# Extending quantum error correction: new continuous measurement protocols and improved fault-tolerant overhead

Thesis by
Charlene Sonja Ahn

In Partial Fulfillment of the Requirements
for the Degree of
Doctor of Philosophy


California Institute of Technology
Pasadena, California

2004

(Defended May 3, 2004)

## Acknowledgements

A great deal of thanks and praise is due my advisor, John Preskill, who not only has been exceptionally generous with his time and energy but also has greatly encouraged and aided me in following my own interests and collaborations while still remaining interested in my work.

Additionally, I have been very fortunate to be able to work with Gerard Milburn and Howard Wiseman during parts of two summers spent at the University of Queensland. These visits were enlightening and productive. Both were always ready with advice and help when needful; their ideas inspired the work in Chapters 5 and 6. Howard Wiseman has been especially helpful in always encouraging me to find more general formulations of my work.

I would like to thank the members of the Institute for Quantum Information, who have made Caltech into a very exciting and interesting place to research quantum information theory. In particular, the guidance of Andrew Doherty and Patrick Hayden has been invaluable, as they practically acted as associate advisors for several years. In addition, I would like to thank not only the above, but also Jim Harrington and Andrew Landahl, along with Mohan Sarovar of the University of Queensland and Kurt Jacobs of Griffith University, for rewarding collaborations. Mohan Sarovar, among other things, has been tremendously helpful in providing simulation results testing various statements in Chapter 5, some at extremely short notice. The physical and mathematical insight, keen critiquing skills, and friendship of Kevin Birnbaum, Sumit Daftuar, Carlos Mochon, Ben Rahn, and Ben Toner have been indispensable in too many ways to list here. Finally, I thank Dave Beckman for not only enlightening discussions but also unwavering support and care, even while he was stressing about his own thesis.

**Extending quantum error correction: new continuous measurement protocols and improved fault-tolerant overhead**

by

Charlene Sonja Ahn

In Partial Fulfillment of the
Requirements for the Degree of
Doctor of Philosophy

## Abstract

Quantum mechanical applications range from quantum computers to quantum key distribution to teleportation. In these applications, quantum error correction is extremely important for protecting quantum states against decoherence. Here I present two main results regarding quantum error correction protocols.

The first main topic I address is the development of continuous-time quantum error correction protocols via combination with techniques from quantum control. These protocols rely on weak measurement and Hamiltonian feedback instead of the projective measurements and unitary gates usually assumed by canonical quantum error correction. I show that a subclass of these protocols can be understood as a quantum feedback protocol, and analytically analyze the general case using the stabilizer formalism; I show that in this case perfect feedback can perfectly protect a stabilizer subspace. I also show through numerical simulations that another subclass of these protocols does better than canonical quantum error correction when the time between corrections is limited.

The second main topic is development of improved overhead results for fault-tolerant computation. In particular, through analysis of topological quantum error correcting codes, it will be shown that the required blowup in depth of a noisy circuit performing a fault-tolerant computation can be reduced to a factor of $O(\log \log L)$, an improvement over previous results. Showing this requires investigation into a local method of performing fault-tolerant correction on a topological code of arbitrary dimension.

# Contents

# List of Figures

# Chapter 1

# Introduction

Quantum mechanics has existed as a theory of nature for almost a century, but only in the last twenty or so years has much of research focused on the question: What are quantum states good for, anyway? More precisely, what tasks can be performed using quantum states that go beyond what can be done with classical states alone?

By 1982, R. P. Feynman had suggested that a quantum computer might be able to simulate quantum systems more efficiently than classical computers [25], and in 1985 D. Deutsch constructed a quantum algorithm able to solve a small problem faster than any classical algorithm [17]. At around the same time, in 1984, C. Bennett and G. Brassard introduced the provably secure scheme for quantum key distribution now known as BB84 [9] (although the proof of security took more than ten years after that to be realized [60, 57, 79]).

In the early nineties, it became clear that quantum states could be utilized in performing a host of exciting and bizarre tasks. Quantum teleportation, discovered by Bennett *et al.* in 1993 [10], is a nifty application in which a quantum state, with the help of an additional entangled state and classical communication, can be transmitted even without any sort of quantum communication channel. Perhaps the most well-known and exciting application of quantum states is Shor's algorithm for factoring numbers in polynomial time in the number of digits, which represented the first exponential speedup of a quantum algorithm relative to the best known classical algorithm [78].

What gives quantum states these myriad powers? Classical states can take on discrete values— a classical bit, for example, takes on the values 0 and 1— but cannot take on coherent combinations of those values. Quantum states, on the other hand, are represented

by elements of a Hilbert space: a quantum bit, or qubit, can take on any value in a two-dimensional Hilbert space. In particular, it can take on 0 or 1 values, just as a classical bit can, and those values are represented in the Hilbert space as

$$\begin{aligned} |0\rangle &= \begin{pmatrix} 1 \\ 0 \end{pmatrix} \\ |1\rangle &= \begin{pmatrix} 0 \\ 1 \end{pmatrix}. \end{aligned} \tag{1.1}$$

A qubit can moreover exist in a *coherent superposition* of 0 and 1:

$$|\psi\rangle = a|0\rangle + b|1\rangle \equiv \begin{pmatrix} a \\ b \end{pmatrix}, \tag{1.2}$$

where $a$ and $b$ are complex numbers with $|a|^2 + |b|^2 = 1$.

One way of understanding the power of these quantum states (albeit an oversimplified and incomplete way) is to think of them as being able to perform a sort of parallel processing on 0's and 1's simultaneously. In this way a quantum algorithm such as Deutsch's or Shor's is able to extract global information about a problem that a classical computer might take much longer to extract.

Unfortunately, we do not have access to all the information encoded in a quantum state; in that case we could store an infinite amount of data in the complex numbers $a$ and $b$ in equation (1.2)! Another step is necessary to interact with the quantum system: it must be measured in some way. In the two-dimensional basis shown above, the measurement will give one of two values. For example, measuring the state in (1.2) in the computational basis will give either a 0, with probability $|a|^2$, or a 1 with probability $|b|^2$. If a 0 is obtained, the state after the measurement will be $|0\rangle$, while $|1\rangle$ is the resulting state after measuring a 1. Notice that neither of those two states are the same as the original. This example illustrates a more general principle: gaining information about a state disturbs it.

Because these delicate superpositions are easily disturbed by noise, controlling and protecting these quantum states becomes an interesting and difficult problem. Indeed, overcoming the effects of strong decohering interactions with the environment is a major hurdle

faced by experimenters studying quantum systems. Classically, error correction is much easier to do. Let us assume that given $n$ bits, errors happen independently and infrequently on each bit with probability $p$. In the presense of errors that may cause a bit with value 0 to flip to the value 1, or conversely from 1 to 0 (a "bit flip"), logical bits may be encoded in, for example, a simple repetition code that will act to protect them from errors. For example, a 0 could be encoded in three zeroes; then even if one of the bits is flipped to a 1, majority voting will act to correct that error. If $p$ is small enough and error correction is fast enough, most errors will be of this one-bit form and will be corrected. In particular, the effective error probability on the encoded bit is reduced from $p$ to an expression of order $p^2$.

Error correction with regard to quantum states is more difficult for several reasons. We run into problems right away: a simple repetition coding cannot be implemented directly for arbitrary superpositions of quantum states even in principle, because arbitrary superpositions cannot be *cloned*.

Even if this sort of repetition coding, or some variant, could be done, there are even more issues. Quantum states admit not only bit-flip errors but also phase-flip errors, in which a state $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ could be flipped to $\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$ and vice versa. Even worse, phase and bit flips are not the only sorts of errors; a quantum state $a|0\rangle + b|1\rangle$ is susceptible to very small errors in the values of $a$ and $b$, and any quantum error correction scheme must be able to correct not just bit or phase flips, but also these possibly very small errors. Finally, the classical repetition code assumed implicitly that the errors could be measured. Classically, of course, this assumption is a trivial matter because the act of measuring the state to determine the error has no effect on the state. However, when measuring quantum states, the very measurements that are performed to determine the error that has occured may disturb the state that needs to be protected.

These concerns were put to rest in 1996, when P. Shor discovered the first known quantum error correction code that could correct not only for bit flips but also for phase flips, and indeed for an arbitrary error on any physical qubit in the code [76] [1]. These quantum

[1] There are several other tools that have been developed to protect quantum states from decoherence, notably decoherence-free subspaces [56] and dynamic decoupling methods [85]. In this thesis we will restrict our attention to quantum error correcting codes.

error correction codes are designed not to require cloning of arbitrary superpositions. Like classical error-correcting codes, quantum error correcting codes work by redundantly encoding quantum information across many quantum systems. The key to this approach is the use of measurements that discretize errors onto a finite set and additionally reveal information about the error rather than about the encoded data. This feature is particularly useful for protecting the unknown quantum states that appear frequently in the course of quantum computations. The physical tools used in this approach are projective von Neumann measurements that discretize the errors and fast unitary gates that restore corrupted data.

Often it is necessary to consider slightly weaker tools. In many quantum systems, the information gained about the quantum state from measurements comes from continuous measurements which give very little information about a state in an infinitesimal time interval, and thus disturb the state very little in that time interval. One might also postulate that the system might be controlled using not unitary gates that change a state suddenly, but rather bounded-strength Hamiltonians that rotate a state in finite time. These sorts of tools are the domain of quantum feedback control.

Quantum control, loosely speaking, is the art of getting a quantum system to do what you want it to do in the presence of various restrictions on how much it is possible to find out about the system and how powerful the controls are allowed to be. There is a large and diverse literature on the theory of quantum feedback control, ranging from practical experimental protocols (e.g., [88]) to abstract theoretical models (e.g., [84]) to combinations of the two (e.g., adaptive quantum measurement of optical phase in [94, 97], experimentally implemented in [5]. The information about the quantum state fed into the controller typically comes from continuous measurements, and the operations the controller applies in response are typically bounded-strength Hamiltonians. In between the measurements and the subsequent applied operations, there may be varying amounts of processing of the measurement results. One well-studied type of feedback relies only on instantaneous measurement results and thus is called Markovian feedback, or Wiseman-Milburn feedback after H. M. Wiseman and G. J. Milburn, who developed the associated formalism [93, 98]. Feedback could also depend on parameters extracted from reconstructing the state of the system due to information gained from the measurement; this type of

feedback is often referred to as state-estimate feedback. Discussing some of the formalism related to these two ways of performing feedback will be the subject of Chapter 3.

Quantum error correction and quantum feedback both rely on performing operations that are conditioned on the result of some measurement on the system, which suggests that exploring the links between these two techniques adds to our understanding of both processes, and may lead to insights into future protocols and experimental implementations. Although these two subjects are similar in broad outline, little work had been done on combining them before the work presented in this thesis. Previous work to account for continuous time using quantum error correction has focused on "automatic" recovery and decoherence modeling but had neglected the role of continuous measurement [7, 14, 66]. Quantum feedback control had, like quantum error correction, been thought of as a tool to protect quantum states, but previous work on quantum state protection using quantum feedback control had focused on protection and preparation of known states and had not addressed the issue of protecting unknown quantum states [89, 53]. The first protocol for continuous quantum error correction via quantum feedback control was a state-estimation protocol given in [2]. This protocol is robust and can be constructed to be optimal; however, it uses a great deal of classical side-processing to obtain that optimal and robust feedback. I will present this protocol and simulations of it in Chapter 4, along with some results I have derived on showing knowledge of the original quantum state is not required, as well as some new simulation results on relaxing some of the model assumptions.

In Chapters 5 and 6, I will then present two new protocols for continuous quantum error correction via quantum feedback control, each of which has its own advantages and disadvantages. One uses less classical processing than the above protocol but is still robust; its drawback is that it only works when the measurement and correction strengths are above a certain threshold. The last can be formulated analytically in terms of the stabilizer formalism because it uses Markovian feedback, which also implies that it requires no side processing whatsoever; to make use of this formalism, however, the error correction protocol must be mutated so that it only corrects for a specific sort of error process. Furthermore, the Markovian nature of the protocol means that it is not robust to measurement inefficiency.

Instead of considering the tools used to perform quantum error correction, let us now consider quantum error correction itself used as a tool for performing quantum computation

with very little error. When combined with fault-tolerant techniques, and when all noise sources are below a critical value known as the accuracy threshold, quantum error correction enables quantum computations of arbitrary length with arbitrarily small output error, also known as fault-tolerant quantum computation. The achievability of fault tolerance leads naturally to the question: how much overhead is required in order to make a circuit fault-tolerant? Suppose that a classical circuit of reliable gates with size $L$ and depth $D$ computes a particular Boolean function. We wish to compute the same function using a circuit of noisy gates. Suppose that the noisy gates fail independently with a probability of failure $\epsilon$ and that the function is to be computed with a probability of error less than $\delta$. Classically, for any fixed positive $\delta$ and for sufficiently small fixed positive $\epsilon$, an equivalent circuit with noisy gates can be chosen such that its size $L^*$ and depth $D^*$ are

$$\begin{aligned} L^* &= O(L \log L) \ , \\ D^* &= O(D) \ ; \end{aligned} \tag{1.3}$$

the cost of achieving fault tolerance is that the size of the circuit blows up by a factor of order $\log L$, and the depth by a constant factor. This result was suggested by Von Neumann [87] in 1952, and proved by Dobrushin and Ortyakov [18] in 1977. Explicit circuit constructions realizing this blowup were first achieved by Pippenger [67].

What is the corresponding statement about simulating a quantum circuit using noisy quantum gates? Pippenger [68] has conjectured that the blowup in the quantum case is

$$\begin{aligned} L^* &= O(L \log^2 L) \ , \\ D^* &= O(D) \ . \end{aligned} \tag{1.4}$$

The intuition underlying this proposal is that to achieve quantum fault tolerance we must control both bit flips and phase errors. If control of bit flips costs a factor of $\log L$ in size and a constant in depth, and control of phase errors has the same cost, we arrive at eq. (1.4). On the other hand, Aharonov and Ben-Or conjecture in the conclusion of [1] that the quantum cost in depth must be at least a factor $D* = O(\log \log L)$. Using *concatenated codes* (a hierarchy of codes within codes) it had been previously established that the blowup in both

size and depth is no worse than a factor polylogarithmic in $L$ [1, 23, 47, 72, 35]. Chapter 7 of this thesis will show that the blowup in depth can be reduced to a factor $O(\log \log L)$.

Another interesting computational model is one in which classical postprocessing of measurement outcomes that is polylog in $L$ is regarded as instantaneous. In fact, it was already known that with polylog classical processing, a constant blowup in quantum depth and a polylog blowup in quantum size can be achieved using concatenated coding. It will also be shown in Chapter 7 that topological coding methods can improve the power of $\log L$ in the blowup of the quantum size over previous results.

# Chapter 2

# Quantum error correcting codes

## 2.1 Introduction

One important tool that has been developed for protecting against decoherence is quantum error correction [76, 80, 50, 38], which is specifically designed for protecting unknown quantum states. In the usual protocol for quantum error correction, projective measurements are performed to acquire an error syndrome. A unitary operation chosen based on the results of the projective measurements is then applied to correct for the error.

