2} U_{GPG}(\theta_k, \phi_k(\ell), \chi)$ we reach a total operation:

$$W(\beta,\ell) = \sum_{M=-J}^{J} e^{-i2\chi\sum_{k=1}^{N/2}\sin(\theta_k M+\phi_k(\ell))} |J,M\rangle\langle J,M| \tag{2.8}$$

for $\ell \in [0, N]$ with the variables $\theta_k, \phi_k(\ell), \chi$ defined as:

$$\begin{aligned} \theta_k &= \frac{2\pi k}{N+1} \\ \phi_k(\ell) &= \frac{2\pi k(N/2-\ell)}{N+1} \\ \chi &= \frac{\beta}{(N+1)} \end{aligned} \tag{2.9}$$

with a number of steps $\frac{N}{2}$ for N even and N steps for N odd. This projector can be simplified down to the projector $W(\beta,\ell) = e^{-i\beta|J,\ell-J\rangle\langle J,\ell-J|}$, hence we are able to apply a β phase shift to any given state $|J,M\rangle\langle J,M|$.

Given this toolbox it can be shown that one can synthesise any arbitrary operator in the space. We first consider the unitary U under a spectral decomposition, given as $U = \sum_{k-1}^{N} e^{i\lambda_k|\lambda_k\rangle\langle\lambda_k|}$, where $\{|\lambda_k\rangle\}$ form an orthonormal basis on the Dicke subspace. We then utilise the properties of the operations $W(\beta,\ell)$, starting with the spectral decomposition:

$$U = \prod_{k=1}^{N-1}[K(\lambda_k)e^{i\lambda_k|J,-J\rangle\langle J,-J|}K^{\dagger}(\lambda_k)] \tag{2.10}$$

Where $K(\lambda_k)$ is a unitary extension of the state synthesis, mapping $|J,-J\rangle \rightarrow |\lambda_k\rangle$. Using these we can construct the unitary $K(\lambda_k)$ by decomposition as:

$$\hat{K} = \prod_{s=1}^{N-1}[e^{i\beta_s\hat{J}_y}U_{GPG}(\theta_s,\phi_s\chi_s)]e^{-i\hat{J}_y\frac{\pi}{2}}U_{GPG}(\frac{\pi}{2},0,\frac{\pi}{4})e^{i\hat{J}_y\frac{\pi}{2}} \tag{2.11}$$

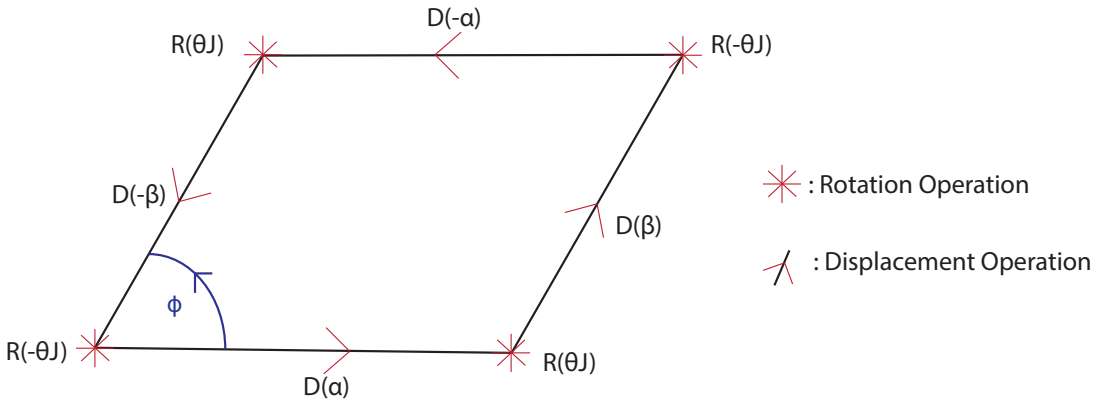


Figure 2.1: A closed loop of a single GPG step in phase space.

This unitary $K(\lambda_k)$ can also be used to prepare any given state $|\psi\rangle$, starting from the state $|\frac{N}{2},-\frac{N}{2}\rangle$ the state is constructed as:

$$|\psi\rangle = \prod_{s=1}^{N-1}[e^{i\beta_s\hat{J}_y}U_{GPG}(\theta_s,\phi_s\chi_s)]e^{-i\hat{J}_y\frac{\pi}{2}}U_{GPG}(\frac{\pi}{2},0,\frac{\pi}{4})e^{i\hat{J}_y\frac{\pi}{2}}|\frac{N}{2},-\frac{N}{2}\rangle \tag{2.12}$$

Using this method we derive the values of $\{\beta_s,\theta_s,\phi_s,\chi_s\}_{s=1}^{N-1}$ for the quantum codes used in this project in the Appendix.A.

2.4 Physical Implementations

A primary reason for our focus on the study of symmetric states in particular is motivated by the progress in the production of spins states in current experimental work. The key requirement for implementation of the Geometric Phase Gate method, the dispersive coupling between the spins in a system and an oscillator, is present in a range of emerging technologies in the quantum landscape [29].

The recent start-up IonQ has claimed to have created a trapped ion system with 79 qubits, capable of single qubit gates, and a more computational system of 11 qubit capable of two qubit gates [51]. These systems are reaching a size where error correction and recovery are becoming realistic, and with their announcement of a quantum computer with 22 "Algorithmic Qubits" processes such as the geometric phase gate may very well see implementation in the near future.

Not only is the Universe stranger
than we think, it is stranger than
we can think.

Werner Heisenberg

3 Error Codes

One of the core requirements for systems attempting quantum information tasks like metrology, quantum communications, and computing is a consideration for the mitigation of errors. Errors in quantum systems arise from near immeasurably many sources, from cosmic radiation to energy level decay. These errors are caused by such a broadly range of sources that complete prevention is unrealistic [15]. Compared to classical error correction we also encounter the obstacle of the no-cloning theorem, meaning that we cannot take an exact copy of a quantum state to protect it from errors as you would in simple classical systems, such as the repetition code.

3.1 The Theory of Quantum Error Correcting Codes

The fundamental structure and requirements for a quantum error correcting code was discussed by Knill and Laflamme [30]. From this paper we draw the definition of code performance, the fidelity and error.

The paper begins by defining a quantum error correcting code as the pair C (the code) and $\mathcal{R}$ the recovery operator. This structure defines a quantum code in a complete form, defining our error correction through gate actions in the form of recovery operators, this gives rise to a robust mathematical model we can easily examine to ensure a code is error correcting. We define an error correcting quantum code as a mapping of a logical state (denoted as $|\psi\rangle \rightarrow |\psi_L\rangle$) onto a larger dimension Hilbert space, often called the physical state.

The noise on a system is defined as a completely positive trace preserving map in Kraus form with operators $\mathcal{E}$ given as:

$$\mathcal{E}(\rho) = \sum_i E_i \rho E_i^\dagger \tag{3.1}$$

Such that the operator E_i satisfies the trace norm condition $\sum_i E_i^\dagger E_i = \mathcal{I}$.

This is used to define the conditions for error correction set forward in the paper, requirements for a given code under the expected error model such that error correction can occur. For the code with logical states $|i_L\rangle, |j_L\rangle \in C$ and an expected error $\mathcal{E}$ with operators $E_a, E_b \in \mathcal{E}$

$$\langle i_L| E_a^\dagger E_b |i_L\rangle = \langle j_L| E_a^\dagger E_b |j_L\rangle \tag{3.2}$$

$$\langle i_L| E_a^\dagger E_b |j_L\rangle = \alpha_{ab}\delta_{ij} \tag{3.3}$$

These equations represent two important requirements for an error correcting code, Eq.3.2 stating that overlap of error on the logical states are equal, ensuring that the recovery operation on the system can act equally on any logical state, hence not needing to measure the state, as this would ruin the result. Eq.3.3 further requires that each logical codeword maps to a unique state under the noise model, this intuitively means that no two inputs will result in the same output, i.e. that each codestate after the error is applied is has a unique recovered state.

We then need an operation to return our erroneous code states to logical ones, for this we define the recovery operator $\mathcal{R}$ as the completely positive trace preserving map:

$$\mathcal{R}(\hat{\rho}) = \sum_i R_i \hat{\rho} R_i^\dagger \tag{3.4}$$

The most general measure of code performance can be constructed as the fidelity, defined by the equation:

$$F(C, \mathcal{R} \circ \mathcal{E}) = \min_{|\Psi_L\rangle \in C} \sum_{i,j} \left| \langle \Psi_L | \, R_i E_j \, | \Psi_L \rangle \right|^2 \tag{3.5}$$

This equation gives the absolute value squared of the expectation value for the error and subsequent recovery operation, returning how close we expect our final state to be to our initial. From this we can draw a mathematical description of a complete error correcting code, as a code (C) and recovery operator ($\mathcal{R}$) pair, acting to correct a known error ($\mathcal{E}$), that take the fidelity in the given system of errors to 1. This value of fidelity depends heavily on your error model, assuming a known number of error occurs at one time in a system gives you the possibility to objectively correct your code, reaching $F = 1$. By assuming your system has an error occur on any given qubit with some probability γ it becomes impossible to guarantee that you fully correct your system [15] as, low in probability as it might be, you take there to exist some finite chance that all of your qubits have errors, hence the space of your error map spans all possible qubit states, and is not fully recoverable. Hence in order to maintain our information, we instead aim to optimise the fidelity of our system, rather than aim for a perfect fidelity.

3.2 Permutation Symmetric Error Correcting Codes

3.2.1 Gross Spin Code

The spin code introduced by Gross [25] built upon the conceptual image of the spins states as a representation of the binary-octahedral group ($2O$) is a useful example. $2O$ is the double cover of the rotational symmetries of the cube, the six faces of the cube can be identified with the one qubit pure states $\{|\pm_x\rangle\langle\pm_x|, |\pm_y\rangle\langle\pm_y|, |\pm_z\rangle\langle\pm_z|\}$. The group elements then act under conjugation with the Pauli operators to permute them, and the double cover is introduced from the multiplicity ± 1 that arises from this. The $2O$ group, represented in dimension $(N+1) \times (N+1)$ matrices, is generated by multiplication of the phase and Hadamard gates, respectively given as:

$$\begin{aligned} S &= e^{-i\frac{\pi}{2}\hat{J}_z} \\ H &= e^{-i\frac{\pi}{\sqrt{2}}(\hat{J}_x + \hat{J}_z)} \end{aligned} \tag{3.6}$$

The code is constructed by taking the projector onto the lowest dimensional irreps of the $2O$ group from the abstract representation of the group SU(2), with group elements $g \in SU(2)$ given as $g = e^{i\theta\hat{n}\cdot\frac{\vec{\sigma}}{2}}$, for the unit vector $\hat{n} = (\sin(\alpha)\cos(\beta), \sin(\alpha)\sin(\beta), \cos(\alpha))$. The spin-$J$ representation is:

$$D^J\left(exp(-i\theta\hat{n}\cdot\frac{\vec{\sigma}}{2})\right) = exp(-i\theta\hat{n}\cdot\hat{\vec{J}}) \tag{3.7}$$

Giving the physical unitaries $D^j(g) = e^{i\theta\hat{n}\cdot\hat{\vec{J}}}$. The projectors onto the irreps of $2O$ take the form:

$$P_\nu = \frac{\dim(\nu)}{|2O|}\sum_{\{g\in 2O\}}\chi_\nu(g) * D^j(g) \tag{3.8}$$

For the ν^{th} irrep of $2O$, $\chi_\nu(g) = Tr(D^\nu(g))$ is the character of a group element in the irrep ν and $D^j(g)$ is the element $g \in SU(2)$ in the j^{th} irrep.

Upon inspection the $2O$ group has two particular irreps of note, each with dimension 2, $\nu = \rho_4$ which performs the mapping S→S and H→H and $\nu = \rho_5$ which performs the mapping S→-S and H→-H, thus they are also inequivalent mappings. This inequivalence means that they have separate codespaces but as they are only a factor of -1 different and have identical projective action, they hence seem identical when acting separately.

The codewords can be constructed by taking the +1 eigenspace of the logical z operator given by:

$$\hat{\sigma}_w := P_\nu\left(iD^J(e^{-i\pi J_z})\right)P_\nu \tag{3.9}$$

The codewords can then be optimised over the two relevant irreps, while satisfying the condition $\langle\phi|\hat{J}_z|\phi\rangle = 0$ by taking the superposition:

$$|\phi\rangle = a\,|0_{\rho_4}\rangle + \sqrt{1-a^2}e^{i\phi}\,|0_{\rho_5}\rangle \tag{3.10}$$

This will give an explicit form for a in terms of ϕ, over which optimisation can occur. Gross has claimed in his paper that this code has displayed superior error recovery ability compared to many commonly known codes; we will do comparisons of our own to verify this result.