More specifically, a binary quantum error correcting code (that is, one that can, by analogy with computer bits, be represented in terms of qubits; for simplicity I will only consider these binary codes) can be thought of as a $2^k$-dimensional subspace of a $2^n$-dimensional Hilbert space, together with a set of correction operations. We can think of this space as storing $k$ logical qubits in $n$ physical qubits; the redundancy involved in encoding $k$ logical qubits in a larger space may allow a certain set of errors to be corrected by measurement and application of the correction operations without disturbing the logical state. Some examples of quantum codes are given below, along with some helpful formalism. In particular, I will discuss the *stabilizer* formalism, a powerful group-theoretical method that provides an elegant and compact way to characterize quantum codes.

In the remainder of this work, I will use the notation of [38] in which $X$, $Y$, and $Z$ denote the Pauli matrices $\sigma_x$, $\sigma_y$ and $\sigma_z$ respectively, and juxtaposition denotes a tensor product; hence any element of the Pauli group

$$P_n = \{\pm 1, \pm i\} \otimes \{I, X, Y, Z\}^{\otimes n} \tag{2.1}$$

may be denoted as a concatenation of letters (*e.g.*, $ZZI = \sigma_z \otimes \sigma_z \otimes I$). For compactness of notation, I will also sometimes refer to an operator $A$ acting on the $(i)$th qubit alone as $A^{(i)}$.

### 2.1.1 Bit-flip code

The salient aspects of quantum error correction can already be seen in the three-qubit bit-flip code, even though it does not correct for arbitrary errors. The bit-flip code protects a single two-state quantum system from bit-flipping errors by mapping it onto the state of three qubits:

$$|0\rangle \rightarrow |000\rangle \equiv |\bar{0}\rangle \tag{2.2}$$

$$|1\rangle \rightarrow |111\rangle \equiv |\bar{1}\rangle. \tag{2.3}$$

The states $|\bar{0}\rangle$ and $|\bar{1}\rangle$ are called the *basis states* for the code and the space spanned by them is called the *codespace*, whose elements are called *codewords*.

The qubits are subjected to bit-flip noise: that is, the operators $XII, IXI, IIX$ are the only possible errors. After the qubits are subjected to noise, quantum error correction proceeds in two steps. First, the parities of neighboring qubits are projectively measured. These are the observables

$$M_0 = ZZI \tag{2.4}$$

$$M_1 = IZZ. \tag{2.5}$$

The *error syndrome* is the pair of eigenvalues $(m_0, m_1)$ returned by this measurement.

Once the error syndrome is known, the second step is to apply one of the following unitary operations conditioned on the error syndrome:

$$(-1,+1) \rightarrow XII \tag{2.6}$$

$$(-1,-1) \rightarrow IXI \tag{2.7}$$

$$(+1,-1) \rightarrow IIX \tag{2.8}$$

$$(+1,+1) \rightarrow III. \tag{2.9}$$

This procedure has two particularly appealing characteristics: the error syndrome measurement does not distinguish between the codewords, and the projective nature of the measurement discretizes all possible quantum errors onto a finite set. These properties hold for general quantum error correcting codes as well.

## 2.2 Bit-flip code in continuous time

For what follows, it is instructive to consider the bit-flip code in the picture of continuous-time evolution [2].

If the bit-flipping errors arise from reservoir-induced decoherence with some decoherence rate $1/\gamma$, then prior to quantum error correction the qubits evolve via the master equation

$$d\rho_{\text{noise}} = \gamma(\mathcal{D}[XII] + \mathcal{D}[IXI] + \mathcal{D}[IIX])\rho\, dt, \tag{2.10}$$

where $\gamma dt$ is the probability of a bit-flip error on each qubit per time interval $[t, t+dt]$, and where

$$\mathcal{D}[c]\rho = c\rho c^\dagger dt - \frac{1}{2}(c^\dagger c\rho + \rho c^\dagger c)dt. \tag{2.11}$$

is a superoperator representing the effects of these bit flips.

This master equation has the solution

$$\begin{aligned}\rho(t) = &\\ &a\,(t)\,\rho_0\\ &+b\,(t)\,(XII\rho_0XII + IXI\rho_0IXI + IIX\rho_0IIX)\\ &+c\,(t)\,(XXI\rho_0XXI + XIX\rho_0XIX + IXX\rho_0IXX)\\ &+d\,(t)\,XXX\rho_0XXX,\end{aligned} \tag{2.12}$$

where

$$\begin{aligned} a(t) &= \left(1 + 3e^{-2\gamma t} + 3e^{-4\gamma t} + e^{-6\gamma t}\right)/8 && (2.13)\\ b(t) &= \left(1 + e^{-2\gamma t} - e^{-4\gamma t} - e^{-6\gamma t}\right)/8 && (2.14)\\ c(t) &= \left(1 - e^{-2\gamma t} - e^{-4\gamma t} + e^{-6\gamma t}\right)/8 && (2.15)\\ d(t) &= \left(1 - 3e^{-2\gamma t} + 3e^{-4\gamma t} - e^{-6\gamma t}\right)/8. && (2.16) \end{aligned}$$

The functions $a(t)$–$d(t)$ express the probability that the system is left in a state that can be reached by zero, one, two, or three bit flips from the initial state, respectively. After quantum error correction is performed, single errors are identified correctly but double and triple errors are not. As a result, the recovered state, averaged over all possible measurement syndromes, is

$$\rho = \left(a\left(t\right) + b\left(t\right)\right)\rho_0 + \left(c\left(t\right) + d\left(t\right)\right) XXX\rho_0 XXX. \tag{2.17}$$

The overlap of this state with the initial state depends on the initial state, but is at least as large as when the initial state is $|\bar{0}\rangle$; namely, it is at least as large as

$$\begin{aligned} F_{\bar{3}} &= \left(2 + 3e^{-2\gamma t} - e^{-6\gamma t}\right)/4 \\ &\simeq 1 - 3(\gamma t)^2. && (2.18) \end{aligned}$$

Recalling that a single qubit subject to this decoherence has error probability $p = \gamma t$, we see that, when applied sufficiently often, the bit-flip code reduces the error probability on each logical qubit from $\mathcal{O}(p)$ to $\mathcal{O}(p^2)$.

## 2.3 A true quantum error correcting code: the Shor code

In fact, there are quantum codes that can protect not only against one specific kind of error, but against any arbitary error on a single qubit! A simple example was first described by Shor [76]. This code is a concatenation of the bit-flip code described previously and the phase-flip code; the phase-flip code protects against phase ($Z$) errors and can be formulated from the bit-flip code by changing all Z's into X's and all eigenstates of Z into eigenstates

of X. That is to say, the phase-flip code is given by the codestates

$$|\bar{0}\rangle = (|0\rangle + |1\rangle)(|0\rangle + |1\rangle)(|0\rangle + |1\rangle) \tag{2.19}$$

$$|\bar{1}\rangle = (|0\rangle - |1\rangle)(|0\rangle - |1\rangle)(|0\rangle - |1\rangle), \tag{2.20}$$

and the Shor code is given by the codestates

$$\begin{aligned} |\bar{0}\rangle &= \frac{(|000\rangle + |111\rangle)(|000\rangle + |111\rangle)(|000\rangle + |111\rangle)}{2\sqrt{2}} \\ |\bar{1}\rangle &= \frac{(|000\rangle - |111\rangle)(|000\rangle - |111\rangle)(|000\rangle - |111\rangle)}{2\sqrt{2}} \end{aligned} \tag{2.21}$$

This code can protect against a bit flip on one qubit; for example, if a bit flip occurs on the first qubit, measuring the operators $ZZIIIIIII$ and $IZZIIIIII$ will diagnose the error in the same way as in the bit-flip code, and the appropriate correction can be performed as in that code. Phase flips are managed similarly; a phase flip occurring on the first qubit will be diagnosed by measuring the operators $XXXXXXIII$ and $IIIXXXXXX$ and a correction can be performed. Finally, an error consisting of both a bit flip and a phase flip can be diagnosed by measuring both the sets of operators above.

From these observations, we can show that an arbitrary error on a single qubit can be corrected with this code. An arbitrary error on qubit $(i)$ can be written as

$$E^{(i)} = \epsilon_I I^{(i)} + \epsilon_X X^{(i)} + \epsilon_Y Y^{(i)} + \epsilon_Z Z^{(i)}. \tag{2.22}$$

Given a state in the codespace $|\bar{\psi}\rangle$, measuring the operators considered above on the state $E^{(i)}|\bar{\psi}\rangle$ will hence collapse the state into four possibilities: $|\bar{\psi}\rangle$, $X^{(i)}|\bar{\psi}\rangle$, $Y^{(i)}|\bar{\psi}\rangle$, or $Z^{(i)}|\bar{\psi}\rangle$. Correction can then be done as above.

As in the bit-flip example, the key to the success of this code is that when the measurements are performed, they give information about the error— and in the process, in fact, the measurements disturb the error so that it is discretized into a finite small set of possible errors— without disturbing the quantum state that we wish to protect.

## 2.4 A quick look at the theory of quantum error correction

Given a particular quantum code, what condition must an error satisfy in order to be correctable? A necessary and sufficient condition for the set $\mathcal{E}$ to be correctable is easily given by the following:

$$\langle\psi_i|E_b^\dagger E_a|\psi_j\rangle = C_{ab}\delta_{ij}, \tag{2.23}$$

for all $E_a, E_b \in \mathcal{E}$, where $|\psi_i\rangle, |\psi_j\rangle$ are orthogonal codewords of the code [50, 11], and $C_{ab}$ is an arbitrary matrix. The idea of this condition is that errors must take states to orthogonal states, and those orthogonal states should not depend on $i$; if they did, finding the error subspace would involve gaining information about the encoded state and thus disturbing it. A proof of (2.23) can be found in Refs. [71, 64].

Another term that is often used in connection with error correcting codes is the *distance*, which is related to the *weight* of errors. Given a Pauli error, that is, an error $E$ such that $E \in \{I, X, Y, Z\}^{\otimes n}$, the weight of $E$ is simply the number of non-identity terms in $E$. For example, the weight of the term $XXX$ is 3, while the weight of $IIX$ is 1. The distance of a code, then, is the smallest weight for which there exists an operator $E$ with that weight such that

$$\langle\psi_i|E|\psi_j\rangle \neq C_{ab}\delta_{ij}. \tag{2.24}$$

Often codes will be parametrized by the distance $d$ of the code as well as the number of encoded qubits $k$, and $n$, the number of physical qubits (or equivalently, the log of the dimension of the total Hilbert space); these numbers are often represented in the form $[[n, k, d]]$. The Shor code from section 2.3, for instance, is a $[[9, 1, 3]]$ code: it encodes one logical qubit in nine physical qubits. It is also not hard to see that the distance of this code is 3.

## 2.5 Stabilizer codes

An important class of quantum error correcting codes is that of the stabilizer codes, which were first introduced and analyzed by D. Gottesman [34]. A stabilizer code may be defined simply as follows: Consider a $2^n$-dimensional ($n$-qubit) Hilbert space and a subgroup of $2^{n-k}$ commuting Pauli operators $\mathcal{S} \in P_n$. This group of operators is the *stabilizer* of the

code; the codespace $\mathcal{C}(\mathcal{S})$ is the simultaneous $+1$ eigenspace of all the operators in $\mathcal{S}$. It can be shown [64] that if $-I$ is not an element of $\mathcal{S}$, the subspace stabilized is non-trivial, and the dimension of $\mathcal{C}(\mathcal{S})$ is $2^k$; hence, we regard this system as encoding $k$ qubits in $n$. The generators of such a group are a subset of this group such that any element of the stabilizer can be described as a product of generators. It is not hard to show that $n-k$ generators suffice to describe the stabilizer group $\mathcal{S}$.

This formalism lends itself to a nice way of describing what errors are correctable with a given stabilizer code. In order for a stabilizer code to correct for a set of Pauli operators $\mathcal{E}$, the condition (2.23) is satisfied if, for every $E_a, E_b \in \mathcal{E}$, either $E_b^\dagger E_a \in \mathcal{S}$ or $\{s_i, E_b^\dagger E_a\} = 0$ for some $s_i \in \mathcal{S}$. This is easily seen: given $\{s_i, E_b^\dagger E_a\} = 0$, since $s_k|\psi_i\rangle = |\psi_i\rangle$ for all $s_k \in \mathcal{S}$ and $|\psi_i\rangle \in \mathcal{C}(\mathcal{S})$,

$$\begin{aligned}
\langle\psi_i|E_b^\dagger E_a|\psi_j\rangle &= \langle\psi_i|s_i E_b^\dagger E_a|\psi_j\rangle \\
&= -\langle\psi_i|E_b^\dagger E_a s_i|\psi_j\rangle \\
&= -\langle\psi_i|E_b^\dagger E_a|\psi_j\rangle \\
&= 0. \qquad (2.25)
\end{aligned}$$

When considering universal quantum computation it is also useful to define the *normalizer* of a code. Given a stabilizer group $\mathcal{S}$, the normalizer $N(\mathcal{S})$ is the group of elements in $P_n$ that commute with all the the elements of $\mathcal{S}$, and it can be shown that the number of elements in $N(\mathcal{S})$ is $2^{n+k}$. Now, $n+k$ generators suffice to describe $N(\mathcal{S})$. Of these, $n-k$ can be chosen to be the generators of $\mathcal{S}$. It can be shown that the remaining $2k$ generators can be chosen to be the *encoded operators* $\bar{Z}_\mu, \bar{X}_\mu, \mu = 1, 2, ..., k$, where $\bar{Z}_\mu, \bar{X}_\mu$ denote the Pauli operators $X$ and $Z$ acting on encoded qubit $\mu$, tensored with the identity acting on all other encoded qubits. These encoded operators act, as their name implies, by taking states in $\mathcal{C}(\mathcal{S})$ to other states in $\mathcal{C}(\mathcal{S})$. Thus these encoded operators can also be thought of as *undetectable errors*: if one of these operators acts without our knowledge, it is impossible to correct, or even detect, the error with this code because the state stays within the codespace.

The usual protocol for stabilizer codes starts with measuring the stabilizer generators. This projection discretizes whatever error has occurred into one of $2^{n-k}$ error syndromes

labeled by the $2^{n-k}$ possible outcomes of the stabilizer generator measurements. The information given by the stabilizer measurements about what error syndrome has occurred is then used to apply a unitary recovery operator that returns the state to the codespace.

We can thus see that the stabilizer notation provides a compact way of describing codes: one must merely list the stabilizer generators of a code to specify the code subspace and the measurements to be performed in order to discretize and identify the errors. In addition, as we have seen, the stabilizer formalism gives a nice characterization of such features of the code as the encoded operators. As an example of the compactness of this description, let us consider the shortest possible code that encodes one qubit and corrects for one error [11, 55]. This code can be described simply with four stabilizer generators:

$$\begin{aligned} S_1 &= XZZXI \\ S_2 &= IXZZX \\ S_3 &= XIXZZ \\ S_4 &= ZXIXZ \end{aligned} \tag{2.26}$$

Writing out the code states in the computational basis, on the other hand, requires sixteen terms and does not give the rich array of other information able to be gleaned from the generators.

## 2.6 Topological quantum codes

Topological quantum codes are a special class of quantum error correcting codes that can be described by the stabilizer formalism. The basic idea of these codes is that information is encoded in the topological properties of the system; that is to say, undetectable errors correspond to homologically nontrivial chains of errors [45, 44, 16]. We will consider a code on a two-dimensional torus and then extend the formalism to a four-dimensional toric code.

### 2.6.1 Two-dimensional toric code

For this code, we assume that the (physical) qubits are arranged as the (one-dimensional) links on a square lattice, where the edges of the lattice are identified in order to form a

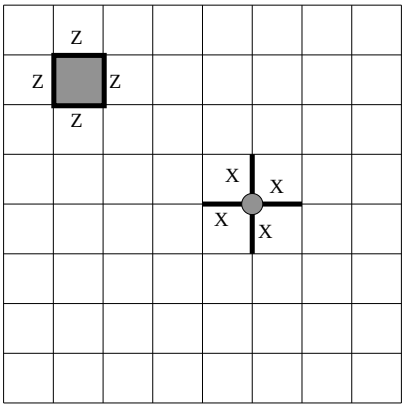



Figure 2.1: The stabilizer generators for the two-dimensional toric code.

torus. The stabilizer generators of this code are of two types. The first are associated with the two-dimensional plaquettes on the lattice. The stabilizer generator associated with the plaquette $P$ consists of the product of the four $Z$ operators acting on the qubits making up the boundary of $P$:

$$S(P) = \otimes_{q \in P} Z_q. \tag{2.27}$$

The second type is associated with the vertices of the lattice: the stabilizer associated with the vertex $v$ consists of the tensor products of the four $X$ operators acting on the links with $v$ as an endpoint:

$$S(v) = \otimes_{q \ni v} X_q. \tag{2.28}$$

These operators are shown in Fig. 2.1.

Let us consider the elements of the stabilizer that are tensor products of $Z$'s. It is not hard to see that those elements correspond to regions that can be tiled with plaquettes: that is, the $Z$'s act on the boundary of that region, which forms a homologically trivial cycle. On the other hand, a homologically nontrivial cycle—one that is not the boundary of anything—also commutes with all the stabilizer generators, but is not itself in the stabilizer. Therefore, it must be in the normalizer. Similar statements can be made for tensor products of $X$'s on the dual lattice. See Fig. 2.2.

Errors can be diagnosed by measuring the stabilizer generators; one of the strengths of

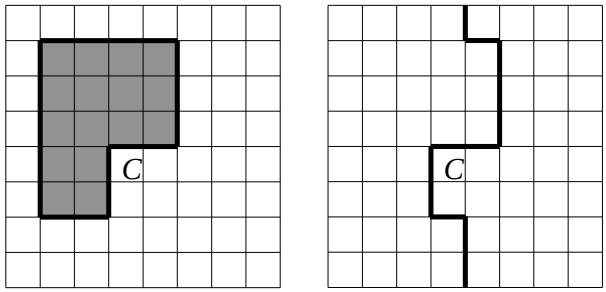



Figure 2.2: (a) shows a homologically trivial cycle (correctable error) while (b) shows a homologically nontrivial cycle.

this scheme is that the stabilizer generators are *local* on a torus. The syndrome is given by the plaquette(vertex) sites on the lattice(dual lattice) where the syndrome measurement is $-1$; these can be thought of as particles, or defects, on the lattice. If a chain of errors (a set of links on the lattice or dual lattice) occurs, the defects created arise at the endpoints, or boundary, of the chain on the lattice or dual lattice. This characterization of the defects means that an error syndrome does not at all uniquely describe an error chain; indeed, there are many error chains that could give rise to a particular error syndrome. However, the beauty of this formulation is that it is not necessary to know the exact placement of the error chain; any correction chain that connects the defects will remove the syndrome. Furthermore, it will do so without ill effect to the encoded information as long as the correction is homologically correct: that is, differs from the actual error chain by a homologically trivial surface. In that case the total operator acting on the system after the correction is simply an element of the stabilizer, and the encoded state is not disturbed. On the other hand, if the correction differs from the actual error chain by a homologically nontrivial surface, an undetectable error is introduced. See Fig. 2.3.

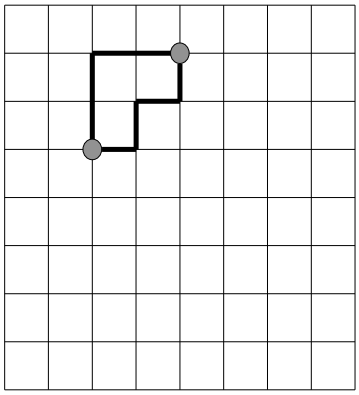


Figure 2.3: One chain connecting the syndrome defects is the error chain, and the other is the correction. Even though they are different, the total operator is a homologically trivial surface in the stabilizer and hence has no effect on the state.

### 2.6.2 Local correction of toric codes

Correcting a chain of errors in the above picture therefore seems naively to involve processing the error syndrome to compute the most likely chain that might have resulted in such a syndrome and then correcting according to that chain. This procedure, while local as regards the quantum processing, involves nonlocal classical processing of the error syndrome. How can this procedure be made into one that is local even as regards the classical processing?

This two-dimensional quantum system is analogous to a one-dimensional classical system in which the defects are again the point boundaries of a one-dimensional error chain. For a long time it was believed that there was no set of uniform local rules that could correct these errors in a way such that an encoded bit was protected over a long period of time. Not until very recently did Peter Gacs [27, 28] construct a counterexample to this belief. This counterexample is quite complex and involves a complicated and elegant hierarichal structure that is able to perform a sort of self-simulation. One approach to local error correction is therefore to use the ideas of self-similarity and simulation that are used in [28]. This approach was taken by J. Harrington and myself; details are to be found in [40].

In this work, we will take a different approach. A fundamental principle of statistical

physics is that systems with a greater number of spatial dimensions are better able to resist the disordering effects of fluctuations [16]. Thus in Chapter 7, we will introduce extra dimensions and consider a system where spins reside on two-dimensional plaquettes in a four-dimensional lattice. Given some edge, one type of stabilizer generator is the tensor product of $X$'s on the spins residing on all six plaquettes which share that edge. Another type is associated with cubes: given a cube, the stabilizer generator is the tensor product of $Z$'s on the six spins that are on the faces of that cube. Phase and flip errors are again dual. The defects here are given by closed one-dimensional loops in both the lattice and the dual lattice, and correctable errors are characterized by two-dimensional error "droplets" of plaquettes with a one-dimensional boundary. We will see that it is then possible to construct an anisotropic rule that locally corrects for these droplets. Because this rule behaves in a nice way, it will be instrumental in the proofs on the cost of quantum fault tolerance given in Chapter 7.

# Chapter 3

# Quantum feedback control

## 3.1 The master equation

Continuous quantum feedback can be defined, for the present purposes, as the process of monitoring a quantum system and using the continuous (in time) measurement record to control its dynamics. It can be analyzed by considering the dynamics of the measured system conditioned on the continuous measurement record; this process is referred to as *unraveling*. The reduced dynamics of a system subject to weak continuous measurement is described by a Markov master equation, which determines the dynamics of the system averaged over all possible measurement records. However, if the time-continuous measurement record (a classical stochastic process) is known, then it is possible to describe the conditional state of the measured system by a stochastic conditional evolution equation. A given master equation does not uniquely determine the conditional evolution equation, as there are many ways in which information about the system may be collected from the environment to which it is coupled as a result of the measurement. That is to say, a given master equation admits many unravelings.

In this section we will introduce some of the results of this formalism; for more details see [71, 96]. It is common to perform measurements in quantum systems by entangling the system (*e.g.*, an atom) with its ancillary environment (*e.g.*, electromagnetic field modes), and then observing the ancilla. It is possible to show [71] that this kind of indirect measurement can be described via Kraus operators [54] $\{M_r\}$, where $r$ indexes the result of the measurement. These operators are required to satisfy the Kraus normalization condi-

tion $\sum_r M_r^\dagger M_r = 1$. Given a pre-measurement state $|\phi\rangle$, the state of the system after the measurement, conditioned on the measurement result $r$, is given by

$$\frac{M_r|\phi\rangle}{\sqrt{\langle\phi|M_r^\dagger M_r|\phi\rangle}} \text{ with probability } \langle\phi|M_r^\dagger M_r|\phi\rangle. \tag{3.1}$$

Normalization of Kraus operators implies that the operators $F_r \equiv M_r^\dagger M_r$ constitute a positive operator-valued measure (POVM).

In terms of a possible mixed state given by the density matrix $\rho$, this measurement leads to the conditioned state

$$\rho_r = \frac{M_r \rho M_r^\dagger}{\langle M_r^\dagger M_r\rangle}. \tag{3.2}$$

Had the result of the measurement not been known, the density matrix would have become the unconditioned

$$\rho_{final} = \sum_r M_r \rho M_r^\dagger. \tag{3.3}$$

Let us now consider a system in which we assume that these measurements are taking place continuously, in an infinitesimal time interval $dt$. If we make the Markovian approximation that $\rho(t+dt)$ is completely determined by $\rho(t)$, which is a reasonable assumption in many systems of interest, we can write down a first-order differential equation that describes the evolution of the density matrix. The form of the Kraus operators dictating the evolution of the state is dictated by the Markovian approximation. For example, if we assume only two Kraus operators, then the change in the state of the system over a time interval $dt$ due to its interaction with the environment can be described by a single Kraus (jump) operator [54] $\Omega_1 = c\sqrt{dt}$, so that the "jumps" occur with probability $\langle c^\dagger c\rangle dt$. Normalization requires another Kraus operator, $\Omega_0 = 1 - c^\dagger c dt/2 - iHdt$, where $H$ is Hermitian. Then the unconditional master equation without feedback is just the familiar Lindblad form [13]

$$\begin{aligned} d\rho &= \Omega_0\rho\Omega_0 + \Omega_1\rho\Omega_1 - \rho \\ &= -i[H,\rho]dt + c\rho c^\dagger dt - \frac{1}{2}(c^\dagger c\rho + \rho c^\dagger c)dt \\ &\equiv -i[H,\rho]dt + \mathcal{D}[c]\rho dt. \end{aligned} \tag{3.4}$$

A bosonic example is given in [26], while a fermionic example is given in [61].

## 3.2 Markovian feedback

### 3.2.1 Jump unravelings

One way to unravel this master equation is to assume that the environment is measured so that the time of each jump event is determined. If the measured number of jumps up to time $t$ is denoted $N(t)$, then the increment $dN(t)$ is defined formally by

$$dN_c(t)^2 = dN_c(t) \tag{3.5}$$

$$E[dN_c(t)] = \langle c^\dagger c\rangle_c dt; \tag{3.6}$$

$dN_c(t)$ can be 0 or 1 depending on whether a jump has occurred or not. Here $E[\ ]$ defines a classical ensemble average, and the subscript $c$ on the quantum average reminds us that the rate of the process at time $t$ depends on the conditional state of the quantum system up to that time. That is to say, it depends on the state of the quantum system conditioned on the entire previous history of the measurement record given by $dN/dt$. This conditional state is determined by a stochastic Schrödinger equation (SSE)

$$\begin{aligned} d|\psi_c(t)\rangle &= dN_c \left(\frac{\Omega_1}{\sqrt{\Omega_1^\dagger\Omega_1}} - 1\right)|psi_c(t)\rangle + (1 - dN_c(t))\left(\frac{\Omega_0}{\sqrt{\Omega_0^\dagger\Omega_0}} - 1\right)|\psi_c(t)\rangle \\ &= \left[dN_c(t)\left(\frac{c}{\sqrt{\langle c^\dagger c\rangle_c(t)}} - 1\right) + dt \right. \\ &\quad \left. \times \left(\frac{\langle c^\dagger c\rangle_c(t)}{2} - \frac{c^\dagger c}{2} - iH\right)\right]|\psi_c(t)\rangle. \end{aligned} \tag{3.7}$$

We will refer to this as a *jump unraveling*. If we average over the measurement record to form $\rho(t) = E[|\psi_c(t)\rangle\langle\psi_c(t)|]$, it is easy to show using Eqs. (3.5) and (3.6) that $\rho(t)$ obeys the unconditional master equation given in Eq. (3.4).

Now consider Markovian Hamiltonian feedback, linear in the current:

$$H_{fb}(t) = \frac{dN(t)}{dt}V, \tag{3.8}$$

with $V$ an Hermitian operator. Taking into account that the feedback must act after the measurement, it can be shown [93] that the feedback modifies the conditional evolution by

changing the $c$ in the numerator of the first term into $e^{-iV}c$. Since likewise changing all of the other occurrences of $c$ has no effect, the ensemble average behaviour is the same as before, with $c$ changed to $e^{-iV}c$. That is to say, the feedback-modified master equation is

$$\dot{\rho} = -i[H, \rho] + \mathcal{D}[e^{-iV}c]\rho. \tag{3.9}$$

### 3.2.2 Diffusive unravelings

A very different unraveling may be defined by first noting that given some complex number $\gamma = |\gamma|e^{i\phi}$, we may make the transformation

$$\begin{aligned} c &\rightarrow c + \gamma \\ H &\rightarrow H - \frac{i|\gamma|}{2}(e^{-i\phi}c - e^{i\phi}c^\dagger) \end{aligned} \tag{3.10}$$

and obtain the same master equation. In the limit as $|\gamma|$ becomes very large, the rate of the Poisson process is dominated by the term $|\gamma|^2$. In this case it may become impossible to monitor every jump process, and a better strategy is to approximate the Poisson stochastic process by a Gaussian white-noise process.

For large $\gamma$, we can consider the system for a time $\delta t$ in which the system changes negligibly but the number of detections $\delta N(t) \approx |\gamma|^2 \delta t$ is very large; then we can approximate $\delta N(t)$ as [98]

$$\delta N(t) \approx |\gamma|^2 \delta t + |\gamma| \langle e^{-i\phi}c + c^\dagger e^{i\phi} \rangle_c \, \delta t + |\gamma| \delta W(t), \tag{3.11}$$

where $\delta W(t)$ is normally distributed with mean zero and variance $\delta t$.

We now define the stochastic measurement record as the current

$$\frac{dQ(t)}{dt} = \lim_{\gamma \to \infty} \frac{\delta N(t) - |\gamma|^2 \delta t}{|\gamma| \delta t} \tag{3.12}$$

$$= \langle e^{-i\phi}c + e^{i\phi}c^\dagger \rangle_c + dW(t)/dt. \tag{3.13}$$

Given this stochastic measurement record, we can determine the conditional state of the quantum system by a stochastic Schrödinger equation analogous to Eq. (3.7). The equivalence (in the ensemble average) to the master equation (3.4) is, in this case, easier to see

by considering $\rho_c = |\psi_c\rangle\langle\psi_c|$, which obeys the stochastic master equation (SME) [1]

$$\begin{aligned} d\rho_c(t) &= -i\,[H,\,\rho_c(t)]\,dt + \mathcal{D}[e^{-i\phi}c]\rho_c(t)dt \\ &\quad + \mathcal{H}[e^{-i\phi}c]\rho_c(t)dW(t). \end{aligned} \tag{3.14}$$

In the above equations, the expectation $\langle a\rangle_c$ denotes $\mathrm{tr}\,(\rho_c a)$, $dW$ is a normally distributed infinitesimal random variable with mean zero and variance $dt$ (a *Wiener increment* [29]), and $\mathcal{H}$ is a superoperator that takes a jump operator as an argument and acts on density matrices as

$$\mathcal{H}[c]\rho = c\rho + \rho c^\dagger - \rho\,\mathrm{tr}\,[c\rho + \rho c^\dagger]. \tag{3.15}$$

We thus have a different unraveling of the original master equation Eq. (3.4). Because of the white noise in the stochastic master equation (3.14) we call this a diffusive unraveling. It applies, for example, when one performs a continuous weak homodyne measurement of a field $c$ by first mixing it with a classical local oscillator in a beamsplitter and then measuring the output beams with photodetectors [98]. In that case the measurement process $dQ(t)$ determines the observed photocurent. Another measurement model in which it may be appropriate to approximate a Poisson measurement process by a white-noise measurement process is the electronic point contact model for monitoring a single quantum dot [33, 32]. In that case the form of the master equation itself determines a large background jump rate, rather than an imposed classical field prior to detection.