3.2.2 GNU Code

An example of a recently discussed error correcting code in quantum metrology is the GNU code developed by Ouyang. The GNU code [35] is a variable code which utilises an adjustable

code structure, performing an exact correction of a known number of errors. This allows for a general formula for the generation of codes of many different sizes. The code has the form:

$$
\begin{aligned}
|0_L\rangle &= \sum_{\substack{l \text{ even} \\ 0 \geq l \geq n}} \sqrt{\frac{\binom{n}{l}}{2^{n-1}}} \, |gnu, gl\rangle \\
|1_L\rangle &= \sum_{\substack{l \text{ odd} \\ 0 \geq l \geq n}} \sqrt{\frac{\binom{n}{l}}{2^{n-1}}} \, |gnu, gl\rangle
\end{aligned}
\tag{3.11}
$$

Where the total number of qubits in the system is given by the product of the variables gnu. The first variable g introduces a spacing between the used energy levels of the logical qubits by a factor g, this allows correction of t boosting or damping events, given by σ_+ and σ_- respectively, where $2t + 1 = g$. This is a clear result as with gap $2t + 1$ there are t possible shifts in angular momentum projection M that can be unambiguously corrected. The variable n introduces repetition into the code, the total number of non-zero terms between the logical codewords is equal to $n + 1$. These repetitions are useful in correcting phase flip errors, as the weighting of a phase flip is proportional to the energy level of a state, hence by having multiple energy levels for each logical codeword, a phase flip will transform each possible code state to a unique error state. Hence as each error type shifts the code state to a unique state we get the relation $\langle i| E_a^\dagger E_b |j\rangle = \delta_{ij}\delta_{ab}$ which fulfills both of the error correcting conditions. The variable u is used for fitting the state to the model, and for more complicated variations of the code, especially where there is a bias towards different error types. Ouyang summarises that for the case that $g = n = 2t + 1$ and $u \geq 1$ the gnu code corrects t arbitrary qubit errors. This is the structure we will focus on for our simulations, as they are the most general of the gnu error correction codes. A (4,4,1) gnu code uses 16 qubits, which draws the closest of any $g = n, u = 1$ gnu code to the 17-qubit Gross code, hence this will be our focus. This code is given by:

$$
\begin{aligned}
|0_L\rangle &= \frac{|\frac{16}{2}, \frac{-16}{2}\rangle + \sqrt{6}\,|\frac{16}{2}, \frac{0}{2}\rangle + |\frac{16}{2}, \frac{16}{2}\rangle}{\sqrt{8}} \\
|1_L\rangle &= \frac{2\,|\frac{16}{2}, \frac{-8}{2}\rangle + 2\,|\frac{16}{2}, \frac{8}{2}\rangle}{\sqrt{8}}
\end{aligned}
\tag{3.12}
$$

3.2.3 Qudit Codes

A qudit is a computational quantum object alternative to the qubit with a higher number of observable dimensions. The larger number of dimensions increase the information that can be stored in a singular object, allowing for a reduction in the complexity required to perform a range of quantum algorithms [50]. While having possible physical differences between a true qudit and a multi-level representation of a qubit system, it is none the less interesting to compare the behaviour of spin codes and known qudit codes, as Dicke states are a representation of a qudit state space. For this reason we will include two known qudit codes, the qudit GKP code and the minimal qudit code.

The minimal qudit code is constructed as the majority voting result of the minimal amplitude code and the minimal phase codes. The minimal amplitude code is derived for the purpose of correcting k "amplitude shift" errors [37], generated by $\hat{J}_x$ in our system, for a system with a Hilbert space of size $4k + 2$. Conversely the minimal phase code is derived as the Fourier transform of the minimal amplitude code and corrects k "phase shift" errors, generated by J_z in our system. As we can only perform error recovery on one code at a time we choose the minimal phase code, as it has been used in similar studies [25], given as:

$$\begin{aligned} |0_L\rangle &= \sum_{M=-J}^{J} \sqrt{\frac{1}{N+1}}\,|J, M\rangle \\ |1_L\rangle &= \sum_{M=-J}^{J} (-1)^{M+J} \sqrt{\frac{1}{N+1}}\,|J, M\rangle \end{aligned} \tag{3.13}$$

The GKP code is a well-known code, designed for use in quantum harmonic oscillators, systems with infinitely many energy levels. For our work we consider a finite energy level analogy for the GKP code, known as the Qudit GKP code. The Qudit GKP code is designed to protect against a known number of "amplitude shifts" and "phase shifts", simultaneously contrary to the minimal qudit codes focus on individual error types. It does this in a similar method to the GNU code, by leaving energy level spacings to protect against amplitude shifts and multiple repetitions to protect against phase shifts. For a system with $N + 1 = 2r_1r_2 = d$ qubits the code is given as[14]:

$$
\begin{aligned}
|0_L\rangle &= \frac{1}{\sqrt{r_2}} \sum_{k=0}^{r_2-1} |J, -J + 2kr_1\rangle \\
|1_L\rangle &= \frac{1}{\sqrt{r_2}} \sum_{k=0}^{r_2-1} |J, -J + (2k+1)r_1\rangle
\end{aligned}
\tag{3.14}
$$

Which for the 17 qubit (or 18 level) system is given as:

$$
\begin{aligned}
|0_L\rangle &= \frac{1}{\sqrt{3}}(|\frac{17}{2}, -\frac{17}{2}\rangle + |\frac{17}{2}, \frac{-5}{2}\rangle + |\frac{17}{2}, \frac{7}{2}\rangle \\
|1_L\rangle &= \frac{1}{\sqrt{3}}(|\frac{17}{2}, -\frac{11}{2}\rangle + |\frac{17}{2}, \frac{1}{2}\rangle + |\frac{17}{2}, \frac{13}{2}\rangle
\end{aligned}
\tag{3.15}
$$

There are two possible outcomes: if the result confirms the hypothesis, then you've made a measurement. If the result is contrary to the hypothesis, then you've made a discovery.

Enrico Fermi

4

Error Models

In our work we focus on the simplest of error models, due to both computational and time constraints, despite this we will discuss the variety of error models that could be applied in order to simulate a variety of different quantum systems, and to more deeply analyse the usefulness of different quantum codes.

4.1 Qubit Representation Noise

The most naive of the quantum error noise models that are used is local noise in the qubit representation. While this allows us to tune our simulation of error to exactly what it is in a physical system it also imposes a great disadvantage as the model has a dimension scaling of $d = 2^N$. As we discuss in section 5.2 this increase in dimension greatly increases computational time resulting in a processing time of order $O(2^{6.4N}) \approx O(85^N)$ which at $N = 15$ would have order 10^{21} times longer processing time than other suggested model. This representation, while useful for smaller systems or less complex processes, is therefore

unusable for simulations in this work.

4.2 Global Symmetric Noise

The most compact error model in our system, the Global symmetric noise model, takes the errors in the system to be applied to the whole system simultaneously and identically, ensuring the system stays in a permutation symmetric state. This is an important simplification as the permutation symmetry of our system is the key to keeping the dimension of our problem as low as possible. Physically this model corresponds to environmental modes that couples to symmetric spin operators, and hence is relevant for spins closely spaced relative to the wavelength of the modes.

The evolution of the system under global symmetric noise is given as an equally weighted rotation under the global spin operators $\hat{J}_x, \hat{J}_y$ and $\hat{J}_z$ with a depolarizing rate γ, with the Lindblad master equation given as [25]:

$$d\rho = \gamma dt \sum_{i\in\{x,y,z\}} \left(\hat{J}_i \rho \hat{J}_i - \frac{1}{2}\hat{J}_i^2 \rho - \frac{1}{2}\rho \hat{J}_i^2 \right) \tag{4.1}$$

This can be then decomposed into the Krauss operators given by:

$$\begin{aligned} E_0 &= \left(\sqrt{1 - J\left(J+1\right)\gamma dt} \right) \mathcal{I} \\ E_i &= \sqrt{\gamma dt}\hat{J}_i \quad \text{for } i \in \{x, y, z\} \end{aligned} \tag{4.2}$$

This representation is the smallest possible reduction in dimension for our simulations, hence is highly valuable for considering larger systems and complex optimisation processes.

4.3 Local Symmetric Noise

In some situations more realistic variant of the global symmetric noise model is the local symmetric noise model [5], which still assumes that the system noise acts identically on each qubit, but acts locally on the qubit itself rather than on the entire system. This forces us to use a different model for the symmetric system we are looking at, using the block diagonal representation to model not just the highest angular momentum systems, but also all the lower dimension systems. This model hence takes a larger size matrix representation, with

the dimension of a N-qubit system being given by the sum of the dimensions of all the smaller symmetric states, e.g. a system with N=5 would look like:

$$\begin{bmatrix} \left[\rho_{J=\frac{5}{2}}\right] & \cdot & \cdot \\ \cdot & \left[\rho_{J=\frac{3}{2}}\right] & \cdot \\ \cdot & \cdot & \left[\rho_{J=\frac{1}{2}}\right] \end{bmatrix} \tag{4.3}$$

Where the dimension of the block matrices is $2J + 1$. This representation, referred to as the "irrep basis" is a representation of all the possible variations of J and M, in the case of local symmetric noise, states that start in the symmetric space evolve into this block diagonal form. The efficiency of this representation is in the degenerate blocks, identical copies of the $[\rho_J]$ blocks, not being distinguished, due to the symmetrical action of our system. In a more complete form, we add an additional label λ to our representation such that our state is given as $|J, \lambda, M\rangle$. The number of degenerate copies of a given block is given as [12]:

$$C_J^N = \frac{2J+1}{\frac{N}{2}+J+1}\binom{N}{\frac{N}{2}-J} \tag{4.4}$$

For example in the case of $N = 5$ given in Eq.4.3:

$$\begin{aligned} C_{5/2}^5 &= 1 \\ C_{3/2}^5 &= 4 \\ C_{1/2}^5 &= 5 \end{aligned} \tag{4.5}$$

As the blocks are identical, we can display only one of each, reducing the representation in size from the qubit basis. This model also uses a far more complex evolution, derived in Ref [5] this evolution takes the form:

$$\dot{\rho} = \mathcal{L}(\rho) = \sum_{J,M,M'} \rho_{J,M;J,M'} f^J_{M,M'} \tag{4.6}$$

Where $\rho_{J,M;J,M'}$ is the term from ρ in the J^{th} block with position in the block $|M\rangle\langle M'|$ and the evolved state $f^J_{M,M'}$ given by:

$$
\begin{aligned}
f = &\sum_{q,r\in\{+,-,z\}} g_{qr} \text{ for} \\
g_{qr} = &\frac{A_q^{J,M} A_r^{J,M'}}{2J}\left(1 + \frac{\alpha_N^{J+1}(2J+1)}{d_N^J(J+1)}\right)|J, M+q\rangle\langle J, M'+r| \\
&+ \frac{B_q^{J,M} B_r^{J,M'} \alpha_N^J}{2J d_N^J}|J-1, M+q\rangle\langle J-1, M'+r| \\
&+ \frac{D_q^{J,M} D_r^{J,M'} \alpha_N^{J+1}}{2(J+1) d_N^J}|J+1, M+q\rangle\langle J+1, M'+r|
\end{aligned}
\tag{4.7}
$$

With the terms:

$$
\begin{aligned}
d_N^J &= \frac{N!(2J+1)}{(N/2-J)!(N/2+J+1)!} \\
\alpha_N^J &= \sum_{J'=J}^{\frac{N}{2}} d_N^{J'} \\
A_\pm^{J,M} &= \sqrt{(J\mp M)(J\pm M+1)} \\
A_z^{J,M} &= M \\
B_\pm^{J,M} &= \pm\sqrt{(J\mp M)(J\mp M-1)} \\
B_z^{J,M} &= \sqrt{(J+M)(J-M)} \\
D_\pm^{J,M} &= \mp\sqrt{(J\pm M+1)(J\pm M+2)} \\
D_z^{J,M} &= \sqrt{(J+M+1)(J-M+1)}
\end{aligned}
\tag{4.8}
$$

This results in a far more complex process of applying the error map and a larger representation with total dimension $d = \frac{N(N+1)}{2}$, or a dimensional scaling $d \propto N^2$, which grows far more rapidly than the previous representation. This scaling combined with the more complex evolution will push our previously discussed computational limits, with the minimum additional quadratic scaling making analysis of our commonly discussed systems too large. Despite this it is possible to consider only the single symmetric representation in the model, an consider all movement outside this representation as loss, as in our system we only measure the symmetric system, hence these would be lost. This does though restrict the recovery operation from recovering any part of the state that leaves the symmetric state and will as a result give a poorer recovery operation than may have been possible with the full representation.