We now consider Markovian feedback of the white-noise measurement record via a Hamiltonian, where the strength of the feedback is a linear function of the measurement current:

$$H_{fb}(t) = \frac{dQ(t)}{dt}F, \tag{3.16}$$

where $F$ is a Hermitian operator. It can be shown that the addition of such feedback leads

[1] For an alternate and elegant interpretation of the SME, in which the SME is considered as a quantum filtering process that changes our state of *knowledge* of the system, see [84].

to the conditioned master equation [93, 99]

$$\begin{aligned}\dot{\rho} &= -i[(e^{i\phi}c^\dagger F + e^{-i\phi}Fc)/2 + H, \rho] \\ &\quad +\mathcal{D}[e^{-i\phi}c - iF]\rho \\ &\quad +dW(t)\mathcal{H}[e^{-i\phi}c - iF]\rho. \end{aligned} \tag{3.17}$$

In order to derive analytic results given such feedback, it is convenient to consider the average over many such evolution trajectories. Since the expectation value of $dW$ is zero, averaging yields an unconditioned master equation

$$\begin{aligned}\dot{\rho} &= -i[(e^{i\phi}c^\dagger F + e^{-i\phi}Fc)/2 + H, \rho] \\ &\quad +\mathcal{D}[e^{-i\phi}c - iF]\rho \end{aligned} \tag{3.18}$$

The above equations are only valid for perfect (unit-efficiency) detection; the correspondences between error correction and feedback are more readily seen in this case, and we discuss the case of imperfect detection more in subsequent chapters.

These feedback equations are easily generalized in the following way: Given $n$ qubits, denote a set of measurement operators by $\{c_1, c_2, \cdots, c_n\}$, where $c_j$ acts on the $j$th qubit, and a set of feedback operators by $\{F_1, \cdots, F_n\}$, where the action of $F_j$ is conditioned on the measurement of the $j$th qubit. Then the unconditional master equation (3.18), for example, generalizes to

$$\begin{aligned}\dot{\rho} = \sum_{j=1}^{n} \quad & \{-i[(e^{i\phi_j}c_j^\dagger F_j + e^{-i\phi_j}F_j c_j)/2 + H, \rho] \\ & +\mathcal{D}[e^{i\phi_j}c_j - iF_j]\rho\}. \end{aligned} \tag{3.19}$$

## 3.3 State-estimate feedback

The second, and more general, way to add feedback is to modulate the Hamiltonian by a functional of the entire measurement record. An important class of this kind of feedback is *state estimate feedback* [22], also called Bayesian feedback, in which feedback is a function of the current conditioned state estimate $\rho_c$. This kind of feedback is of especial interest

because of the quantum Bellman theorem [21], which proves that the optimal feedback strategy will be a function only of conditioned state expectation values for a large class of physically reasonable cost functions. An example of such an estimate feedback control law is to add the Hamiltonian $\langle I_Q(t)\rangle_c F = \langle c + c^\dagger\rangle_c F$, which depends on what we *expect* the current $I_Q(t)$ should be given the previous measurement history rather than its actual instantaneous value. Adding this feedback to the SME (3.14) leads to the dynamics

$$\begin{aligned} d\rho_c(t) &= -i\,[H,\,\rho_c(t)]\,dt \\ &\quad +\mathcal{D}[c]\rho_c(t)dt + \mathcal{H}[c]\rho_c(t)dW(t) \\ &\quad -i\langle I_Q\rangle_c\,[F,\rho_c(t)]\,dt \qquad (3.20) \\ dQ(t) &= \langle c + c^\dagger\rangle_c\,dt + dW(t). \qquad (3.21) \end{aligned}$$

This feedback has further advantages over Markovian feedback: as we shall see, because state-estimation feedback can depend on the entire measurement record rather than on instantaneous values of the current, estimate feedback protocols are rather more forgiving of measurement error.

These benefits come at a price, however. To perform the state estimation itself may take a nontrivial amount of post-processing: we will quantify this computational cost for the quantum error correction protocol given in the next chapter. Furthermore, because state estimate feedback is essentially non-Markovian in nature, analyzing state estimation schemes analytically is difficult. Therefore, the state-estimation protocols given in Chapters 4 and 5 will be analyzed via numerical simulations.

# Chapter 4

# Continuous quantum error correction via estimate feedback control

This chapter presents a method for continuously protecting an unknown quantum state using weak measurement, state estimation, and Hamiltonian correction. First a very simple one-qubit example will be presented. Building on the intuition gained from that model, the state estimation procedure will be detailed for the example of the bit-flip code, and finally, a general stabilizer code will be considered. Results obtained from simulating the bit-flip version of the model will then be presented, including some further results on relaxing the assumptions of perfect responsiveness and perfect detection efficiency.

## 4.1 Continuous quantum error correction protocol

### 4.1.1 One-qubit picture

Before showing how the procedure works for the bit-flip code, we can gain some intuition about how it works by considering an even simpler "code": the spin-up state (*i.e.*, $|0\rangle$) of a single qubit. The stabilizer is $M_0 = Z$, and will be weakly measured with strength $\kappa$. This one-dimensional code protects against bit flips $X$, which we will assume happen with some probability $\sim \gamma dt$. To corect for these flips, a correction Hamiltonian proportional to $X$ is applied with control strength $\lambda$. The resulting stochastic master equation can be rewritten as a set of Bloch sphere equations as follows:

$$d\langle X\rangle_c = -2\kappa\langle X\rangle_c\,dt - 2\sqrt{\kappa}\langle X\rangle_c\langle Z\rangle_c\,dW \tag{4.1}$$

$$d\langle Y\rangle_c = -2\gamma\langle Y\rangle_c\,dt - 2\kappa\langle Y\rangle_c - 2\sqrt{\kappa}\langle Y\rangle_c\langle Z\rangle_c\,dW - 2\lambda\langle Z\rangle_c\,dt \tag{4.2}$$

$$d\langle Z\rangle_c = -2\gamma\langle Z\rangle_c\,dt + 2\sqrt{\kappa}(1-\langle Z\rangle_c^2)dW + 2\lambda\langle Y\rangle_c\,dt. \tag{4.3}$$

The Bloch vector representation $(\langle X\rangle, \langle Y\rangle, \langle Z\rangle)$ [71] of the qubit provides a simple geometric picture of how it evolves. Decoherence (the $\gamma$ term) shrinks the Bloch vector, measurement (the $\kappa$ terms) lengthens the Bloch vector and moves it closer to the $z$-axis, and correction (the $\lambda$ term) rotates the Bloch vector in the $y$–$z$ plane. Fig. 4.1 depicts the desired evolution: depending on whether the Bloch vector is in the hemisphere with $\langle Y\rangle > 0$ or $\langle Y\rangle < 0$, the feedback should rotate the vector as quickly as possible in such a way that it is always moving towards the codespace (spin-up state). Therefore, if the maximum feedback strength possible is $\lambda_{max}$, the optimal feedback is given by

$$\lambda = \lambda_{max}\mathrm{sgn}\langle Y\rangle. \tag{4.4}$$

Note that if the Bloch vector lies exactly on the $z$-axis with $\langle Z\rangle < 0$, rotating it either way will move it towards the spin-up state—the two directions are equivalent, and it suffices to choose one of them arbitrarily.

### 4.1.2 Bit-flip code model

Suppose $\rho$ is subjected to bit-flipping decoherence as in (2.10); to protect against such decoherence, we have seen that we can encode $\rho$ using the bit-flip code (2.2–2.3). In this section, a similar protocol is defined that operates continuously and uses only weak measurements and slow corrections.

The first part of the protocol is to weakly measure the stabilizer generators $ZZI$ and $IZZ$ for the bit-flip code, even though these measurements will not completely collapse the errors. To localize the errors even further, we also measure the remaining nontrivial stabilizer operator $ZIZ$. The second part of our protocol is to apply the slow Hamiltonian corrections $XII$, $IXI$, and $IIX$ corresponding to the unitary corrections $XII$, $IXI$, and $IIX$, with

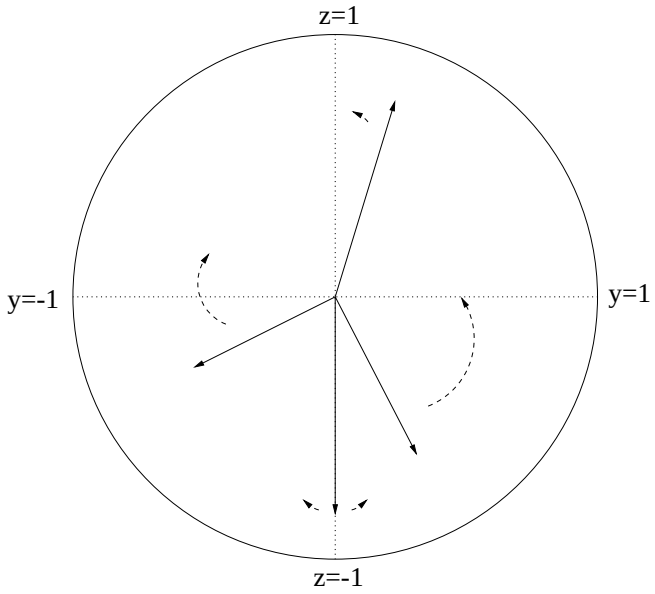



Figure 4.1: Bloch sphere showing the action of our feedback scheme on one qubit. Wherever the Bloch vector is in the $y$–$z$ plane, the feedback forces it back to the spin-up state, which is the codespace of this system. All the vectors shown lie, without loss of generality, in the $x = 0$ plane.

control parameters $\lambda_k$ that are to be determined. If we parameterize the measurement strength by $\kappa$ and perform the measurements using the unravelling (3.13–3.14), the SME describing our protocol is

$$\begin{aligned}
d\rho_c &= \gamma(\mathcal{D}[XII] + \mathcal{D}[IXI] + \mathcal{D}[IIX])\rho_c dt \\
&\quad +\kappa(\mathcal{D}[ZZI] + \mathcal{D}[IZZ] + \mathcal{D}[ZIZ])\rho_c dt \\
&\quad +\sqrt{\kappa}(\mathcal{H}[ZZI]dW_1 + \mathcal{H}[IZZ]dW_2 \\
&\qquad + \mathcal{H}[ZIZ]dW_3)\rho_c \\
&\quad -i[F, \rho_c]dt \qquad (4.5) \\
dQ_1 &= 2\kappa\langle ZZI\rangle_c dt + \sqrt{\kappa}dW_1 \qquad (4.6) \\
dQ_2 &= 2\kappa\langle IZZ\rangle_c dt + \sqrt{\kappa}dW_2 \qquad (4.7) \\
dQ_3 &= 2\kappa\langle ZIZ\rangle_c dt + \sqrt{\kappa}dW_3, \qquad (4.8)
\end{aligned}$$

where

$$F = \lambda_1 XII + \lambda_2 IXI + \lambda_3 IIX \qquad (4.9)$$

is the feedback Hamiltonian having control parameters $\lambda_k$.

Following the logic of quantum error correction, it is natural to choose the $\lambda_k$ to be functions of the error syndrome. For example, the choice

$$\begin{aligned}
\lambda_1 &= \lambda(\frac{1-\langle ZZI\rangle_c}{2})(\frac{1+\langle IZZ\rangle_c}{2})(\frac{1-\langle ZIZ\rangle_c}{2}) \\
\lambda_2 &= \lambda(\frac{1-\langle ZZI\rangle_c}{2})(\frac{1-\langle IZZ\rangle_c}{2})(\frac{1+\langle ZIZ\rangle_c}{2}) \\
\lambda_3 &= \lambda(\frac{1+\langle ZZI\rangle_c}{2})(\frac{1-\langle IZZ\rangle_c}{2})(\frac{1-\langle ZIZ\rangle_c}{2}), \qquad (4.10)
\end{aligned}$$

where $\lambda$ is the maximum feedback strength that can be applied, is reasonable: it acts trivially when the state is in the codespace and applies a maximal correction when the state is orthogonal to the codespace. Unfortunately this feedback is sometimes harmful when it need not be. For example, when the controller receives no measurement inputs (*i.e.*, $\kappa = 0$), it still adds an extra coherent evolution which, on average, will drive the state of the system away from the state we wish to protect.

This weakness of the feedback strategy suggests that we should choose our feedback more carefully. To do this, we introduce a cost function describing how far away our state is from its target and choose a control which minimizes this cost. The difficulty is that our target is an *unknown* quantum state. However, we can choose the target to be the codespace, which we do know. We choose our cost function, therefore, to be the norm of the component of the state outside the codespace. Since the codespace projector is $\Pi_{\mathcal{C}} = \frac{1}{4}(III + ZZI + ZIZ + IZZ)$, the cost function is $1 - f$, where $f(\rho) = \mathrm{tr}(\rho\Pi_{\mathcal{C}})$. Under the SME (4.5), the time evolution of $f$ due to the feedback Hamiltonian $F$ is

$$\begin{aligned} \dot{f}_{fb} &= 2\lambda_1\langle YZI + YIZ\rangle_c \\ &\quad +2\lambda_2\langle ZYI + IYZ\rangle_c \\ &\quad +2\lambda_3\langle ZIY + IZY\rangle_c . \end{aligned} \tag{4.11}$$

Maximizing $\dot{f}_{fb}$ minimizes the cost, yielding the optimal feedback coefficients

$$\begin{aligned} \lambda_1 &= \lambda\,\mathrm{sgn}\langle YZI + YIZ\rangle_c \\ \lambda_2 &= \lambda\,\mathrm{sgn}\langle ZYI + IYZ\rangle_c \\ \lambda_3 &= \lambda\,\mathrm{sgn}\langle ZIY + IZY\rangle_c , \end{aligned} \tag{4.12}$$

where, again, $\lambda$ is the maximum feedback strength that can be applied. This feedback scheme is a *bang-bang* control scheme, meaning that the control parameters $\lambda_k$ are always at the maximum or minimum value possible ($\lambda$ or $-\lambda$, respectively), which is a typical control solution both classically [101] and quantum mechanically [86].

### 4.1.3 Feedback for a general code

Our approach generalizes for a full $[[n, k, d]]$ quantum error correcting code, which can protect against depolarizing noise [71] acting on each qubit independently. This noise channel, unlike the bit-flip channel, generates a full range of quantum errors—it applies either $X$, $Y$, or $Z$ to each qubit equiprobably at a rate $\gamma$. The $n-k$ stabilizer generators $\{M_l\}$ are weakly measured with strength $\kappa$. For each syndrome $m$, we apply a slow Hamiltonian correction $F_m$ with control strength $\lambda_m$, the weight of each correction being $d$ or less. The

SME describing this process is

$$\begin{aligned} d\rho_c &= \gamma \sum_{j=x,y,z} \sum_{i=1}^{n} (\mathcal{D}[\sigma_j^{(i)}])\rho_c dt + \kappa \sum_{l=1}^{n-k} \mathcal{D}[M_l]\rho_c dt \\ &\quad + \sqrt{\kappa} \sum_{l=1}^{n-k} \mathcal{H}[M_l] dW_j \rho_c - i \sum_{r=1}^{R} \lambda_r [F_r, \rho_c] dt. \end{aligned} \tag{4.13}$$

The number of feedback terms $R$ needed will be less than or equal to the number of errors the code corrects against. The reason that this equality is not strict is that quantum error correcting codes can be *degenerate*, meaning that there can exist inequivalent errors that have the same effect on the state—a purely quantum mechanical property [34].

The $\lambda_r$ may be optimized relative to a cost function equal to the state's overlap with the codespace. For a general stabilizer code $\mathcal{C}$, the codespace projector is

$$\Pi_{\mathcal{C}} = \frac{1}{2^{n-k}} \prod_{l=1}^{n-k} (I + M_l)$$

and the rate of change of the codespace overlap due to feedback is

$$\dot{f}_{fb} = -i \operatorname{tr} \sum_{r=0}^{n-k} \lambda_r [\Pi_{\mathcal{C}}, F_r] \rho.$$

Maximizing this overlap subject to a maximum feedback strength $\lambda$ yields the feedback coefficients

$$\lambda_r = \lambda \operatorname{sgn} \langle [\Pi_{\mathcal{C}}, F_r] \rangle_c . \tag{4.14}$$

## 4.2 Feedback based on the completely mixed state

The control solutions (4.12) and (4.14) require the controller to integrate the SME (4.5) using the measurement currents $Q_i(t)$ and the initial condition $\rho_c$. However, typically the initial state $\rho_c(0)$ will be unknown. Fortunately the calculation of the feedback (4.12) does not depend on where the initial condition is within the codespace, so the controller may assume the maximally mixed initial condition $\rho_e = \frac{1}{2}(|\bar{0}\rangle\langle\bar{0}| + |\bar{1}\rangle\langle\bar{1}|)$ for its calculations. This section will prove that this property generalizes for a wide class of stabilizer codes; simulations show that this property does not hold for all possible stabilizer codes, but it

does hold for most codes of interest.

Even though the quantum error correction feedback control scheme described in Section 4.1.2 does not distinguish between codewords, it is not obvious that the initial codeword can remain unknown when integrating its SME and calculating the relevant expectation values. Since the goal is to protect unknown quantum states, this property is crucial to this scheme's success. Fortunately, for a large class of stabilizer codes, the computation of the feedback can be done by assuming the initial state is the completely mixed codespace state $\rho_e = \frac{1}{2^n}\prod_{l=1}^{n-k}(I + M_l)$, which I prove here.

The first step is to define the set $G$ for the $[[n, k, d]]$ code $\mathcal{C}$ with stabilizer $S(\mathcal{C})$ as

$$G = \{\alpha s \,|\, \alpha \in P_n, s \in S(\mathcal{C}), [s, \alpha] = 0 \text{ iff } |\alpha| \text{ is even}\}, \tag{4.15}$$

where $|\alpha|$ denotes the weight of $\alpha$ as defined in Chapter 2.

The conditions required for the computation of the feedback to be insensitive to the initial codeword can be rewritten in terms of the *Pauli basis coefficients* $R_g(\rho)$ which are defined as follows. Let $g = \sigma_{i_1} \otimes ... \otimes \sigma_{i_n}$, where $i_1 \ldots i_n$ take on the values $x, y, z, I$ and $\sigma_I = I$. Then

$$R_g(\rho) \equiv \mathrm{tr}(\rho g)/2^n = \langle g \rangle / 2^n. \tag{4.16}$$

The problem can then be formulated in terms of proving conditions on $G$ as follows:

1. For every $R_g$ used in this feedback scheme, $g \in G$.

2. For every $g \in G$ and every $\rho_1$ and $\rho_2$ in $\mathcal{C}$, $R_g(\rho_1) = R_g(\rho_2)$.

3. Evolution under the SME couples members of the set $\{R_g | g \in G\}$ only to each other.

**Theorem** *Let $\mathcal{C}$ be an $[[n, 1, 3]]$* [1] *stabilizer code whose stabilizer $S(\mathcal{C})$ has generators of only even weight and whose encoded operations set $N(S) \setminus S$ has elements of only odd*

[1] The restriction to $[[n, 1, 3]]$ codes is for simplicity of analysis; the proof may be extended to larger codes. Note that for an $[[n, 1, 3]]$ code, the $F_l$ in the master equation (4.13) are all of the form $\sigma_j^{(k)}$, where this notation denotes the weight-one Pauli operator $\sigma_j$ acting on qubit $k$.

*weight.*[2] *Then the conditions 1–3 above are satisfied; consequently, this scheme does not require knowledge of where the initial codeword lies in* $\mathcal{C}$.

**Proof:**

In this proof, any variable of the form $\alpha_a$ is an arbitrary element of $P_n$, and any variable of the form $s_a$ is an arbitrary element of $S(\mathcal{C})$. Each of the conditions listed above is proved separately.

*Condition 1:* By construction, $G$ contains all $M$ of the form $M = s_i\sigma_j^{(k)}$, where $[s_i, \sigma_j^{(k)}] \neq 0$. These are precisely the operators used to compute the feedback in (4.14) for a code encoding one qubit.

*Condition 2:* Let $g = \alpha s \in G$ and let $\rho \in \mathcal{C}$. Either $\alpha \in S$, $\alpha \in N(S) \setminus S$, or $\alpha \notin N(S)$. Suppose $\alpha \in S$. Then $g \in S$ acts trivially on all states in the codespace, so $R_g = 1/2^n \mathrm{tr}(\rho g) = 1/2^n$ for this case. Now suppose $\alpha \in N(S) \setminus S$. Then $[\alpha, s] = 0$, and since $\alpha s \in G$, $|\alpha|$ is even. But every element of $N(S) \setminus S$ has odd weight by hypothesis, which is a contradiction. Hence $\alpha$ cannot be in $N(S) \setminus S$. Finally, suppose $\alpha \notin N(S)$. Then there exists some $s' \in S$ such that $[\alpha, s'] \neq 0$; let $s'$ be such an element. Then for $|\psi\rangle, |\phi\rangle \in \mathcal{C}$,

$$\begin{aligned} \langle\psi|\alpha|\phi\rangle &= \langle\psi|\alpha s'|\phi\rangle = -\langle\psi|s'\alpha|\phi\rangle \\ &= -\langle\psi|\alpha|\phi\rangle = 0. \end{aligned} \tag{4.17}$$

Hence for this case $R_g = 1/2^n \mathrm{tr}(\rho\alpha s) = 0$. Note that these expressions for $R_g$ must be the same no matter where $\rho$ is in the codespace; therefore, for every $g \in G$ and $\rho_1, \rho_2 \in \mathcal{C}$, $R_g(\rho_1) = R_g(\rho_2)$.

*Condition 3:* This is proved by considering $dR_M$, where $M \in G$: it will be shown that $dR_M = f(\{R_N | N \in G\})$ for some real function $f$. Now, for any $M \in P_n$, $dR_M = \mathrm{tr}(d\rho\, M)$, where $d\rho$ is given by the master equation (4.13), and condition 3 can be shown for each term of the master equation separately. First, substituting in the master equation shows

[2] It is possible that this restriction may be able to be relaxed; however, it is sufficiently general that it holds for the most well-known codes, including the bit-flip code, the five-bit code, the Steane code, and the nine-bit Shor code. This condition also ensures that the definition of $G$ is consistent, *i.e.*, if $\alpha_j s_k \in G$ and $\alpha_j = \alpha_n s_m$, then $\alpha_n$ and $s_m s_k$ also fulfill the conditions for $\alpha_n(s_m s_k)$ to be in $G$.

that any term of the form $\mathcal{D}[c]\rho dt$ contributes either 0 or the simple exponential damping term $-2R_M$ to $dR_M$ if $M$ and $c$ commute or anticommute, respectively.

As for the master equation term $\mathcal{H}[s_j]dW_j\rho$, by writing the master equation in the Pauli basis it is possible to see that $R_N$ contributes to $dR_M$ through this term precisely when $Ns_j = M$ and $\{s_j, N\} \neq 0$. Since $M \in G$, it is possible to write $M = \alpha_k s_l$ (with the appropriate restriction on $[\alpha_k, s_l]$ depending on the weight of $\alpha_k$) . $N = \alpha_k s_l s_j = \alpha_k s_m$, so the condition above that $[s_j, N] = 0$ becomes $[s_j, \alpha_k s_l s_j] = (\alpha_k[s_j, s_l s_j] + [s_j, \alpha_k]s_l s_j)$ $\Rightarrow [s_j, \alpha_k] = 0$. Therefore, $[\alpha_k, s_m] = s_l[\alpha_k, s_j] + [\alpha_k, s_l]s_j = [\alpha_k, s_l]s_j$ which is zero or not depending on the original weight of $\alpha_k$. So if $M = \alpha_k s_l$ is such that $M \in G$, $N = \alpha_k s_m$ must fulfill that same condition, implying that $N \in G$ also.

Similarly, $R_N$ contributes to $dR_M$ through the master equation term $[\sigma_j^{(k)}, \rho]$ when $N\sigma_j^{(k)} = M$ and $[\sigma_j^{(k)}, N] \neq 0$. Now, $M \in G$ so $M = \alpha_l s_m$, again with the appropriate restriction on $[\alpha_l, s_m]$ depending on the weight of $\alpha_l$. Then $N = \sigma_j^{(k)}\alpha_l s_m \equiv \alpha_n s_m$, so the condition above that $\{\sigma_j^{(k)}, N\} \neq 0$ becomes

$$\begin{aligned} \{\sigma_j^{(k)}, \sigma_j^{(k)}\alpha_l s_m\} &= \sigma_j^{(k)}[\sigma_j^{(k)}, \alpha_l]s_m + \sigma_j^{(k)}\alpha_l\{\sigma_j^{(k)}, s_m\} \\ &= \sigma_j^{(k)}\{\sigma_j^{(k)}, \alpha_l\}s_m - \sigma_j^{(k)}\alpha_l[\sigma_j^{(k)}, s_m] \\ &= 0. \end{aligned} \tag{4.18}$$

The analysis of this term can now be divided into two cases. Case 1 occurs when $\sigma_j^{(k)}\alpha_l$ has weight $|\alpha_l|$, implying that $\{\alpha_l, \sigma_j^{(k)}\} = 0$. Then $\{\sigma_j^{(k)}, \sigma_j^{(k)}\alpha_l s_m\} = -\sigma_j^{(k)}\alpha_l[\sigma_j^{(k)}, s_m] = 0$, which implies that $[s_m, \alpha_n] = [s_m, \sigma_j^{(k)}]\alpha_l + \sigma_j^{(k)}[s_m, \alpha_l] = \sigma_j^{(k)}[s_m, \alpha_l]$. So $[s_m, \alpha_n] = 0$ just when $[s_m, \alpha_l] = 0$, which means that $N \in G$ since $|\alpha_n| = |\alpha_l|$.

In Case 2, $\sigma_j^{(k)}\alpha_l$ has weight $|\alpha_l \pm 1| \Rightarrow [\alpha_l, \sigma_j^{(k)}] = 0$. Then (4.18) becomes $\{\sigma_j^{(k)}, \sigma_j^{(k)}\alpha_l s_m\} =$ $\sigma_j^{(k)}\alpha_l\{\sigma_j^{(k)}, s_m\} = 0$, which implies that $[s_m, \alpha_n] = \{s_m, \sigma_j^{(k)}\}\alpha_l + \sigma_j^{(k)}\{s_m, \alpha_l\} = \sigma_j^{(k)}\{s_m, \alpha_l\}$. So $[s_m, \alpha_n] = 0$ just when $\{s_m, \alpha_l\} = 0$, which means that $N \in G$ since $|\alpha_n| = |\alpha_l \pm 1|$.

Thus we have proved the following three conditions: that all the $R$'s used to compute the feedback are of the form $R_{N\in G}$; that for a given $M \in G$, $R_M$ will be the same for any state in the codespace; and that evolution via the master equation mixes the $R$'s of the form $R_{N\in G}$ only with each other. Therefore, taking the initial state to be *any* state in the codespace, including the true initial state and the entirely mixed state, produces

the same expression for the feedback when the master equation is evolved conditioned on a measurement record, and so it is not necessary to know the true initial state to use this protocol.

Another consequence of using the completely mixed state for feedback arises from the fact that doing so corresponds to discarding information about the state of the system. Therefore, this procedure should reduce the number of parameters needed to compute the feedback. Unfortunately, this only leads to a modest reduction in the number of parameters, which can be found by using a simple counting argument. There are $2^n/2^k = 2^{n-k}$ different error subspaces, including the no-error (code) space, and if we start with the completely mixed state in the codespace we do not need to worry at all about any movement within any of these spaces. We must only worry about which error space we are actually in, along with coherences between these spaces, so we find that $(2^{n-k})^2$ parameters are needed to describe the system.

At first this does not seem promising. However, if one encodes $mk$ qubits using $m$ copies of an $[[n, k, d]]$ code, as might well be the case for a quantum memory, the SME (4.13) will not couple the dynamics of the $m$ logical qubits; and, as in the bit-flip case, the initial condition for the controller's integration can still be the completely mixed state in the total codespace. Then the relevant scaling for this system, the dependence on $m$, is linear: the number of parameters is $m(2^{n-k})^2$.

## 4.3 Simulation of the bit-flip code

Because the bit-flip code feedback control scheme (4.5–4.8) uses a nonlinear feedback Hamiltonian, numerical simulation is the most tractable route for its study. Simulation of a quantum code protecting against an arbitrary error was not feasible due to limitations on computer power; however, simulation of the bit-flip code proved to be possible. In this section, we present the results of Monte Carlo simulations of the implementation of the protocol described in Section 4.1 for the bit-flip code.

### 4.3.1 Simulation details

To obtain $\rho_c(t)$, the quantum state conditioned on feedback, we directly integrated these equations using a simple Euler integrator and a Gaussian random number generator. We found stable convergent solutions when we used a dimensionless time step $\gamma dt$ on the order of $10^{-6}$ and averaged over $10^4$ quantum trajectories. As a benchmark, a typical run using these parameters took 2–8 hours on a 400MHz Sun Ultra 2. We found that more sophisticated Milstein [52] integrators converged more quickly but required too steep a reduction in time step to achieve the same level of stability. All of our simulations began in the state $\rho_c(0) = |\bar{0}\rangle\langle\bar{0}|$ because it is maximally damaged by bit-flipping noise and therefore yielded the most conservative results.

We used two measures to assess the behavior of our bit-flip code feedback control scheme. The first measure we used is the *codeword fidelity* $F_{cw}(t) = \mathrm{tr}(\rho_c(0)\rho_c(t))$, the overlap of the state with the target codeword. This measure is appropriate when one cannot perform strong measurements and fast unitary operations, a realistic scenario for many physical systems. We compared $F_{cw}(t)$ to the fidelities of one unprotected qubit $F_1(t) = \frac{1}{2}(1+e^{-2\gamma t})$ and of three unprotected qubits $F_3(t) = (F_1(t))^3$.