The Kraus operator representation for this model can be derived as the possible actions on a state, e.g. the ket side for each possible g operator, such that:

$$E(\gamma) = \sum_{M} \sum_{i \in q} \frac{A_i^{J,M}}{\sqrt{2J}} \sqrt{\gamma \left(1 + \frac{\alpha_N^{J+1}(2J+1)}{d_N^J(J+1)}\right)} |J, M+q\rangle\langle J, M| \tag{4.9}$$

Where the term γ is the error rate. Using all the terms compressed into a single Kraus operator allows for all the combinations of q and r from Eq.4.7 as well as all the different energy level mappings M, M' to be included. Additionally an identity term is included for the case where no error occurs, this term is given weighting $1 - \sqrt{dim(q) * \gamma}$ where $dim(q)$ is the number of the different error types being used. This system is using an even weighting of all the different error types for the sake of general application, but this can be tailored to any ratio to better represent specific systems.

The important thing in science is not so much to obtain new facts as to discover new ways of thinking about them.

William Lawrence Bragg

5 Quantum Error Recovery

5.1 Optimal Error Recovery

Unlike a classical system, which is often represented by a vector in space or a scalar value, it is not particularly straight forward to determine the distance between or difference in two quantum states. A simplistic measure of the closeness of states is the fidelity measure given by:

$$F(\rho,\sigma) = \left(tr\sqrt{(\sqrt{\rho}\sigma\sqrt{\rho})}\right)^2 \tag{5.1}$$

Which gives the probability that one state will, by measurement, appear to be the other state. This is useful in simplistically comparing states in error correction as it gives a probability that the state will successfully be recognised by the system after introducing some error. Further as most error correction codes are pure states the calculation can be simplified to:

$$F(\rho,\sigma) = \langle\rho|\,\sigma\,|\rho\rangle \tag{5.2}$$

Where the pure state (ρ) is compared to the (possibly) impure state (σ) that has had some

error introduced. We begin expanding this definition by considering the effect of an error channel $\mathcal{E}$ on a system ρ as:

$$F(\rho, \mathcal{E}) = F(\rho, \mathcal{E}(\rho)) = \langle\rho| \mathcal{E}(|\rho\rangle\langle\rho|) |\rho\rangle \tag{5.3}$$

In our work to focus of the goal of optimisation of state recovery we use study the entanglement fidelity of a system [33], effectively considering the state as a part of a larger system and taking the fidelity of the larger system. This consideration is important as some errors on a system will not show losses in entanglement within a system. As an example we can consider the state $\rho(0) = \frac{I}{2}$ under a total depolarizing channel $\mathcal{E} \equiv E_0 = \sqrt{1 - \frac{3\gamma}{4}}, E_j = \sqrt{\frac{\gamma}{4}}\sigma_j$ for $j \in \{x, y, z\}$. Under this evolution the trace of this system is clearly 1 has the initial and final states are identical, but in consideration of a larger system $|\psi\rangle_{RQ} = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$, from which we can see that $\rho_Q = \mathrm{Tr}_R[|\psi\rangle\langle\psi|]$ is a subsystem of, we see that the fidelity gives:

$$\begin{aligned} F_e(\rho, \mathcal{E}) =& \sum_i \langle\psi| (I \otimes E_i)(|\psi\rangle\langle\psi|)(I \otimes E_i)^\dagger |\psi\rangle \\ =& 1 - \frac{3\gamma}{4} \langle\psi| (|\psi\rangle\langle\psi|) |\psi\rangle + \frac{\gamma}{4} \langle\psi| (|01\rangle\langle 01| + |01\rangle\langle 01|) |\psi\rangle \\ &+ \frac{\gamma}{4} \langle\psi| (|01\rangle\langle 01| + |01\rangle\langle 01|) |\psi\rangle + \frac{\gamma}{4} \langle\psi| (|\psi\rangle\langle\psi|) |\psi\rangle \\ =& 1 - \frac{\gamma}{2} \end{aligned} \tag{5.4}$$

It can be seen that a channel can effect an entangled state on a larger space even if the reduced state is unchanged. Hence we consider entanglement fidelity by considering our system Q as a part of the joint system RQ. In this system we denote the larger operator ψ^{RQ} such that $tr_R(\psi^{RQ}) = \rho^Q$ We consider the evolution of our system given by the superoperator $\mathcal{E}^Q$ such that our system undergoes an overall evolution given by $(I^R \otimes \mathcal{E}^Q)$, hence our final state is given by:

$$\mathcal{E}(\psi^{RQ}) = (I^R \otimes \mathcal{E}^Q)(|\psi^{RQ}\rangle \langle\psi^{QR}|) \tag{5.5}$$

As the initial state is pure the fidelity is given by:

$$F_e(\rho, \mathcal{E}) = \langle\psi^{RQ}| \mathcal{E}(\psi^{RQ}) |\psi^{RQ}\rangle \tag{5.6}$$

This measure of fidelity is the entanglement fidelity, which has some very nice properties for manipulation and optimisation. First we can see that the superoperator can be written in

the Kraus operator form as:

$$(I \otimes \mathcal{E})(|\psi^{RQ}\rangle \langle\psi^{QR}|) = \sum_{\mu}(I^R \otimes E_\mu^Q)(|\psi^{RQ}\rangle \langle\psi^{QR}|)(I^R \otimes E_\mu^Q)^\dagger \tag{5.7}$$

Hence the entanglement fidelity will be given by:

$$F_e(\rho, \mathcal{E}) = \langle\psi^{RQ}| (I \otimes \mathcal{E})(|\psi^{RQ}\rangle \langle\psi^{RQ}|) |\psi^{RQ}\rangle \tag{5.8}$$

By taking this in the form of Eq5.2 we take:

$$F_e(\rho, \mathcal{E}) = \sum_{\mu} \langle\psi^{RQ}| (I^R \otimes E_\mu^Q)(|\psi^{RQ}\rangle \langle\psi^{RQ}|)(I^R \otimes E_\mu^Q)^\dagger| |\psi^{RQ}\rangle \tag{5.9}$$

Which can be separated to:

$$\begin{aligned} F_e(\rho, \mathcal{E}) &= \sum_{\mu}(\langle\psi^{RQ}| (I^R \otimes E_\mu^Q) |\psi^{RQ}\rangle)(\langle\psi^{RQ}| (I^R \otimes E_\mu^Q)^\dagger |\psi^{RQ}\rangle) \\ &= \sum_{\mu} \left|(\langle\psi^{RQ}| (I^R \otimes E_\mu^Q) |\psi^{RQ}\rangle)\right|^2 \\ &= \sum_{\mu} \left|\mathrm{Tr}\left[(I^R \otimes E_\mu^Q)|\psi^{RQ}\rangle\langle\psi^{RQ}|\right]\right|^2 \end{aligned} \tag{5.10}$$

As the partial trace over the system R is ρ by requirement we can simplify this expression as:

$$\begin{aligned} F_e(\rho, \mathcal{E}) &= \sum_{\mu} \left|\mathrm{Tr}\left[\mathrm{Tr}_R\left((I^R \otimes E_\mu^Q)|\psi^{RQ}\rangle\langle\psi^{RQ}|\right)\right]\right|^2 \\ &= \left|\sum_{\mu} \mathrm{Tr}\left[E_\mu^Q \mathrm{Tr}_R\left(|\psi^{RQ}\rangle\langle\psi^{RQ}|\right)\right]\right|^2 \\ &= \sum_{\mu} |tr(\rho E_\mu)|^2 \end{aligned} \tag{5.11}$$

For the elements of the Krauss Superoperator $\mathcal{E}$ given by A_i acting on the state ρ of the system $\mathcal{H}_1$. It is then our goal to implement a recovery operator $\mathcal{R}$ that maximises the entanglement fidelity $F_e(\rho, \mathcal{R} \circ \mathcal{E})$.

To perform this optimisation we now introduce the concept of vectorisation to our definitions. The vectorisation for an operator $A = \sum_{i,j} a_{ij} |i\rangle \langle j|$ is:

$$|A\rangle\rangle = \sum_{i,j} a_{ij} |i\rangle_1 |j\rangle_2 \tag{5.12}$$

These vectorized states have a number of useful properties that can be manipulated for the use of this proof. Firstly the order of operations within a vectorized state relates to the system on which the operator was acting as so:

$$A \otimes B|C\rangle\rangle = |ACB^T\rangle\rangle \tag{5.13}$$

Where the dimension of the components allows. Additionally we can show that the partial trace of a vectorized matrix A & B can be given by:

$$AB^\dagger = tr_2(|A\rangle\rangle\langle\langle B|) \tag{5.14}$$

Proof:

$$LetA = \sum_{i,j} a_{ij} |i\rangle \langle j| and B = \sum_{k,l} b_{kl} |k\rangle \langle l| \tag{5.15}$$

$$AB^\dagger = \sum_{i,j,k,l} a_{ij} b^*_{kl} |i\rangle \langle j| |l\rangle \langle k| \tag{5.16}$$

To find non-zero entries we take j=l giving:

$$AB^\dagger = \sum_{i,j,l} a_{ij} b^*_{kj} |i\rangle \langle k| \tag{5.17}$$

And for the right-hand side we take:

$$|A\rangle\rangle = \sum_{i,j} a_{ij} |i\rangle_1 |j\rangle_2 and |B\rangle\rangle = \sum_{k,l} b_{kl} |k\rangle_1 |l\rangle_2 \tag{5.18}$$

Hence:

$$\langle\langle B| = \sum_{k,l} b^*_{kl} \langle l|_2 \langle k|_1 \tag{5.19}$$

And

$$|A\rangle\rangle\langle\langle B| = \sum_{i,j,k,l} a_{ij} b^*_{kl} |i\rangle_1 |j\rangle_2 \langle l|_2 \langle k|_1 \tag{5.20}$$

And hence the partial trace is given by summing over the second system, i.e:

$$tr_2(|A\rangle\rangle\langle\langle B|) = \sum_{i,j,k,l} \langle j|_2 a_{ij} b^*_{kl} |i\rangle_1 |j\rangle_2 \langle l|_2 \langle k|_1 |j\rangle_2 \tag{5.21}$$

Which gives non-zero values for j=l and hence can be simplified to:

$$tr_2(|A\rangle\rangle\langle\langle B|) = \sum_{i,k,j} a_{ij} b^*_{kj} |i\rangle_1 \langle k|_1 = \sum_{i,k,j} a_{ij} b^*_{kj} |i\rangle \langle k| = AB^\dagger \tag{5.22}$$

These relationships allow us to introduce the use of vectorized notation into our equations for entanglement fidelity using Eq.5.11 as by the cyclic property of the trace and the Hermitian property of density matrices we take:

$$tr(\rho B_i) = tr(B_i \rho^\dagger) \tag{5.23}$$

And by Eq.(5.14) we take:

$$tr(\rho B_i) = tr(tr_2(|B_i\rangle\rangle\langle\langle\rho|) \tag{5.24}$$

And as the trace of the partial trace is the trace of the larger system:

$$tr(\rho B_i) = tr(|B_i\rangle\rangle\langle\langle\rho|) = \langle\langle\rho|B_i\rangle\rangle \tag{5.25}$$

Hence we can take the entanglement fidelity we found earlier to be given by:

$$F_e(\rho, B_i) = \langle\langle\rho|B_i\rangle\rangle\langle\langle B_i|\rho\rangle\rangle \tag{5.26}$$

We then can take the central term $||B_i\rangle\rangle\langle\langle B_i|\,|$ to form an operator $X_{\mathcal{B}_i}$. Hence by summing over we get the form:

$$F_e(\rho, B_i) = \langle\langle\rho|X_B|\rho\rangle\rangle \tag{5.27}$$

Which takes a form for which we can apply a semidefinite program to optimise over. To do this we consider the error channel $\mathcal{E}$ and apply after it a recovery channel $\mathcal{R}$ giving us the composite channel $\mathcal{E} \circ \mathcal{R} : \mathcal{L}(\mathcal{H}_1) \to \mathcal{L}(\mathcal{H}_1)$. We take our central term now as the operator X_{RE} such that:

$$X_{\mathcal{RE}} = \sum_{i,j} |R_i E_j\rangle\rangle\langle\langle R_i E_j| \tag{5.28}$$

By using equation (5.13) we can then separate out the error operations as:

$$X_{\mathcal{RE}} = \sum_{i,j} I \otimes E_j^T |R_i\rangle\rangle\langle\langle R_i| I \otimes E_j^* \tag{5.29}$$

$$= \sum_j (I \otimes E_j^T) X_{\mathcal{R}} (I \otimes E_j^*) \tag{5.30}$$

Reintroducing this into the equation for fidelity gives us the form:

$$F(\rho, \mathcal{R} \circ \mathcal{E}) = \sum_j \langle\langle\rho|(I \otimes E_j^T) X_{\mathcal{R}} (I \otimes E_j^*)|\rho\rangle\rangle \tag{5.31}$$

$$= \sum_j \langle\langle\rho E_j^\dagger | X_{\mathcal{R}} | \rho E_j^\dagger\rangle\rangle \tag{5.32}$$

Thus we see this takes the form of a pure state and an operator hence we rewrite this as a trace problem:

$$C_{\rho,\mathcal{E}} = \sum_j |\rho E_j^\dagger\rangle\rangle\langle\langle \rho E_j^\dagger| \tag{5.33}$$

$$F(\rho, \mathcal{R} \circ \mathcal{E}) = tr(X_{\mathcal{R}} C_{\rho,\mathcal{E}}) \tag{5.34}$$

Hence we get our system in the form of an optimisation problem, and by using the property proven in Fletcher, Shor, and Win that $tr_{\mathcal{H}_1} X = I$ we can create a semidefnite program to solve for $X_{\mathcal{R}}$, such that:

$$\begin{gathered} X_{\mathcal{R}}^* : max(tr(X_{\mathcal{R}} C_{\rho,\mathcal{E}})) \\ \text{such that } X_{\mathcal{R}} \geq 0 \text{ and } tr_{\mathcal{H}_2}(X_{\mathcal{R}}) = I_{\mathcal{H}_1} \end{gathered} \tag{5.35}$$

For $X_{\mathcal{R}}$ & $C_{\rho,\mathcal{E}}$ being $(N+1)^2 \times (N+1)^2$ matrix representations of the vectorised system.

5.2 Coding Optimal Recovery

In order to perform our semidefinite program (SDP) optimisations we use the Python package CVXPY [19] to implement our solver. For our solver we have taken the freely available SCS (Splitting Conic Solver) solver, used by default in CVXPY, to solve this form to first order.

The SCS solver implements an operator splitting method called "alternating direction method of multipliers" and some assumptions about the structure of a conic to be optimised. This differs from the standard method for optimising a semidefinite problem using the interior points method. The resulting change allows the SCS solver to solve a SDP faster, but at the cost of accuracy.

The SCS solver has been chosen over the alternative CVXOPT solver package for the purpose of processing efficiency. As we discuss in section (optimal recovery limit) there is a known limit to the optimality of recovery, and hence we optimise for the purpose of calculating example solutions. Using this limit we can observe the closeness of the first order approximation generated by the SCS solver and compare it to the best possible solution before accepting it. The computational complexity for the SCS solver, as compared to a second order solver, such as the MATLAB solver SDPT3 [48], has a speedup shown in Ref [34]

table 3, along with the changes to the ability of the solver. As our problem uses a matrix of dimension n^2 due to the vectorisation in our rearrangement our SDP problem dimension can easily exceed 100, for example the 17-qubit gross code will have dimension 324, and appears to have fewer constraints than the case given. From this we can assume that the problem will unlikely be easily solvable on any reasonably sized computer.

Further Ref [10] page 618 discusses the computational complexity of applying the interior points method of analysis, which scales as the max of the terms $\{np^3, n^2p^2, n^3\}$ where the object being maximised is $x \in R^n$ and p is given by the complexity of the constraints. As the constraints are trivially small compared to the dimension of n we look at the n^3 term and notice the definition of n is given as the total size of the variable x. As our vectorised operators are $N^2 \times 1$ and we are optimising $|R\rangle\rangle\langle\langle R|$ the operator $X_{\mathcal{R}}$ is of size $N^2 \times N^2$ where N is the number of qubits in our system we take $n = N^4$ hence we get, at worst, scaling of order $O(N^{12})$. By analysing the data, and specifically the factorisation time, which is the bulk of the computation time, (section 3.2.3), we instead see $n^{1.55}$ scaling, calculated empirically. This scaling is nearly quadratically faster than the interior points method, with qubit dimension scaling $O(N^{6.2})$.

Taking this and our known code size of 15-17 qubits we can assume that the change in efficiency is extremely large, with an estimated increase in efficiency of around order 10^8. An example of this is shown in Ref [22] Fig 6 where they show the run time for simulations with matrices of increasing dimension n, as the MOSEK [1] second order solver scales rapidly to a point where it is not comparable, or realistically useable in the same time frame as SCS or the Julia package COSMO.

Estimating the time scaling for our data with MOSEK we can see in Ref [22] that a matrix of dimension of 200 (slightly smaller than ours) has a solve time of around 1000 times slower than SCS. With our simulations taking around 2 minutes per timestep this would result in our 10 time step simulations to produce data going from an average run time of 20 minutes to almost two weeks. Assuming the accuracy loss is not highly impactful, as our goal is not a complete working system but a feasibility test, this improvement in the availability of simulations is too important to ignore. Additionally we come into the issue of non-commercial solvers, especially the CVXOPT solver available through the CVXPY package, is unable to handle a problem on the scale of our optimisations. Further the requirements of our processor increase with the order of the analysis, as well as the size of the problem,

as mentioned in O'Donoghue et al. the SCS solver was able to complete some problems that were not possible with the second order solvers, purely due to the amount of memory required. This is an additional limitation to our simulations, as the memory available to us, without the use of a commercial web server, is capped at around 25-30 Gb and many of the simulations were pushing upwards of 22GB in the SCS solver, rendering them likely unable to be completed with to a second order.

5.3 Optimal Recovery Limit

As our solver for SDP resolves the problem to only first order resolution, we will need some form of measure to test the effectiveness of this solver and the applicability of the solutions it produces. To do this we will use a method discussed in Ref [20] to find the maximum optimal recovery that can be achieved and analyse our closeness to this value. We consider eq 5.33 and introduce the Lagrange multiplier $\Lambda = \lambda \otimes 1_H$, using the constraint from [21] that $tr_2 X = I$, this gives the functional:

$$\hat{F}(X) = tr(XC) - tr(X\Lambda) \tag{5.36}$$

By expanding X into its eigenstates we get:

$$\hat{F}(X) = \sum_i x_i \langle x_i | (C - \Lambda) | x_i \rangle \tag{5.37}$$

And by taking the extremal of the Lagrange multiplier we get:

$$(C - \Lambda) x_i | x_i \rangle = 0 \tag{5.38}$$

and by summing over all these terms we get:

$$(C - \Lambda) X = 0 \tag{5.39}$$

with some rearranging we get:

$$X = \Lambda^{-1} C X \tag{5.40}$$

and by taking the Hermitian conjugate and using the knowing its equivalence we can take that:

$$X = \Lambda^{-1} C X C \Lambda^{-1} \tag{5.41}$$

Hence by reintroducing the constraint on the partial trace we can see that:

$$\lambda^2 = tr_2(CXC) \tag{5.42}$$

And by taking λ to be a positive Hermitian operator we can take the square root. By also taking the eigen-decomposition of the operators C we can see that:

$$\lambda = \left(\sum_{i,j} c_i c_j tr_2(|c_i\rangle\langle c_i|X|c_j\rangle\langle c_j|) \right)^{\frac{1}{2}} \tag{5.43}$$

And as in the method in Ref [18] we take the maximum eigenvalues of C, given as c_{max} to limit the function, as this will be the maximum value that λ can take:

$$\begin{aligned} \lambda &\leq c_{max} \left(\sum_{i,j} tr_2(|c_i\rangle\langle c_i|X|c_j\rangle\langle c_j|) \right)^{\frac{1}{2}} \\ \lambda &\leq c_{max} \left(tr_2(IXI) \right)^{\frac{1}{2}} \end{aligned} \tag{5.44}$$

Hence we find a limit for λ and hence the extremal solution for the fidelity:

$$\begin{aligned} \lambda &\leq c_{max} I \\ F &= tr(\Lambda X) \\ F &= tr(tr_2(\Lambda X)) \\ F &= tr(\lambda I_1) \\ F &\leq c_{max} tr(I_1) \\ F &\leq c_{max} dim(I_1) \end{aligned} \tag{5.45}$$

Hence we have a upper bound for our optimised fidelity, and if we reach this limit our fidelity is optimal [20]. Further we can use this limit to determine how close our first order approximation is to optimality.

5.4 Near Optimal Recovery

As we have shown optimal recovery suffers from extremely high computational complexity scaling for actual implementation we will have limitations on our ability to perform these

calculations for a realistically sized system. With this in mind we consider the study of near optimal recovery [6], which forgoes some level of accuracy for a decrease in computation requirement. They give the near optimal recovery $\mathcal{R}$, given an error map $\mathcal{E}$ with Kraus operators $\{E_i\}$ as:

$$R_{\mathcal{E},\rho,i} = \rho^{\frac{1}{2}} E_i^{\dagger} \mathcal{E}(\rho)^{-\frac{1}{2}} \tag{5.46}$$

This recovery operation is shown to be near optimal, with a recovery given by $F_e(\rho, \mathcal{R} \circ \mathcal{E}) \geq F_e(\rho, \mathcal{R}_{\mathcal{E},\rho} \circ \mathcal{E}) \geq (F_e(\rho, \mathcal{R} \circ \mathcal{E}))^2$ for the optimal recovery given by $F_e(\rho, \mathcal{R} \circ \mathcal{E})$. Such that by taking our recovered fidelity to be given as $1 - \nu$ we get the limits:

$$1 - \nu \geq F_e(\rho, \mathcal{R}_{\mathcal{E},\rho} \circ \mathcal{E}) \geq (1 - \nu)^2 \geq 1 - 2\nu \tag{5.47}$$

Giving that our recovery is at most a factor 2 worse than the optimal recovery method. Especially for values close to 1 this may by insignificant enough that it makes this system viable. This limited recovery is useful for lower error ranges, restricting the systems that can be corrected with this method, but the trade-off is an enormously faster computational procedure, with the main computational complexity arising from the generalised inverse with complexity $O(n^3)$ [49] for an nxn matrix with the Moore-Penrose inverse using standard methods.

[Science is] a great game. It is inspiring and refreshing. The playing field is the universe itself.

Isidor Isaac Rabi

6

Error Recovery Results

An additional obstacle we encountered in the application of our SDP optimisation procedure is that the available solvers in both Python and Mathematica solve only real valued problems [19][48]. While the codes we are looking at are all real valued codes there is no immediate evidence that the optimisations of our code should be purely real valued. To overcome this we used a lifting method ([10] exercise 4.42 and solution) to convert a complex operator to a real operator using the equivalency:

$$X \geq 0 \leftrightarrow \begin{pmatrix} \Re X & -\Im X \\ \Im X & \Re X \end{pmatrix} \geq 0 \tag{6.1}$$

As this operation is necessary in the case that our optimal recovery operator is complex, which is not an obvious requirement of our system, we compared the performance of the codes using both the real only and complex valued operators. This showed identical behaviour between the two. The performance of our codes will be considered using the three measures of code recovery earlier mentioned, our first order SDP, the near optimal recovery operation and finally the optimal recovery limit as the measure of performance for the two other methods.

For the sake of clearly showing differences in performance, the plots used in this section showing the change in entanglement fidelity over time will use a log-log scale and will show $1 - F_e$ for the entanglement infidelity. such that a value of 0 indicates perfect recovery.

6.1 Global Symmetric Noise Analysis

6.1.1 Gross Spin Code Results

For our simulations of the Gross code we focus of the 17 qubit code, as this is claimed to have the best performance of any of the tested states. To first ensure the maximally efficient version of the code was used we plotted the limiting case fidelity of the code against the variable ϕ under the Global Symmetric Noise model (6.1). This yields a range of possible choices of ϕ, and we we find that $\phi = \pi$ has the best recovery.