The second measure we used is the *correctable overlap*

$$F_{corr}(t) = \mathrm{tr}(\rho_c(t)\Pi_{corr}), \tag{4.19}$$

where

$$\begin{aligned} \Pi_{corr} \quad = \quad & \rho_0 + XII\rho_0XII \\ & + IXI\rho_0IXI + IIX\rho_0IIX \end{aligned} \tag{4.20}$$

is the projector onto the states that can be corrected back to the original codeword by discrete quantum error correction applied (once) at time $t$. This measure is appropriate when one can perform strong measurements and fast unitary operations, but only at discrete time intervals of length $t$. We compared $F_{corr}(t)$ to the fidelity $F_{\bar{3}}(t)$ obtained when, instead of using our protocol up to time $t$, no correction was performed until the final discrete quantum error correction at time $t$. As was shown in equation (2.18), the expression for

$F_{\bar{3}}(t)$ may be calculated analytically; it is $F_{\bar{3}}(t) = \frac{1}{4}(2 + 3e^{-2\gamma t} - e^{-6\gamma t}) \sim 1 - 3\gamma^2 t^2$.

### 4.3.2 Results

The simulations show that both the optimized estimate feedback scheme (4.12) and the heuristically-motivated feedback scheme (4.10) effectively protect a qubit from bit-flip decoherence. In Figs. 4.2 and 4.3 we show how these schemes behave for the (scaled) measurement and feedback strengths $\kappa/\gamma = 64$, $\lambda/\gamma = 128$ when averaged over $10^4$ quantum trajectories. Using our first measure, we see that at very short times, both schemes have codeword fidelities $F_{cw}(t)$ that follow the three-qubit fidelity $F_3(t)$ closely. For both schemes, $F_{cw}(t)$ improves and surpasses the fidelity of a single unprotected qubit $F_1(t)$. Indeed, perhaps the most exciting feature of these figures is that eventually $F_{cw}(t)$ surpasses $F_{\bar{3}}(t)$, the fidelity achievable by discrete quantum error correction applied at time $t$. In other words, our scheme alone outperforms discrete quantum error correction alone if the time between corrections is sufficiently long.

Looking at our second measure in Figs. 4.2 and 4.3, we see that $F_{corr}(t)$ is as good as or surpasses $F_{\bar{3}}(t)$ almost everywhere. For times even as short as a tenth of a decoherence time, the effect of using our protocol between discrete quantum error correction cycles is quite noticeable. This improvement suggests that, even when one can approximate discrete quantum error correction but only apply it every so often, it pays to use our protocol in between corrections. Therefore, our protocol offers a means of improving the fidelity of a quantum memory even after the system has been isolated as well as possible and discrete quantum error correction is applied as frequently as possible.

There is a small time range from $t \cong 0.01$ to $t \cong 0.05$ for the parameters used in Fig. 4.2 in which using our protocol before discrete quantum error correction actually underperforms the protocol of not doing anything before the correction. The simulations suggest that the reason for this narrow window of deficiency is that, in the absence of our protocol, it is possible to have two errors on a qubit (*e.g.*, two bit flips) that cancel each other out before discrete quantum error correction is performed. In contrast, this protocol will immediately start to correct for the first error before the second one happens, so we lose the advantage of this sort of cancellation. This view is supported by the fact that $F_{corr}(t)$ in our simulations always lies above the fidelity line obtained by subtracting such fortuitous cancellations from

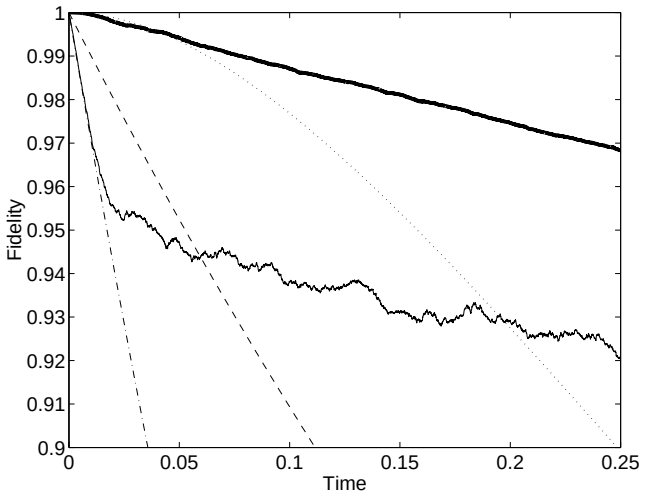



Figure 4.2: Behavior of our protocol with optimized feedback (4.12) for parameters $\kappa/\gamma = 64$, $\lambda/\gamma = 128$, averaged over $10^4$ quantum trajectories. The analytical curves shown are as follows: the dashed line is the fidelity of one decohering qubit, $F_1(t)$; the dashed-dotted line is the fidelity of three decohering qubits, $F_3(t)$; and the dotted line is the fidelity of an encoded qubit after one round of discrete error correction, $F_{\bar{3}}(t)$. Our simulation results are as follows: the solid line is the codeword fidelity $F_{cw}(t)$, and the thick solid line is the correctable overlap $F_{corr}(t)$.

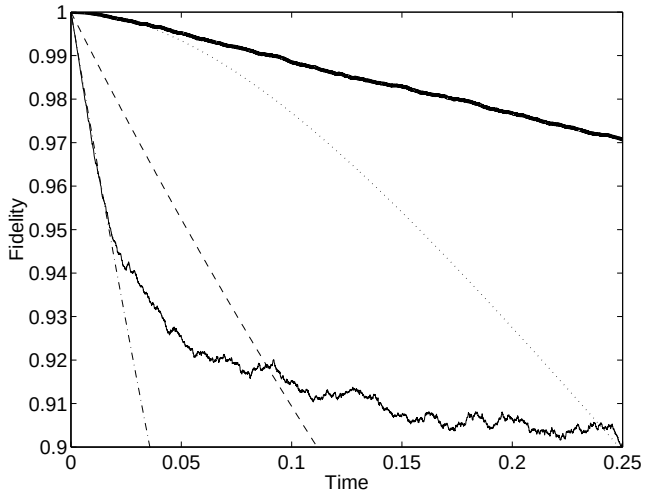



Figure 4.3: Behavior of our protocol with non-optimized feedback (4.10) for parameters $\kappa/\gamma = 64$, $\lambda/\gamma = 128$, averaged over $10^4$ quantum trajectories. As in Fig. 4.2, the dashed line is $F_1(t)$, the dashed-dotted line is $F_3(t)$, the dotted line is $F_{\bar{3}}(t)$, the solid line is $F_{cw}(t)$ and the thick solid line is $F_{corr}(t)$. Note that this feedback is qualitatively similar to that in Fig. 4.2 but does not perform as well.

$F_{\bar{3}}(t)$. In any case, this window can be made arbitrarily small and pushed arbitrarily close to the beginning of our protocol by increasing the measurement strength $\kappa$ and the feedback strength $\lambda$.

In Figs. 4.2 and 4.3, the $F_{cw}(t)$ line is much more jagged than the $F_{corr}(t)$ line. The jaggedness in both of these lines is due to statistical noise in our simulation and is reduced when we average over more than $10^4$ trajectories. The reason for the reduced noise in the $F_{corr}(t)$ line has to do with the properties of discrete quantum error correction—on average, neighboring states get corrected back to the same state by discrete quantum error correction, so noise fluctuations become smoothed out.

The improvement our optimized estimate feedback protocol yields beyond our heuristically-motivated feedback protocol is more noticeable in $F_{cw}(t)$ than in $F_{corr}(t)$ as seen in Figs. 4.2 and 4.3. Our optimized protocol acts to minimize the distance between the current state and the codespace, not between the current state and the space of states correctable back to the original codeword, so this observation is perhaps not surprising. In fact, optimizing feedback relative to $F_{corr}(t)$ is not even possible without knowing the codeword being pro-

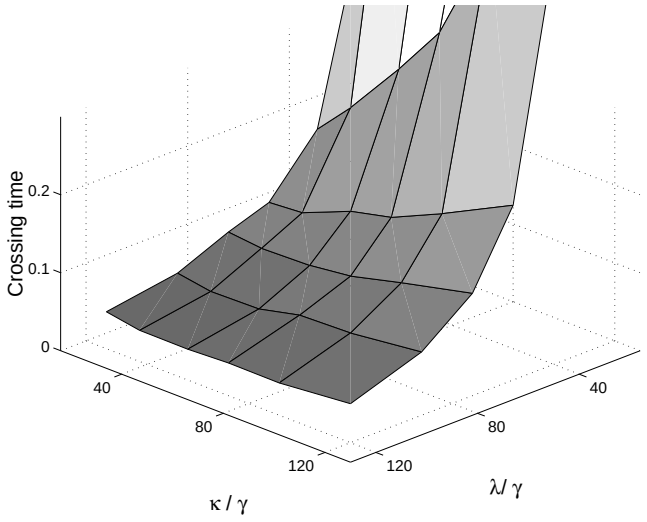



Figure 4.4: Time $\tau$ at which $F_{cw}(\tau) = F_1(\tau)$ as a function of measurement strength $\kappa/\gamma$ and feedback strength $\lambda/\gamma$. This crossing time is the time after which our optimized protocol improves the fidelity of a qubit beyond what it would have been if it were left to itself.

tected. Nevertheless, our optimized protocol does perform better, so henceforth we shall restrict our discussion to it.

We investigated how the protocol behaved when the scaled measurement strength $\kappa/\gamma$ and feedback strength $\lambda/\gamma$ were varied using the two measures described in Sec. 4.3.1. Our first measure, the codeword fidelity $F_{cw}(t)$, crosses the unprotected qubit fidelity $F_1(t)$ at various times $\tau$ as depicted in Fig. 4.4. This time is of interest because it is the time after which our optimized protocol improves the fidelity of a qubit beyond what it would have been if it were left to itself. Increasing the scaled feedback strength $\lambda/\gamma$ improves our scheme and reduces $\tau$, but the dependence on the scaled measurement strength $\kappa/\gamma$ is not so obvious from Fig. 4.4.

By looking at cross sections of Fig. 4.4, such as at $\lambda/\gamma = 80$ as in Fig. 4.5, we see that for a given scaled feedback strength $\lambda/\gamma$ there is a minimum crossing time $\tau$ as a function of measurement strength $\kappa/\gamma$. In other words, there is an optimal choice of measurement strength $\kappa/\gamma$. This optimal choice arises because syndrome measurements, which localize states near error subspaces, compete with Hamiltonian correction operations, which coherently rotate states between the nontrivial error subspaces to the trivial error subspace. This

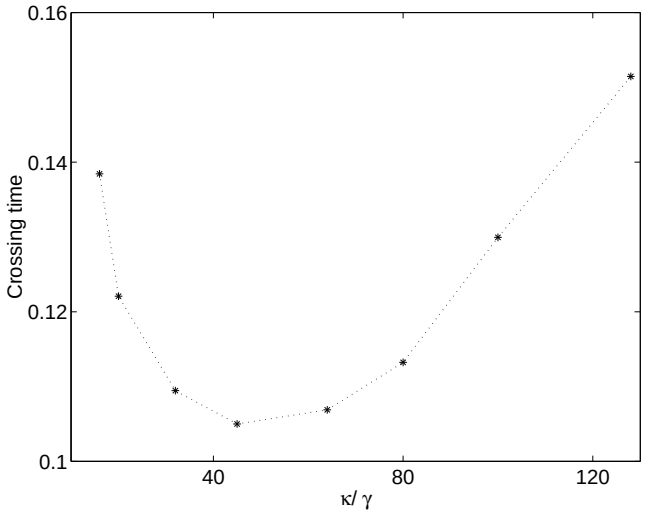



Figure 4.5: Time $\tau$ at which $F_{cw}(\tau) = F_1(\tau)$ as a function of measurement strength $\kappa/\gamma$, keeping correction strength fixed at $\lambda/\gamma = 80$.

phenomenon is a feature of our continuous-time protocol that is not present in discrete quantum error correction; in the former, measurement and correction are simultaneous, while in the latter, measurement and correction are separate non-interfering processes.

In order to study how our second measure, the correctable overlap $F_{corr}(t)$, varies with $\kappa$ and $\lambda$, we examined its behavior at a particular time. In Fig. 4.6 we plot $F_{corr}(t)$, evaluated at the time $t = 0.2/\gamma$, as a function of $\kappa$ and $\lambda$. As we found with the crossing time $\tau$, increasing $\lambda$ always improves performance, but increasing $\kappa$ does not because measurement can compete with correction. Since $F_{\bar{3}}(0.2/\gamma) \cong 0.927$, for all $\kappa$ and $\lambda$ plotted in Fig. 4.6, using our protocol between discrete quantum error correction intervals of time $0.2/\gamma$ improves the reliability of the encoded data.

## 4.4 Relaxing assumptions

### 4.4.1 Bandwidth-limited control

The feedback given in (4.12) is a bang-bang control: that is, the optimal solution occurs when the control parameters $\lambda_k$ are always at the maximum or minimum value allowable ($\lambda$ or $-\lambda$, respectively). Indeed, (4.12) assumes that it is possible to change the direction

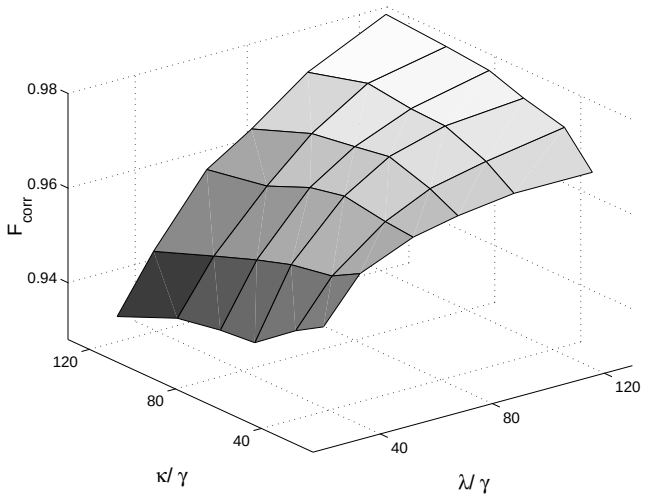



Figure 4.6: $F_{corr}$ at $\gamma t = 0.2$ as a function of measurement strength $\kappa/\gamma$ and feedback strength $\lambda/\gamma$. This quantity corresponds to the fidelity of a state given continuous error correction up to $\gamma t = 0.2$, at which point discrete error correction is performed.

of the feedback infinitely quickly.

Of course, in practice this is not the case, and one might envision replacing the sgn function in the feedback (4.12) with a sigmoid such as a tanh function. Fig. 4.7 shows numerical results for sgn feedback and tanh feedback. The figure of merit here is the codeword fidelity $F_{cw}(t) = \mathrm{tr}\,(\rho_c(0)\rho_c(t))$, the overlap of the state with the target codeword. This graph is a typical average over $10^4$ quantum trajectories for the (scaled) measurement and feedback strengths $\kappa/\gamma = 64$, $\lambda/\gamma = 128$, assuming perfect detector efficiency $\eta = 1$. The curves describing $F_{cw}(t)$ for both sgn and tanh feedback are qualitatively similar: they both improve and surpass the fidelity of a single unprotected qubit as well as eventually surpass the fidelity achievable by a round of discrete quantum error correction applied at time $t$. Further numerical study suggests that, unsurprisingly, the closer the sigmoid gets to a step function, the better the feedback protects the state.