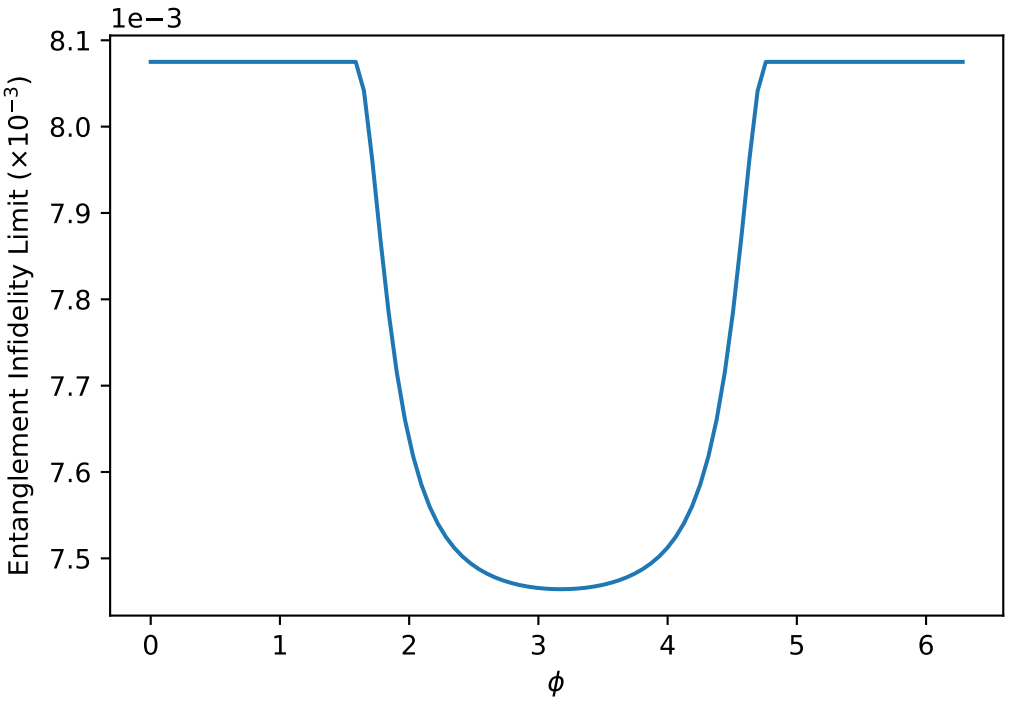


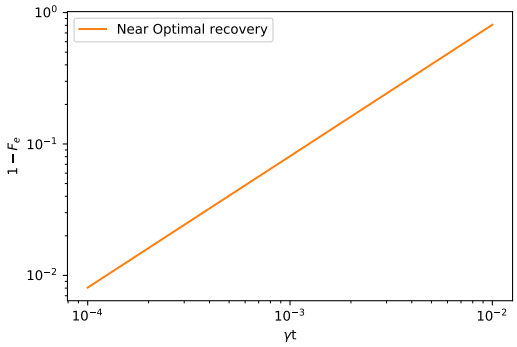
Figure 6.1: Variation of the limit of Recovered Entanglement Infidelity of the 17-qubit Gross code as a function of the variable ϕ for the optimisation of the ρ_4 and ρ_5 irreps given as $|\phi\rangle = a\,|0_{\rho_4}\rangle + \sqrt{1-a^2}e^{i\phi}\,|0_{\rho_5}\rangle$

We first performed our three performance test methods on the Gross code with the Global Symmetric Noise. Figure 6.2 show the results of the simulations to yield almost

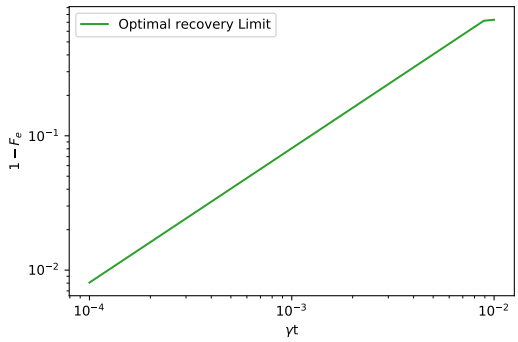
identical, the exact values can be seen in the GitHub repository in the appendix (A). This result seemed contrary to our expected performance, as both the Barnum-Knill near Optimal Recovery and the Optimal Error recovery methods were assumed to have been lower accuracy representations of the best possible recovery. To inspect the recovery operator for the system we took the eigenvectors of the output of our SDP, as it takes the form:

$$X_{\mathcal{R}} = \sum_i |\mathcal{R}_i\rangle\rangle\langle\langle\mathcal{R}_i| \tag{6.2}$$

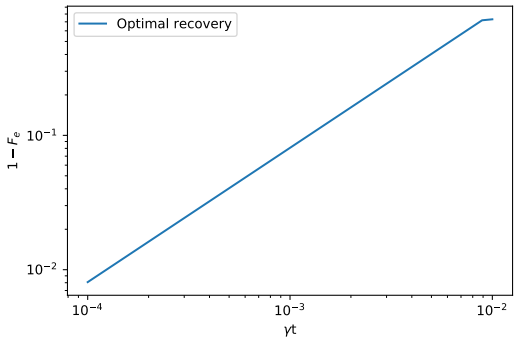
Interestingly this resulted in only one non-zero eigenvalue, excluding values below 10^{-8}. This implied that the optimal recovery operator for the gross code under Global noise is a projector, the same way that the Barnum-Knill recovery effectively projects an "undoing" of the error onto the system, except that this process can be done with a projector rather than a number of operations.



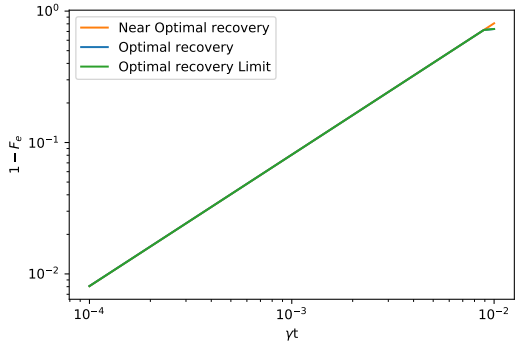
(a) Barnum-Knill near optimal recovery



(b) Limit of possible optimal recovery



(c) First order solution for SDP optimal recovery



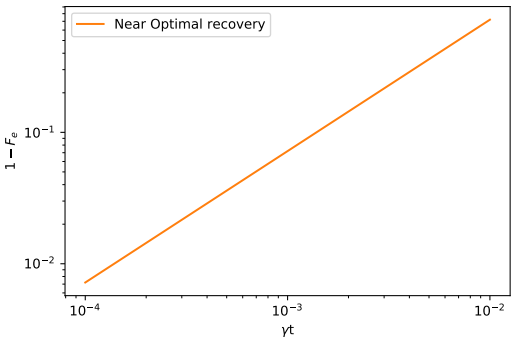
(d) Combined plot of the three results

Figure 6.2: Comparison of the three different error recovery methods on the 17- qubit Gross code under global symmetric noise.

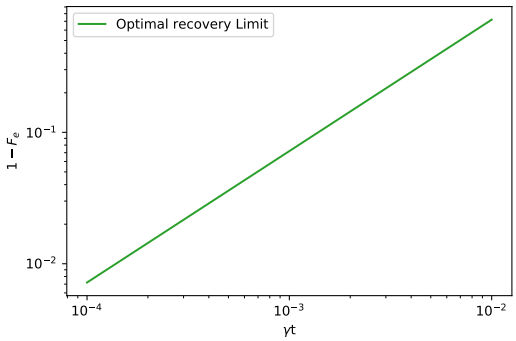
6.2 GNU Code Results

Similarly to the 17-qubit Gross code we studied the (4,4,1) GNU code under the effects of global symmetric noise. This study differs from the expected uses for the GNU code as it known to correct a set number of errors, alternatively this model investigates the case of probabilistic errors occurring, such that many errors can occur, with a very low probability.

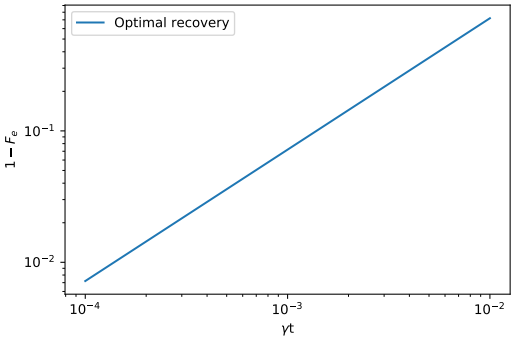
We see from our results (fig6.3) again a functionally identical performance from the three different recovery methods. Again by taking the eigenvalues of the recovery operator X we can see that there is only a single non-zero eigenvalue, hence the optimised recovery operation is a projector.



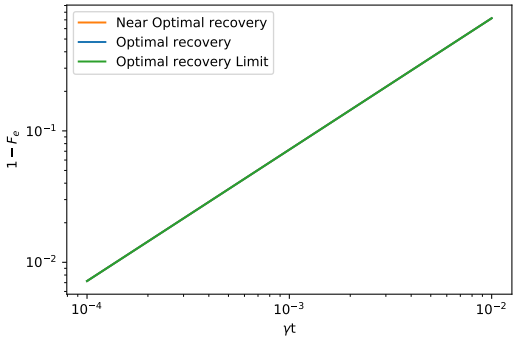
(a) Barnum-Knill near optimal recovery



(b) Limit of possible optimal recovery



(c) First order solution for SDP optimal recovery



(d) Combined plot of the three results

Figure 6.3: Comparison of the three different error recovery methods on the (4,4,1) gnu code under global symmetric noise.

6.2.1 Qudit GKP Code and Minimal Qudit code Data

We see much the same behaviour from the Qudit GKP code as in the case for the Gross and GNU codes, where all three recovery methods showed near identical results. On the other hand our data for Minimal Qudit codes 6.5 shows a marked difference from the gnu and Gross codes, as the worst performance is the optimal recovery SDP. This was not a highly unexpected result, as we were expecting the optimisation to not meet the optimal limit with only a first order solution of the SDP, but the less expected result is the greater performance of the near optimal recovery. The near optimal recovery is limited to perform at best equal to the perfectly calculated optimal recovery, and up to quadratically worse [6], hence its consistency is highly encouraging to its potential uses.

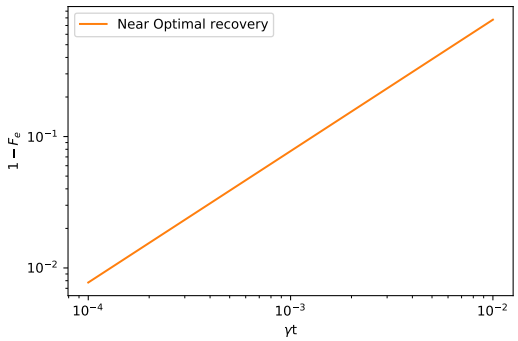
Interestingly the optimal recovery SDP again generated recovery operators for both the minimal qudit and GKP codes with a single non-zero eigenvalue. Hence implying that there exists no projector that can optimally correct the minimal qudit system.

6.2.2 Global Symmetric Noise Comparisons

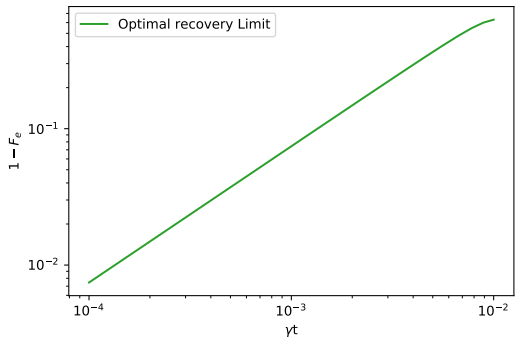
To draw conclusions on the performance of the different codes under these different schemes we compared the different results for each recovery method, comparing their performances to gauge their usefulness. Granted this is not a conclusive result as the global symmetric model is not a perfect nor even representative of realistic quantum systems, and to compare actual code performance one would need a highly accurate study of the noise within the system.

We start by investigating the performance under the limiting case, as with sufficient resources the calculations for the semi-definite program optimisation could be performed with near perfect accuracy which would result in performance close to the optimal limit, hence this calculation is likely the closest to realistically optimal.

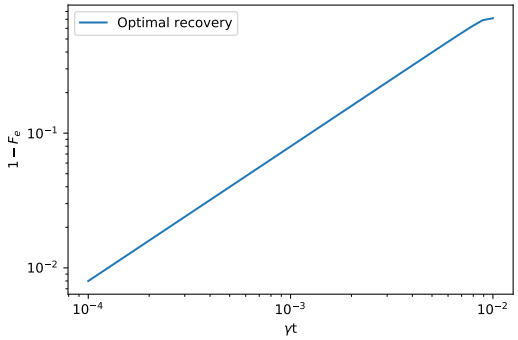
This result has been the core of much anguish in this thesis, as surprisingly it does not agree with the results presented by Gross [25] specifically with a major difference in the performance of the Gross code. In retrospect this may have arisen due to differences in the method of calculation for the entanglement fidelity, but the results for the minimal qudit is similar, with an almost indiscernible difference for the minimal qudit code. The Qudit GKP code has a sizeable difference of a factor ≈ 10 while the result for the Gross code is different by a factor of about 10^4. This difference in results was one of the motivating factors behind



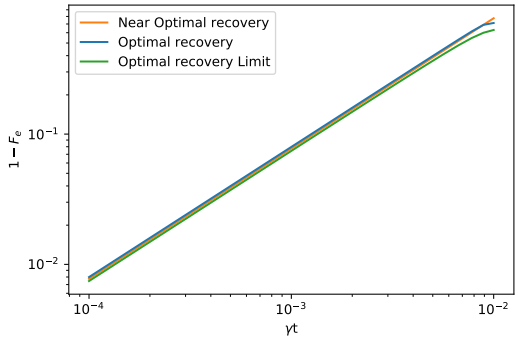
(a) Barnum-Knill near optimal recovery



(b) Limit of possible optimal recovery



(c) First order solution for SDP optimal recovery



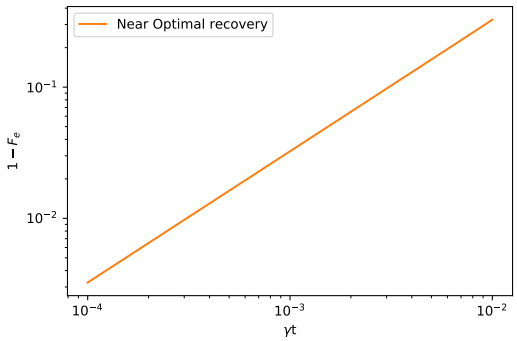
(d) Combined plot of the three results

Figure 6.4: Comparison of the three different error recovery methods on the 17- qubit GKP code under global symmetric noise.

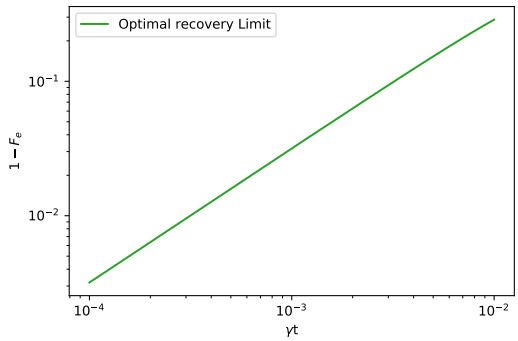
the multiple methods of fidelity analysis used and is still a discomforting difference.

Alternatively the difference may lie in the Gross code calculated, as the 17-qubit Gross code is not specifically given in Gross' paper. To ensure that this was not the case we calculated not only the 17-qubit code but the 7 and 9 qubit codes as well, as these were given in the paper. For these we derived the same codes as Gross. To further ensure that no miscalculations had occurred we computed the variation of the codes entanglement fidelity limit against the variables a, ϕ in eq.3.10. Fig.6.1, as discussed earlier found $\phi = \pi$ among other values to be optimal.

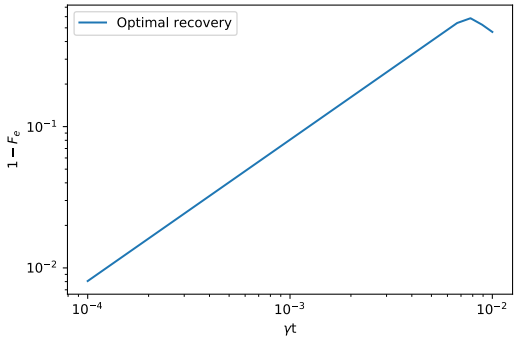
Despite the fact that a had a specified value in terms of ϕ we varied this value to ensure that this was correctly the most efficient value for a, as the condition that defined a was for use in error correction and Gross may have used an alternate value for a for error recovery. We found that a for the 17-qubit code should be given as:



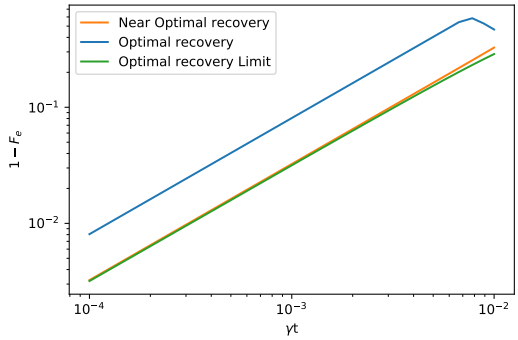
(a) Barnum-Knill near optimal recovery



(b) Limit of possible optimal recovery



(c) First order solution for SDP optimal recovery



(d) Combined plot of the three results

Figure 6.5: Comparison of the three different error recovery methods on the Minimal Qudit code with dimension 18 under global symmetric noise.

$$a = \frac{1}{2}\sqrt{\frac{-34\sqrt{910}\sqrt{7280\cos(2\phi)+8699}|\cos(\phi)| + 61880\cos(2\phi) + 72071}{61880\cos^2(\phi)+6241}} \tag{6.3}$$
$$\approx 0.115 \text{ for } \phi = \pi$$

As can be seen in Fig6.7 this value corresponds to a peak in the entanglement infidelity, this may result in a variation from the optimal code. The condition that a is given as Eq 6.3 is given by Gross as a necessity to maintain the error correction condition $\langle 0| J_z |0\rangle$, but may be varied in their simulations. Unfortunately there are few published simulations in the field and as a result comparison with the exact method of simulation used by Gross would be necessary to verify or invalidate either result.

Despite the differences in results the limit of recovered entanglement fidelity is highly useful an analysing code as it has been far more efficient at performing calculations, with

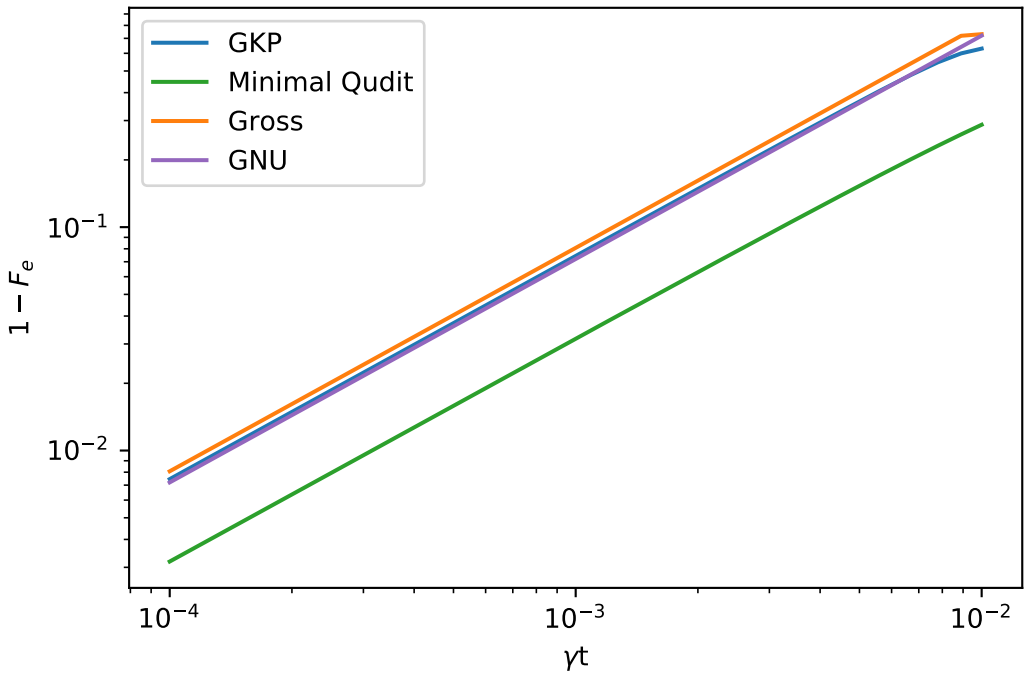


Figure 6.6: Comparison of the limit of recovered entanglement fidelity for the Qudit GKP, minimal qudit, GNU and Gross codes

thousands of time-steps only taking minutes, compared to the SDP optimisation, which would take 20-30 minutes to complete only 10 time-steps. Despite not being commonly talked about in many of the papers on entanglement fidelity this limit may prove to be a huge improvement for comparisons of code where the recovery operators are not needed to be known.

A further surprise in the performance of error recovery was the Barnum-Knill near-optimal recovery, which performed at or near the limit of recoverable entanglement fidelity in all cases. This encourages the idea that Barnum-Knill recovery may be an alternative to an SDP optimisation with a near equal performance, which could allow for far easier calculations of recovery operators and encourages the possibility of introducing a real time error detection and recovery system that could use ancillary qubit in a system to measure variations in the noise. Such a system would be far more applicable for sensing purposes as having to require a sensor to perform a large and complex calculation for its calibration would either require a large computational unit on board or access to a remote computational service, both of which would affect the viability of a mobile sensor. The Barnum-Knill recovery requiring only a fraction of the time and computational power of the SDP optimisation would be able

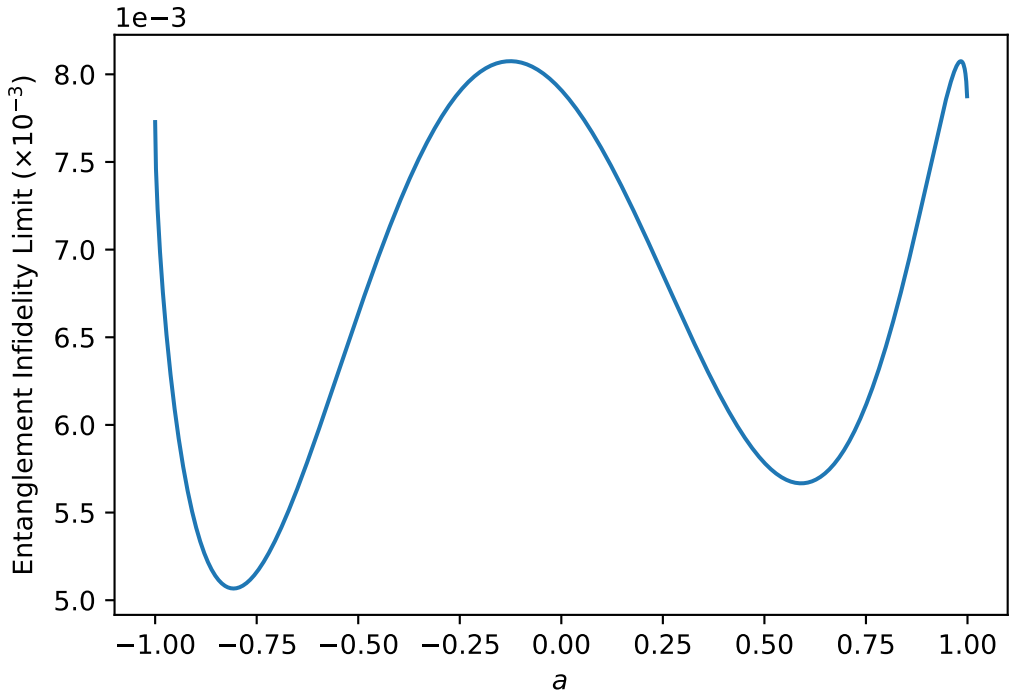


Figure 6.7: Variation of the limit of Recovered Entanglement Infidelity of the 17-qubit Gross code as a function of the variable a for the optimisation of the ρ_4 and ρ_5 irreps given as $|\phi\rangle = a\,|0_{\rho_4}\rangle + \sqrt{1-a^2}e^{i\phi}\,|0_{\rho_5}\rangle$

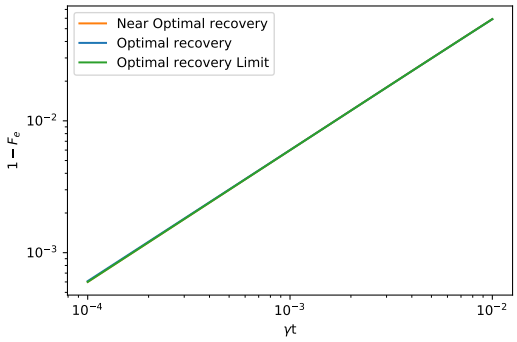
to be completed by a small on board computer, as both the memory and processing power requirements are within the capabilities of modern mobile devices.

The simulation and investigation performed on the SDP optimal recovery from for the completion of this thesis has been discouraging for its implementation into realistic systems. The requirements of a computation system to perform these simulations appears to be far above any reasonably applicable limit with current technology. Additionally processing speed and duration of an SDP optimisation are far too slow for the current societal expectation of measurement and sensing equipment, having to spend upwards of 20 waiting for a device to calibrate would likely restrict the use of the device to a very limited consumer base. As a result, with current technology and societal expectations, the SDP optimisation method for error recovery appears to be obsolete.

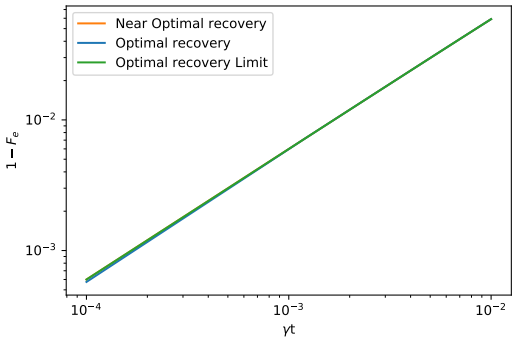
6.2.3 Local Symmetric Noise

By taking the assumption in section 4.3 and discarding all outputs outside the symmetric space, i.i. treating it as leakage error, we were able to attempt a reduced accuracy simulation of the quantum codes under Local Symmetric Dephasing. Unfortunately all the results (fig6.8) returned the same value hence either the simulation is invalid and there is a major flaw or the error model acts identically on all four codes.