## 4.5 Imperfect detection

Another natural question to ask is the following: how do our results change if the detectors are not perfect? Parametrizing the detector efficiency by the parameter $0 \leq \eta \leq 1$ (where

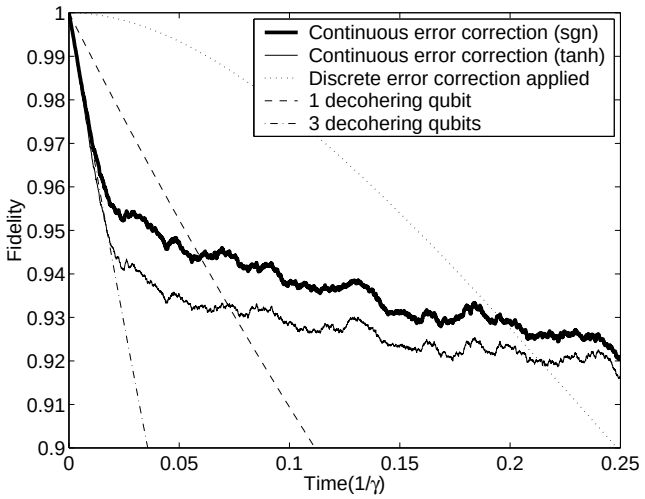



Figure 4.7: Behavior of our protocol with sgn and tanh feedback for parameters $\kappa/\gamma = 64$, $\lambda/\gamma = 128$, averaged over $10^4$ quantum trajectories. The analytical curves shown are as follows: the dashed line is the fidelity of one decohering qubit, the dashed-dotted line is the fidelity of three decohering qubits, and the dotted line is the fidelity of an encoded qubit after one round of discrete error correction at time $t$. The simulation results are given by the solid lines: the thick solid line is the codeword fidelity given the sgn feedback in (4.12), while the thin solid line is the codeword fidelity where $\mathrm{sgn}(x)$ in the feedback has been replaced by $\tanh(50x)$.

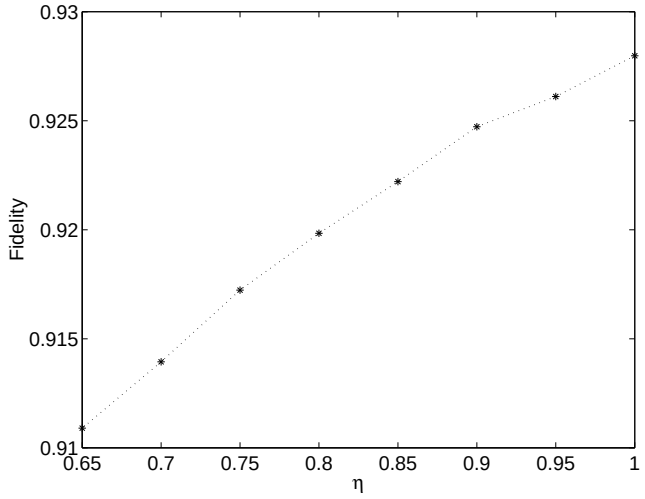



Figure 4.8: The codeword fidelity at time $\gamma t = 0.2$ as a function of detector efficiency $\eta$.

$\eta = 1$ denotes perfect efficiency) yields the following stochastic master equation describing the state conditioned upon the measurement results:

$$\begin{aligned} d\rho_c &= \gamma(\mathcal{D}[XII] + \mathcal{D}[IXI] + \mathcal{D}[IIX])\rho_c dt \\ &\quad +\kappa(\mathcal{D}[ZZI] + \mathcal{D}[IZZ] + \mathcal{D}[ZIZ])\rho_c dt \\ &\quad +\sqrt{\kappa\eta}(\mathcal{H}[ZZI]dW_1 + \mathcal{H}[IZZ]dW_2 + \mathcal{H}[ZIZ]dW_3)\rho_c \\ &\quad -i[F, \rho_c]dt \qquad (4.21) \\ dQ_1 &= 2\kappa\eta\langle ZZI\rangle dt + \sqrt{\kappa\eta}dW_1 \qquad (4.22) \\ dQ_2 &= 2\kappa\eta\langle IZZ\rangle dt + \sqrt{\kappa\eta}dW_2 \qquad (4.23) \\ dQ_3 &= 2\kappa\eta\langle ZIZ\rangle dt + \sqrt{\kappa\eta}dW_3. \qquad (4.24) \end{aligned}$$

Using the feedback (4.9,4.12) as before, the protocol can again be numerically tested using the SME above. Fig. 4.8 graphs the codeword fidelity at a particular time as a function of $\eta$; the time $\gamma t = 0.2$ was chosen as indicative of the general behavior of $F_{cw}$ as a function of $\eta$.

This figure shows that for efficiencies close to 1, the protocol still performs very well. In particular, there is no exponential dropoff when $\eta$ is decreased. This feedback does not propagate errors badly because it is a function of the entire measurement record, not

just instantaneous measurement results, which lends the protocol a limited inherent fault-tolerance.

## 4.6 Conclusion

In many realistic quantum computing architectures, weak measurements and Hamiltonian operations are likely to be the tools available to protect quantum states from decoherence. Moreover, even in quantum systems in which strong measurements and fast operations are well-approximated, such as ion traps [92], it is likely that these operations will only be possible at some maximum rate. This protocol is able to continuously protect unknown quantum states using only weak measurements and Hamiltonian corrections and can improve the fidelity of quantum states beyond rate-limited quantum error correction. Bandwidth-limited control and imperfect efficiency must be considered if our protocol for protecting unknown quantum states is to be of practical importance. These limitations, as long as they are not too severe, do not greatly hinder our ability to protect an unknown quantum state. In fact, one particular strength of this protocol is that, because it responds to the entire measurement record and not to instantaneous measurement results, it will not propagate errors badly and therefore has a limited inherent fault-tolerance that ordinary quantum error correction does not.

We expect that our protocol will be applicable to other continuous-time quantum information processes, such as reliable state preparation and fault-tolerant quantum computation. We also expect that our approach will work when different continuous-time measurement tools are available, such as direct photodetection. Finally, although current computing technology has limited our simulation investigation to few-qubit versions of our protocol, we are confident that many of the salient features we found in our three-qubit bit-flip code protocol will persist when our protocol is applied to larger codes.

# Chapter 5

# A more practical scheme for continuous error correction via estimate feedback

The protocol given in Chapter 4 has the rather unfortunate property that the state estimation feedback used in that protocol requires keeping track of a number of parameters that is exponential in the size of the code. Is there a way to reduce this computational overhead? In fact, this chapter will show through discussion of numerical simulations that if one is not concerned with optimality of the feedback, and if the measurement strength is sufficiently high compared to the decoherence time, it is possible to perform a relatively simple filtering post-processing on the measurement results that will serve to protect the quantum states against decoherence.

This work was done mainly in collaboration with M. Sarovar, and the numerical simulations in Sec. 5.2 are due to him. Sec. 5.2.4, a discussion of a possible experimental implementation, is due solely to M. Sarovar and G.J. Milburn and is included here for completeness.

## 5.1 The error correction scheme

When measurement strength is sufficiently high compared to decoherence time, the recorded measurement current bears a strong resemblance to a noisy series of quantum jumps. Although the noisiness of the currents renders them unsuitable for direct use in performing a continuous error correction protocol, this observation suggests that smoothing via a simple low-pass filter on the measurement currents may be sufficient post-processing for making

them useful for error correction.

The general form of the error correcting scheme proposed here is similar to discrete error control, but with a few modifications to deal with the incomplete information gained from the weak measurements. The scheme can be stated in four steps:

1. Encode information in a stabilizer code suited to the errors of concern.

2. Continuously perform weak measurements of the stabilizer generators, and smooth the measurement currents.

3. Depending on the signatures of the smoothed measurement currents, form conditioning signals for feedback operators on each physical qubit. These conditioning currents will be highly non-linear functions of the measurement currents because the conditional switching based on signatures is a non-linear operation.

4. Apply feedback Hamiltonians to each physical qubit, where the strength of the Hamiltonians is given by the conditioning signals formed in the previous step.

Given $m$ stabilizer generators and $d$ errors possible on our system, the stochastic master equation describing the evolution of a system under this error control scheme is

$$\begin{aligned} d\rho_c(t) &= \sum_{k=1}^{d} \gamma \mathcal{D}[E_k]\rho_c(t)dt \\ &+ \sum_{l=1}^{m} \kappa \mathcal{D}[M_l]\rho_c(t)dt + \sqrt{\kappa}\mathcal{H}[M_l]\rho_c(t)dW_l(t) \\ &+ \sum_{k=1}^{d} -iG_k(t)[F_k, \rho_c(t)]dt, \end{aligned} \tag{5.1}$$

where $\gamma$ is the error rate, $E_k$ are the errors, $\kappa$ is the measurement strength, $M_l$ are the measurement operators, $F_k$ is the feedback Hamiltonian correcting for error $E_k$, and $G_k$ is the feedback conditioning signal for $F_k$. Each $G_k$ is a conditional function of the signatures of all the smoothed stabilizer measurements, $\{M_l\}$. The assumption made here that the error rate is the same for all errors and that the measurement strength is the same for all measurement operators is made for simplicity and can be removed. In equation (5.1), the first line describes the effects of the error operators, the second line describes the effects of

the weak stabilizer generator measurements, and the third line describes the effect of the feedback.

This general scheme is illustrated by the following examples. The systems described by these examples are also the ones simulated in section 5.2.

### 5.1.1 Example: A one-qubit toy model

In this toy model, as in Chapter 4, the 'codespace' we want to protect is simply the state $|0\rangle$, and the errors are random applications of $X$; the protocol gathers information by measuring the stabilizer generator $Z$. Obviously this 'code' cannot be used for any information processing, but it is useful for investigating the behaviour of our feedback scheme.

The dynamics of this system before the application of feedback are described by the following SME:

$$\begin{aligned} d\rho_c(t) &= \gamma\mathcal{D}[X]\rho_c(t)dt + \kappa\mathcal{D}[Z]\rho_c(t)dt \\ &\quad +\sqrt{\kappa}\mathcal{H}[Z]\rho_c(t)dW(t), \end{aligned} \tag{5.2}$$

where $\gamma$ is the error rate and $\kappa$ is the measurement rate. The measurement current has the form

$$dQ(t) = 2\kappa\langle Z\rangle_c(t)dt + \sqrt{\kappa}dW(t). \tag{5.3}$$

Now, the measurement of $Z$ reveals whether the systems is in the 'codespace' or not, with $\langle Z\rangle = 1$ indicating the codespace, and $\langle Z\rangle = -1$ showing deviation from it. If the measurement is strong enough, it will tend to localize the state in one or the other of these two possibilities very quickly. However, we do not have direct access to $\langle Z\rangle_c$, but rather only to the noisy measurement current (5.3). Therefore we must smooth out the noise on it to obtain error information, and we will choose the following simple filter to do so:

$$R(t) = \frac{1}{\mathcal{N}}\int_{t-T}^{t} e^{-r(t-t')}dQ(t') \tag{5.4}$$

This integral is a convolution in time between the measurement signal and an exponentially decaying signal. In frequency space, this acts as a low pass filter, and thus the output of this operation is a smoothed version of the measurement current with high frequency

oscillations removed [1]. The filter parameters $r$ and $T$ determine the decay rate and length of the filter, respectively, and $\mathcal{N} = \frac{2\kappa}{r}(1 - e^{-rT})$ serves to normalize $R(t)$ such that it is centred around $\pm 1$.

We will use the signature of this smoothed measurement signal to infer the state of the system and thus to condition the feedback. Explicitly, the form of the feedback conditioning current is

$$G(t) = \begin{cases} R(t) & \text{if } R(t) < 0 \\ 0 & \text{otherwise} \end{cases} \tag{5.5}$$

The behavior of the system *with* feedback becomes

$$\begin{aligned} d\rho_c(t) &= \gamma\mathcal{D}[X]\rho_c(t)dt + \kappa\mathcal{D}[Z]\rho_c(t)dt \\ &\quad +\sqrt{\kappa}\mathcal{H}[Z]\rho_c(t)dW(t) \\ &\quad -i\lambda G(t)[X, \rho_c(t)]dt \end{aligned} \tag{5.6}$$

where $\lambda$ is the maximum feedback strength.

This feedback conditioning current is non-Markovian, so the most direct route to evaluating this error correction protocol is numerical simulation, which is discussed in section 5.2.

### 5.1.2 Example: Bit flip correction

This example is similar to the toy model above but looks at a more realistic error control situation. We will describe the dynamics of a continuous error correction scheme designed to protect against bit flips using the three qubit bit-flip code from chapter 2.

The measurement currents and SME of the system before the application of feedback

[1] This low pass filter is far from ideal. It is possible to design low-pass filters with much finer frequency selection properties (*e.g.* Butterworth filters) [65], and we expect schemes using such filters to perform better than this simpler version.

are

$$\begin{aligned} d\rho_c(t) &= \gamma(\mathcal{D}[XII] + \mathcal{D}[IXI] + \mathcal{D}[IIX])\rho_c(t)dt \\ &\quad +\kappa(\mathcal{D}[ZZI] + \mathcal{D}[IZZ])\rho_c(t)dt \\ &\quad +\sqrt{\kappa}(\mathcal{H}[ZZI]dW_1(t) \\ &\quad\quad + \mathcal{H}[IZZ]dW_2(t))\rho_c(t) \qquad (5.7) \\ dQ_1(t) &= 2\kappa\langle ZZI\rangle_c(t)dt + \sqrt{\kappa}dW_1(t) \qquad (5.8) \\ dQ_2(t) &= 2\kappa\langle IZZ\rangle_c(t)dt + \sqrt{\kappa}dW_2(t), \qquad (5.9) \end{aligned}$$

where $\gamma$ is the error rate for each qubit, and $\kappa$ is the measurement strength. Again, we will assume that the errors on different qubits are independent and occur at the same error rate, and also that the measurement strength is the same for both stabilizer generators.

As in the toy model, we must smooth the measurement currents in order to gain reliable error information. Therefore, the steps involved in the error correction scheme are the following:

1. Smooth the measurement currents using the following filter:

$$R_i(t) = \frac{1}{\mathcal{N}} \int_{t-T}^{t} e^{-r(t-t')} dQ_i(t') \qquad i = 1, 2 \qquad (5.10)$$

   The definition of this filter is analogous to (5.4).

2. Depending on the signatures of $R_1(t)$ and $R_2(t)$ apply the appropriate feedback Hamiltonian. That is,

   (a) If $R_1(t) < 0$ and $R_2(t) > 0$, apply $XII$.

   (b) If $R_1(t) > 0$ and $R_2(t) < 0$, apply $IIX$.

   (c) If $R_1(t) < 0$ and $R_2(t) < 0$, apply $IXI$.

   (d) If $R_1(t) > 0$ and $R_2(t) > 0$, do not apply any feedback.