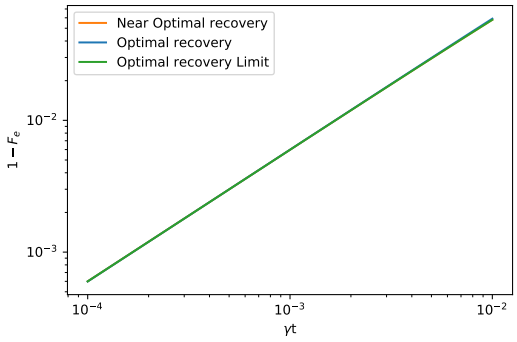
By comparing the results of the local symmetric noise and the global symmetric noise fig6.9, we can also see that despite the losses of large amounts of state information the local symmetric noise appears to still allow better performance than the global symmetric noise model. We believe that this is a result of the global noise model having the intensity of the noise go as proportional to $N/2$. This is used to increase to the amount of error per qubit introduced, but effectively make the noise worse the larger your system, where the local symmetric noise acts on the qubits directly, resulting in a consistent amount of error regardless of system size.



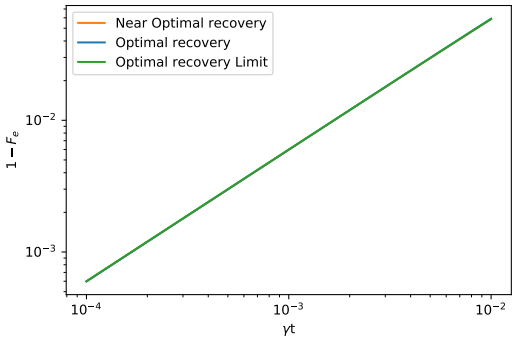
(a) Gross code results under local noise



(b) GNU code results under local noise



(c) Minimal qudit code results under local noise



(d) Qudit GKP code results under local noise

Figure 6.8: Recovery results for the four codes under local symmetric noise

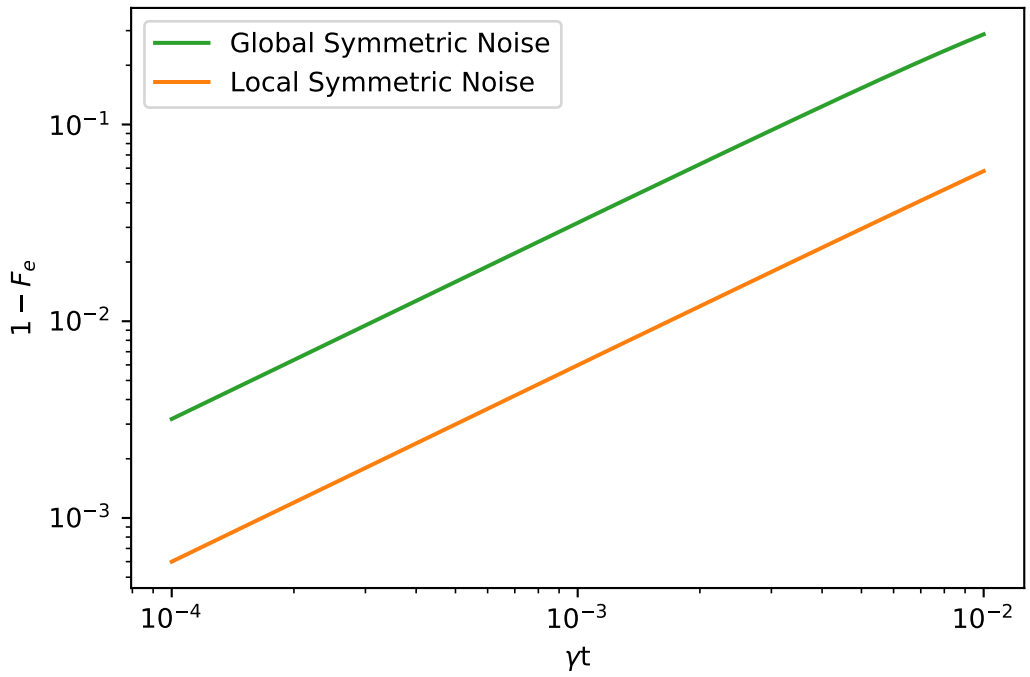


Figure 6.9: Comparison of limit of optimal recovery for the Minimal Qudit code under Global Symmetric Noise and Local Symmetric Noise

The measure of greatness in a scientific idea is the extent to which it stimulates thought and opens up new lines of research.

Paul Dirac

7 Conclusion

In this thesis we have discussed the applicability of Dicke state quantum error codes in modern systems, as the many experimental teams approach a realistically implementable quality of coupled spins. We considered two recently proposed error correction codes for the Dicke state, the GNU code being designed as a variable code and the Gross code being easily implementable as it is derived from a representation of Clifford operators. We also introduce two multi-level qudit codes, as the symmetric subspace of the Dicke states is structured as a multi-level system. This comparison is useful for analysis to some extent, but not strictly accurate as discussed, a selection of error models for Dicke states, we observe that a realistic model may take a Dicke state out of the symmetric space, such as in the local symmetric dephasing model. Further analysing these models, we observe the rise in dimensional representation as we move towards more realistic models, and this brings us to the problem of computational requirement. In our analysis of the error recovery, we note that the first order SCS solver for a semidefinite program optimisation has scaling $O(N^{6.2})$, when we take this scaling into account we observe that even doubling the dimension of the problem results in an

enormous increase in both processing time and required processing capabilities. As a result complex models for noise are likely to require far more processing power than is realistically available if the system is expected to ever be implemented into a consumer product. Further from our results on the global symmetric noise model it appears that the difference in performance between the limit of the optimal SDP recovery and the Barnum-Knill near optimal recovery is negligible, which considering the most computationally complex step in the Barnum-Knill recovery is the pseudo inverse which using the Moore-Penrose method [49] has order $O(N^3)$ which is a vast improvement over the SDP method.

As a result of this it would appear that the SDP optimal recovery is obsolete for Global Symmetric noise, and with further investigation may be proven generally obsolete for realistic implementation.

Further work remains in this field, as these simulations do not include any part of the state preparation or unitary evolution, which may be sources of further error in the system. Additionally with improved resources and improvements in available technology at some point in the future a deeper exploration into the applicability of the SDP optimisation method may be possible, but as mentioned in Ref [34] even a powerful server was incapable of processing large semidefinite programs, so it is still doubtful.

For future work we would like to incorporate different error models into our simulation system, as current technology progresses and research groups start publishing information on error sources in Dicke state systems it will be highly useful to develop simulations to test quantum codes against these models in the hopes of optimising systems without the need to physical modification. Further with consideration for optimising the performance of our simulations and improvement on formatting we could eventually release our Python code as a package for public use.

With these potential future steps in mind further research into the field of quantum error recovery simulation could yield major developments in sensing and computing applications. While recovery systems are not particularly optimised for large computational arrays they may be useful in smaller computational systems and may especially be introduced into quantum sensors.

A

An Appendix

all results and code used for simulations can be found in the github repository at:

https://github.com/DanielMMadden/Error-recovery-analysis-

Here we present the value of the pulse parameters $\{(\beta_s, \theta_s, \phi_s, \chi_s\}$ to prepare $|0_L\rangle$ for the QECC considered in this thesis.

Table A.1: Geometric Phase Gate parameters for the preparation of the Qudit GKP and (4,4,1)-gnu codes in state $|0_L\rangle$

	GKP code				(4,4,1)-GNU code			
Pulse	2χ	θ	ϕ	β	2χ	θ	ϕ	β
1	-0.1233	0.0124	-0.0818	-0.5971	0.2447	-0.0201	-1.6272	-0.3335
2	1.1405	-0.4537	-0.0098	0.4314	-1.3454	1.2635	0.171	1.0429
3	0.2763	-0.4048	2.0038	0.7946	0.5045	-1.5403	0.012	1.4669
4	0.8227	1.3656	0.4506	-0.6146	-1.8822	-2.2335	-0.5823	2.454
5	0.0504	1.052	-0.0957	0.4977	-1.8975	-0.3556	0.3149	0.4699
6	0.4692	0.5755	-0.3845	-0.831	-0.1072	-0.0106	-0.361	0.6161
7	-0.266	1.1882	-1.8052	-0.1984	-0.0201	-2.201	-1.753	-0.0644
8	-1.6692	-0.625	-0.0241	0.439	-0.4677	0.8732	2.5003	-0.6376
9	0.4027	-0.4679	-0.7088	0.3335	-0.5301	-1.5306	-0.1429	-0.0377
10	1.3044	-0.3803	-1.1434	1.0654	0.2837	-0.7288	-0.0787	0.482
11	-1.6021	-0.1391	-0.002	0.7244	-0.12	1.9464	0.3631	0.0098
12	-1.8165	-0.6654	-0.1987	-1.2736	0.4735	-1.3692	0.6005	-1.1039
13	-1.3756	-1.239	0.2347	0.7614	-0.4219	0.9707	0.0206	-0.7858
14	-1.4586	-0.6753	-1.2301	1.321	1.8765	-0.8508	0.9005	1.5444
15	0.4545	0.1219	1.9681	0.1412	-0.9424	1.3285	-1.5833	-0.544
16	0.3224	0.7363	0.4191	-1.2395				

Table A.2: Geometric Phase Gate parameters for the preparation of the Gross and the minimal qudit codes in state $|0_L\rangle$

	Gross code				Minimal Qudit code			
Pulse	2χ	θ	ϕ	β	2χ	θ	ϕ	β
1	0.0613	0.7402	0.0734	-1.9973	-1.5646	-1.1685	-1.2266	0.5843
2	0.3034	0.8949	3.6876	-0.6345	4.103	-0.008	-0.2835	1.571
3	-0.2308	-0.5987	-0.0173	-0.2438	-2.3875	-0.0648	0.7833	-0.3899
4	-1.0099	-0.2304	0.8992	0.0793	0.1914	0.5247	1.3362	0.3074
5	1.7161	0.4125	0.8108	1.1202	-1.2474	0.6431	0.0474	-0.6487
6	0.7099	2.0693	-1.0015	-0.0528	0.8506	1.6811	1.3123	-0.0642
7	-1.3356	0.7678	0.6663	0.2627	-0.8262	-0.4093	1.5705	-0.9054
8	0.549	0.4559	0.4416	1.114	0.101	0.1331	-0.9434	-0.4152
9	4.0402	-0.0607	0.2763	2.2234	-0.6311	-0.0016	2.0422	-2.2682
10	0.4435	-0.1031	-0.0065	-0.2788	3.7256	0.1686	0.5227	-1.1184
11	-1.3305	1.0307	2.4741	-1.9969	-0.3328	-0.6139	0.0376	0.5249
12	-1.2757	0.4177	3.4371	0.2975	-0.4589	1.1785	0.237	-0.4703
13	1.8606	1.5984	0.3001	-2.2792	-1.5456	-0.8233	-1.3002	0.9725
14	-0.2691	-0.9268	1.7335	1.7667	-0.5387	-0.6451	0.0134	-1.9447
15	-0.4859	-0.7265	1.056	-1.402	-0.708	0.8568	-0.0749	-0.3679
16	2.1251	0.5072	-0.1127	-2.7303	-1.3933	-1.7163	0.00447	-0.1045

Bibliography

[1] MOSEK ApS. *The MOSEK optimization toolbox for MATLAB manual. Version 9.0.* 2019. URL: http://docs.mosek.com/9.0/toolbox/index.html.

[2] Frank Arute et al. "Quantum Supremacy using a Programmable Superconducting Processor". In: *Nature* 574 (2019), pp. 505–510. URL: https://www.nature.com/articles/s41586-019-1666-5.

[3] Koenraad Audenaert and Bart De Moor. "Optimizing completely positive maps using semidefinite programming". In: *Physical Review A* 65.3 (Feb. 2002). ISSN: 1094-1622. DOI: 10.1103/physreva.65.030302. URL: http://dx.doi.org/10.1103/PhysRevA.65.030302.

[4] Ben Q. Baragiola, Bradley A. Chase, and JM Geremia. "Collective uncertainty in partially polarized and partially decohered spin-$\frac{1}{2}$ systems". In: *Phys. Rev. A* 81 (3 Mar. 2010), p. 032104. DOI: 10.1103/PhysRevA.81.032104. URL: https://link.aps.org/doi/10.1103/PhysRevA.81.032104.

[5] Ben Q. Baragiola, Bradley A. Chase, and JM Geremia. "Collective uncertainty in partially polarized and partially decohered spin-12systems". In: *Physical Review A* 81.3 (Mar. 2010). ISSN: 1094-1622. DOI: 10.1103/physreva.81.032104. URL: http://dx.doi.org/10.1103/PhysRevA.81.032104.

[6] H. Barnum and E. Knill. *Reversing quantum dynamics with near-optimal quantum and classical fidelity*. 2000. arXiv: quant-ph/0004088 [quant-ph].

[7] Andreas Bärtschi and Stephan Eidenbenz. "Deterministic Preparation of Dicke States". In: *Lecture Notes in Computer Science* (2019), pp. 126–139. ISSN: 1611-3349. DOI:

10.1007/978-3-030-25027-0_9. URL: http://dx.doi.org/10.1007/978-3-030-25027-0_9.

[8] Claude Berrou, Alain Glavieux, and Punya Thitimajshima. "Near Shannon limit error-correcting coding and decoding: Turbo-codes. 1". In: vol. 2. June 1993, 1064–1070 vol.2. ISBN: 0-7803-0950-2. DOI: 10.1109/ICC.1993.397441.

[9] H. Bombin. *An Introduction to Topological Quantum Codes*. 2013. arXiv: 1311.0277 [quant-ph].

[10] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. New York: Cambridge University Press, 2004.

[11] Samuel L. Braunstein and Carlton M. Caves. "Statistical distance and the geometry of quantum states". In: *Phys. Rev. Lett.* 72 (22 May 1994), pp. 3439–3443. DOI: 10.1103/PhysRevLett.72.3439. URL: https://link.aps.org/doi/10.1103/PhysRevLett.72.3439.

[12] G. K. Brennen et al. *Global operations for protected quantum memories in atomic spin lattices*. 2009. arXiv: 0901.3920 [quant-ph].

[13] Richard Brito, Vitor Cardoso, and Paolo Pani. "Superradiance". In: *Lecture Notes in Physics* (2020). ISSN: 1616-6361. DOI: 10.1007/978-3-030-46622-0. URL: http://dx.doi.org/10.1007/978-3-030-46622-0.

[14] Carlo Cafaro, Federico Maiolini, and Stefano Mancini. "Quantum stabilizer codes embedding qubits into qudits". In: *Phys. Rev. A* 86 (2 Aug. 2012), p. 022308. DOI: 10.1103/PhysRevA.86.022308. URL: https://link.aps.org/doi/10.1103/PhysRevA.86.022308.

[15] A. A. Clerk et al. "Introduction to quantum noise, measurement, and amplification". In: *Reviews of Modern Physics* 82.2 (Apr. 2010), pp. 1155–1208. ISSN: 1539-0756. DOI: 10.1103/revmodphys.82.1155. URL: http://dx.doi.org/10.1103/RevModPhys.82.1155.

[16] B. Jack Copeland. *The modern history of computing*. June 2006. URL: https://plato.stanford.edu/entries/computing-history/.

[17] C. L. Degen, F. Reinhard, and P. Cappellaro. "Quantum sensing". In: *Rev. Mod. Phys.* 89 (3 July 2017), p. 035002. DOI: 10.1103/RevModPhys.89.035002. URL: https://link.aps.org/doi/10.1103/RevModPhys.89.035002.

[18] R. Derka, V. Buzek, and A. K. Ekert. “Universal Algorithm for Optimal Estimation of Quantum States from Finite Ensembles via Realizable Generalized Measurement”. In: *Physical Review Letters* 80.8 (Feb. 1998), pp. 1571–1575. ISSN: 1079-7114. DOI: 10.1103/physrevlett.80.1571. URL: http://dx.doi.org/10.1103/PhysRevLett.80.1571.

[19] Steven Diamond and Stephen Boyd. “CVXPY: A Python-embedded modeling language for convex optimization”. In: *Journal of Machine Learning Research* 17.83 (2016), pp. 1–5.

[20] Jaromír Fiurášek. “Extremal equation for optimal completely positive maps”. In: *Physical Review A* 64.6 (Nov. 2001). ISSN: 1094-1622. DOI: 10.1103/physreva.64.062310. URL: http://dx.doi.org/10.1103/PhysRevA.64.062310.

[21] Andrew S. Fletcher, Peter W. Shor, and Moe Z. Win. “Optimum quantum error recovery using semidefinite programming”. In: *Physical Review A* 75.1 (Jan. 2007). ISSN: 1094-1622. DOI: 10.1103/physreva.75.012338. URL: http://dx.doi.org/10.1103/PhysRevA.75.012338.

[22] Michael Garstka, Mark Cannon, and Paul Goulart. “COSMO: A Conic Operator Splitting Method for Convex Conic Problems”. In: *Journal of Optimization Theory and Applications* 190.3 (Aug. 2021), pp. 779–810. ISSN: 1573-2878. DOI: 10.1007/s10957-021-01896-x. URL: http://dx.doi.org/10.1007/s10957-021-01896-x.

[23] Iulia Georgescu. “Trapped ion quantum computing turns 25”. In: *Nature Reviews Physics* 2.6 (2020), pp. 278–278. ISSN: 2522-5820. DOI: 10.1038/s42254-020-0189-1. URL: https://doi.org/10.1038/s42254-020-0189-1.

[24] David J. Griffiths. *Introduction to Quantum Mechanics (2nd Edition)*. Pearson Prentice Hall, 2004. URL: http://www.amazon.com/exec/obidos/redirect?tag=citeulike07-20%5C&path=ASIN/0131118927.

[25] Jonathan A. Gross. “Designing Codes around Interactions: The Case of a Spin”. In: *Physical Review Letters* 127.1 (July 2021). ISSN: 1079-7114. DOI: 10.1103/physrevlett.127.010504. URL: http://dx.doi.org/10.1103/PhysRevLett.127.010504.

[26] Lov K. Grover. *A fast quantum mechanical algorithm for database search*. 1996. arXiv: quant-ph/9605043 [quant-ph].

[27] R. W. Hamming. "Error detecting and error correcting codes". In: *The Bell System Technical Journal* 29.2 (1950), pp. 147–160. DOI: 10.1002/j.1538-7305.1950.tb00463.x.

[28] Klaus Hepp and Elliott H. Lieb. "Equilibrium Statistical Mechanics of Matter Interacting with the Quantized Radiation Field". In: *Phys. Rev. A* 8 (5 Nov. 1973), pp. 2517–2525. DOI: 10.1103/PhysRevA.8.2517. URL: https://link.aps.org/doi/10.1103/PhysRevA.8.2517.

[29] Mattias T. Johnsson et al. "Geometric Pathway to Scalable Quantum Sensing". In: *Phys. Rev. Lett.* 125 (19 Nov. 2020), p. 190403. DOI: 10.1103/PhysRevLett.125.190403. URL: https://link.aps.org/doi/10.1103/PhysRevLett.125.190403.

[30] Emanuel Knill and Raymond Laflamme. "Theory of quantum error-correcting codes". In: *Phys. Rev. A* 55 (2 Feb. 1997), pp. 900–911. DOI: 10.1103/PhysRevA.55.900. URL: https://link.aps.org/doi/10.1103/PhysRevA.55.900.

[31] Andrew J. Landahl, Jonas T. Anderson, and Patrick R. Rice. *Fault-tolerant quantum computing with color codes*. 2011. arXiv: 1108.5738 [quant-ph].

[32] Moonjoo Lee et al. "Ion-Based Quantum Sensor for Optical Cavity Photon Numbers". In: *Phys. Rev. Lett.* 122 (15 Apr. 2019), p. 153603. DOI: 10.1103/PhysRevLett.122.153603. URL: https://link.aps.org/doi/10.1103/PhysRevLett.122.153603.

[33] Michael A Nielsen. "A simple formula for the average gate fidelity of a quantum dynamical operation". In: *Physics Letters A* 303.4 (2002), pp. 249–252. ISSN: 0375-9601. DOI: https://doi.org/10.1016/S0375-9601(02)01272-0. URL: https://www.sciencedirect.com/science/article/pii/S0375960102012720.

[34] Brendan O'Donoghue et al. *Conic Optimization via Operator Splitting and Homogeneous Self-Dual Embedding*. 2016. arXiv: 1312.3039 [math.OC].

[35] Yingkai Ouyang. "Permutation-invariant quantum codes". In: *Phys. Rev. A* 90 (6 Dec. 2014), p. 062317. DOI: 10.1103/PhysRevA.90.062317. URL: https://link.aps.org/doi/10.1103/PhysRevA.90.062317.

[36] James L. Park. "The concept of transition in quantum mechanics". In: *Foundations of Physics* 1.1 (Mar. 1970), pp. 23–33. DOI: 10.1007/BF00708652.

[37] Stefano Pirandola et al. “Minimal qudit code for a qubit in the phase-damping channel”. In: *Physical Review A* 77 (June 2007). DOI: 10.1103/PhysRevA.77.032309.

[38] R. Prevedel et al. “Experimental Realization of Dicke States of up to Six Qubits for Multiparty Quantum Networking”. In: *Physical Review Letters* 103.2 (July 2009). ISSN: 1079-7114. DOI: 10.1103/physrevlett.103.020503. URL: http://dx.doi.org/10.1103/PhysRevLett.103.020503.

[39] R. Prevedel et al. “Experimental Realization of Dicke States of up to Six Qubits for Multiparty Quantum Networking”. In: *Phys. Rev. Lett.* 103 (2 July 2009), p. 020503. DOI: 10.1103/PhysRevLett.103.020503. URL: https://link.aps.org/doi/10.1103/PhysRevLett.103.020503.

[40] I. S. Reed and G. Solomon. “Polynomial Codes Over Certain Finite Fields”. In: *Journal of the Society for Industrial and Applied Mathematics* 8.2 (1960), pp. 300–304. DOI: 10.1137/0108018. eprint: https://doi.org/10.1137/0108018. URL: https://doi.org/10.1137/0108018.

[41] Mary Beth Ruskai. “Pauli Exchange Errors in Quantum Computation”. In: *Phys. Rev. Lett.* 85 (1 July 2000), pp. 194–197. DOI: 10.1103/PhysRevLett.85.194. URL: https://link.aps.org/doi/10.1103/PhysRevLett.85.194.

[42] J. J. Sakurai and Jim Napolitano. *Modern Quantum Mechanics*. 3rd ed. Cambridge University Press, 2020. DOI: 10.1017/9781108587280.

[43] C. E. Shannon. “A mathematical theory of communication”. In: *The Bell System Technical Journal* 27.3 (1948), pp. 379–423. DOI: 10.1002/j.1538-7305.1948.tb01338.x.

[44] P.W. Shor. *Algorithms for quantum computation: discrete logarithms and factoring*. 1994. DOI: 10.1109/SFCS.1994.365700.

[45] Peter W. Shor. “Scheme for reducing decoherence in quantum computer memory”. In: *Phys. Rev. A* 52 (4 Oct. 1995), R2493–R2496. DOI: 10.1103/PhysRevA.52.R2493. URL: https://link.aps.org/doi/10.1103/PhysRevA.52.R2493.

[46] Andrew Steane. “Multiple-Particle Interference and Quantum Error Correction”. In: *Proceedings: Mathematical, Physical and Engineering Sciences* 452.1954 (1996), pp. 2551–2577. ISSN: 13645021. URL: http://www.jstor.org/stable/52827.

[47] John K. Stockton, Ramon van Handel, and Hideo Mabuchi. "Deterministic Dicke-state preparation with continuous measurement and control". In: *Phys. Rev. A* 70 (2 Aug. 2004), p. 022106. DOI: 10.1103/PhysRevA.70.022106. URL: https://link.aps.org/doi/10.1103/PhysRevA.70.022106.

[48] K. C. Toh, M. J. Todd, and R. H. Tütüncü. "SDPT3 — A Matlab software package for semidefinite programming, Version 1.3". In: *Optimization Methods and Software* 11.1-4 (1999), pp. 545–581. DOI: 10.1080/10556789908805762. eprint: https://doi.org/10.1080/10556789908805762. URL: https://doi.org/10.1080/10556789908805762.

[49] Qiujuan Tong et al. "A Fast Algorithm of Moore-Penrose Inverse for the Symmetric Loewner-Type Matrix". In: *2009 International Conference on Information Engineering and Computer Science*. 2009, pp. 1–4. DOI: 10.1109/ICIECS.2009.5364014.

[50] Yuchen Wang et al. "Qudits and High-Dimensional Quantum Computing". In: *Frontiers in Physics* 8 (2020). ISSN: 2296-424X. DOI: 10.3389/fphy.2020.589504. URL: https://www.frontiersin.org/article/10.3389/fphy.2020.589504.

[51] K. Wright et al. "Benchmarking an 11-qubit quantum computer". In: *Nature Communications* 10.1 (Nov. 2019). ISSN: 2041-1723. DOI: 10.1038/s41467-019-13534-2. URL: http://dx.doi.org/10.1038/s41467-019-13534-2.