These conditions translate into the following feedback conditioning currents:

$$G_1(t) = \begin{cases} R_1(t) & \text{if } R_1(t) < 0 \text{ and } R_2(t) > 0 \\ 0 & \text{otherwise} \end{cases} \tag{5.11}$$

$$G_2(t) = \begin{cases} R_2(t) & \text{if } R_1(t) > 0 \text{ and } R_2(t) < 0 \\ 0 & \text{otherwise} \end{cases} \tag{5.12}$$

$$G_3(t) = \begin{cases} R_1(t) & \text{if } R_1(t) < 0 \text{ and } R_2(t) < 0 \\ 0 & \text{otherwise} \end{cases} \tag{5.13}$$

Under this scheme, the SME describing the system dynamics *with* feedback becomes simply

$$\begin{aligned} d\rho_c(t) &= \gamma(\mathcal{D}[XII] + \mathcal{D}[IXI] + \mathcal{D}[IIX])\rho_c(t)dt \\ &\quad +\kappa(\mathcal{D}[ZZI] + \mathcal{D}[IZZ])\rho_c(t)dt \\ &\quad +\sqrt{\kappa}(\mathcal{H}[ZZI]dW_1(t) + \mathcal{H}[IZZ]dW_2(t))\rho_c(t) \\ &\quad -i\lambda(G_1(t)[XII, \rho_c(t)] + G_2(t)[IXI, \rho_c(t)] \\ &\quad\quad + G_3(t)[IIX, \rho_c(t)])dt \end{aligned} \tag{5.14}$$

where $\lambda$ is the maximum feedback strength, which is assumed for simplicity to be the same for all the feedback Hamiltonians.

As before, the non-Markovian feedback signals make numerical simulation the most direct method of solution of this SME.

## 5.2 Simulation results

As a way of evaluating the performance of the general error control scheme using weak measurements and feedback, we numerically solved the SMEs described in the two examples of section 5.1. In these simulations, the error rate $\gamma$ was fixed, and the parameter space formed by $r, \lambda, T$, and $\kappa$ was explored. The numerical results showed that the parameter $T$ can be optimized as a function of $r$. In particular, $T$ should be chosen to be large enough so that the decaying exponential filter is not truncated prematurely. A $T$ that is some large enough multiple of the filter's time constant, $1/r$, is ideal. Therefore, in what follows we

will consider the results based on the parameter space formed by the three parameters $r, \lambda$, and $\kappa$.

Since the one qubit toy model has all these free parameters, it is a good model in which to qualitatively explore this parameter space, and the smaller state space of this model makes its simulation far more computationally tractable than simulating a larger code, even the relatively small bit-flip code.

### 5.2.1 The toy model

We chose to simulate the dynamics of (5.6) by way of an associated stochastic Schrödinger equation (SSE) for two reasons: (i) it is less computationally intensive, (ii) it allows us to look at individual trajectories of the system if desired. The form of this associated SSE is as follows:

$$\begin{aligned} d|\psi_c(t)\rangle &= dN(t)(X|\psi_c(t)\rangle - |\psi_c(t)\rangle) + \sqrt{\kappa}\, dW(t)(Z - \langle Z\rangle(t))|\psi_c(t)\rangle \\ &\quad -\frac{\kappa}{2}(1 - \langle Z\rangle(t)Z)^2|\psi_c(t)\rangle dt - i\lambda G(t)X|\psi_c(t)\rangle dt, \end{aligned} \tag{5.15}$$

where $dN(t)$ is a random variable that is either 0 or 1 at each time step, and is distributed according to the error rate $\gamma$.

The SSE was solved using Euler numerical integration with time steps $dt = 10^{-4}$. When ensemble averages were required — that is, when we were interested in the behaviour of $\rho_c(t)$ — 600 trajectories were averaged over. To evaluate the performance of the protocol, we used the *codeword fidelity*: $F(t) = \langle\psi(0)|\rho(t)|\psi(0)\rangle$. Here, $|\psi(0)\rangle$ is the initial state of the system, which is taken to be $|0\rangle$ unless otherwise specified.

Figure 5.1 shows a sample trajectory from the one qubit simulation. The figure shows the expectation value of the $Z$ measurement as a function of time and also the superimposed filtered measurement signal, $R(t)$. The transitions of the expectation value of $Z$ to $-1$ are due to errors, and the transitions back to $+1$ are due to feedback correction.

We used this toy model primarily to gain insight into the choice of parameters that lead to optimal error correction. The conclusions drawn from exploring the parameter space using this one qubit simulation are very similar to those in Chapter 4:

1. Performance improves as $\lambda$, the feedback strength, is increased. This improvement is

to be expected because the greater feedback strength corresponds to faster correction.

2. The interplay between the two processes of measurement and feedback must be considered. In particular, if the measurement strength is too strong compared to the feedback strength, the measurement process, which tends to localize the state in the codespace, disrupts the feedback correction process, which tends to take the state out of the codespace. Numerical results as well as heuristic analytical calculations show that the magnitude of the measurement strength should be roughly of the same order of magnitude as the feedback Hamiltonian strength for optimal correction.

3. The decay rate of the filter, $r$, is determined by the strength of the feedback, $\lambda$: that is, given a strong feedback Hamiltonian, it is necessary to have a responsive conditioning current, one with little memory.

Given the strong dependence between parameters identified by these one qubit simulations, there are really only two free controllable parameters in the system: $\kappa$ and $\lambda$. In practice, neither the measurement strength or the feedback strength are completely configurable. The physical implementation scheme typically limits the range of these parameters, and in section 5.2.4 we shall consider the practical ranges for one particular implementation and put bounds on the error rate allowing for error control via this feedback scheme.

It is instructive to note that the free parameters of the protocol are all physical parameters: the optimal operating regime of the protocol is defined by the system's physical features rather than those of the introduced filter. Therefore, it is in principle possible to design a filter that allows the protocol to perform optimally for a given set of physical parameters ($\kappa$ and $\lambda$).

### 5.2.2 Three qubit code simulation

The simulation of the three qubit bit-flip code behaves in much the same way as the one qubit version, but with one key difference: for the one qubit 'code', a double error event – where an error occurs on the qubit before we have corrected the last error – is not too damaging: in this case, the error correcting feedback mechanism detects a traversal back into the 'codespace' and thus stops correcting. In the three qubit code, this situation is a little more complicated. Let us consider the situation in which a second error happens

while a previous error is being corrected. If this second error happens to be on the same qubit as the one being corrected, then in consonance with the one qubit 'code', it is not too damaging. However, if the second error is on one of the two qubits not being corrected, an irrecoverably damaging event occurs, because in this case the stabilizer measurements cease to provide accurate information about the error location, and the protocol's 'corrections' actually introduce errors.

This problem identifies a key consideration in any continuous, feedback based error correction scheme. The finite duration of the detection and correction window means that we wish to choose our parameters with this finite window small enough that the probability of an error we cannot correct (in this case, two errors on different qubits) is negligible.

The SSE that describes the dynamics of the three qubit error correction scheme is

$$\begin{aligned} d|\psi_c(t)\rangle &= dN_1(t)(XII|\psi_c(t)\rangle - |\psi_c(t)\rangle) + dN_2(t)(IXI|\psi_c(t)\rangle - |\psi_c(t)\rangle) \\ &\quad + dN_3(t)(IIX|\psi_c(t)\rangle - |\psi_c(t)\rangle) \\ &\quad + \sqrt{\kappa}\, dW_1(t)(ZZI - \langle ZZI\rangle(t))|\psi_c(t)\rangle + \sqrt{\kappa} dW_2(t)(IZZ - \langle IZZ\rangle(t))|\psi_c(t)\rangle \\ &\quad - \frac{\kappa}{2}(1 - \langle ZZI\rangle(t)ZZI)^2\ |\psi_c(t)\rangle dt\ -\ \frac{\kappa}{2}(1 - \langle IZZ\rangle(t)IZZ)^2\ |\psi_c(t)\rangle dt \\ &\quad - i\lambda G_1(t)XII\ |\psi_c(t)\rangle dt\ -\ i\lambda G_2(t)IXI\ |\psi_c(t)\rangle dt\ -\ i\lambda G_3(t)IIX\ |\psi_c(t)\rangle dt \end{aligned} \tag{5.16}$$

This SSE is of course an unravelling of the SME (5.14), and all parameters are defined as for that equation.

As in the one qubit case, we solved this differential equation using an Euler method with timesteps $dt = 10^{-4}$. Again, ensemble averages were done over 600 trajectories when needed. The initial state used was $|000\rangle$, and the performance was measured using the codeword fidelity $F_3(t) = \langle 000|\rho(t)|000\rangle$. A true fidelity measure of the protocol performance would average over all possible input states; however, because $|000\rangle$ is most susceptible to bit-flip errors, the fidelity we use can be considered a worst case performance analysis.

The performance of the error correction scheme using this code is summarized by Figure 5.2. This figure shows the fidelity versus time curves ($F_3(t)$) for several values of error rate ($\gamma$). Each plot also shows the fidelity curve ($F_1(t)$) for one qubit in the absence of error correction. A comparison of these two curves shows that the fidelity is preserved for a longer period of time by the error correction scheme for small enough error rates. Furthermore, for

small error rates ($\gamma < 0.3$) the $F_3(t)$ curve shows a vast improvement over the exponential decay in the absence of error correction. However, we see that past a certain error rate, the error correcting scheme becomes unable to handle the errors and becomes ineffective.

The third line in the plots of figure 5.2 is of the average fidelity achievable by discrete quantum error correction—using the same three qubit code—when the time between the detection-correction operations is $t$. The value of this fidelity ($F_{3d}(t)$) as a function of time was analytically calculated in Chapter 2:

$$F_{3d} = \frac{1}{4}(2 + 3e^{-2\gamma t} - e^{-6\gamma t}). \tag{5.17}$$

A comparison between $F_3(t)$ and $F_{3d}(t)$ highlights the relative merits of the two schemes. The fact that the two curves cross each other for large $t$ indicates that if the time between applications of discrete error correction is sufficiently large, then a continuous protocol will preserve fidelity better than a corresponding discrete scheme. In fact, this comparison suggests that a hybrid scheme, where discrete error correction is performed relatively infrequently on a system continuously protected by a feedback protocol, might be a viable approach to error control.

All the $F_3(t)$ curves show an exponential decay at very early times, $t \approx 0$ to $t \approx 0.1$. This decay occurs because our simulation does not smooth the measurement signal until enough time has passed to get a full buffer of measurements; that is, filtering and feedback only start at $t = T$, and thus the $F_3(t)$ curve follows the decay of three unprotected qubits for $t < T$. This problem should be at least partially remedied by a more complicated scheme that smooths the measurement signal and applies feedback even when it has access to fewer than $T/dt$ measurements.

### 5.2.3 Inefficient measurement

We have modeled all our measurement processes as being perfect. In reality, detectors will be inefficient and thus yield imperfect measurement results. This inefficiency is typically represented by a parameter $\eta$ that can range from 0 to 1, where 1 denotes a perfect detector. How is this feedback protocol affected by non-unit efficiency detection?

To examine this question, we simulated the three qubit code with inefficient detection.

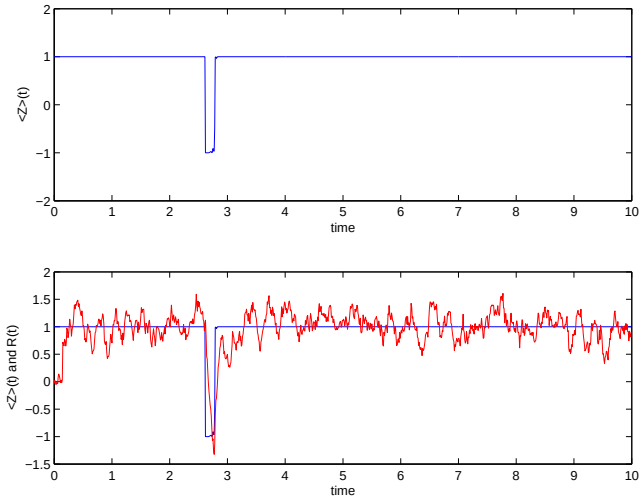



Figure 5.1: A sample trajectory of the one qubit "code" with feedback. The top graph just shows the expectation value of $Z$, and the bottom graph shows expectation value of $Z$ and the filtered signal $R(t)$.

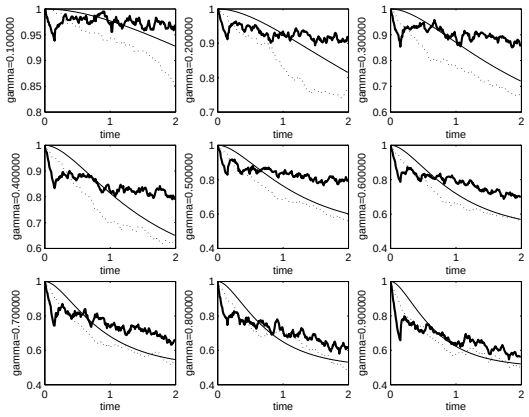



Figure 5.2: Fidelity curves with and without error correction for several error rates. The thick solid curve is the fidelity of the three qubit code with error correction, $F_3(t)$ (parameters used: $dt = 10^{-4}, \kappa = 150, \lambda = 150, r = 20, T = 1500 \times dt$). The dotted curve is the fidelity of one qubit without error correction, $F_1(t)$. The thin solid curve is the fidelity achievable by discrete quantum error correction when the duration between applications is $t$, $F_{3d}(t)$.

The evolution SME and the measurement currents in the presence of inefficient detection are as follows:

$$\begin{aligned} d\rho_c(t) &= \gamma(\mathcal{D}[XII] + \mathcal{D}[IXI] + \mathcal{D}[IIX])\rho_c(t)dt \\ &\quad +\kappa(\mathcal{D}[ZZI] + \mathcal{D}[IZZ])\rho_c(t)dt \\ &\quad +\sqrt{\kappa\eta}(\mathcal{H}[ZZI]dW_1(t) + \mathcal{H}[IZZ]dW_2(t))\rho_c(t) \\ &\quad -i\lambda(G_1(t)[XII, \rho_c(t)] + G_2(t)[IXI, \rho_c(t)] \\ &\quad\quad + G_3(t)[IIX, \rho_c(t)])dt \qquad (5.18) \\ dQ_1(t) &= 2\kappa\sqrt{\eta}\langle ZZI\rangle_c(t)dt + \sqrt{\kappa}dW_1(t) \qquad (5.19) \\ dQ_2(t) &= 2\kappa\sqrt{\eta}\langle IZZ\rangle_c(t)dt + \sqrt{\kappa}dW_2(t) \qquad (5.20) \end{aligned}$$

where $0 < \eta \leq 1$ is the measurement efficiency, and all other quantities are the same as in equations (5.7) and (5.14).

The results of these simulations are summarized by figure 5.3. Interestingly, the slope of the decay of fidelity with decreasing $\eta$ is very small. In particular, the graph does not exponentially decay as do Markovian feedback protocols, which suggests that this protocol has a certain tolerance to inefficiencies in measurement. This tolerance occurs because the filtering has a finite time window: the feedback is computed using more information than the instantaneous measurement results. Also, as in the full state estimation protocol of Chapter 4, because the feedback conditioning current is a function of a measurement record history —as opposed to just the current measurement— errors induced by inefficient measurement tend not to be so damaging. Here we see the true strength of this error correction scheme: it combines the robustness of a state estimation based feedback protocol with the low post-processing overhead normally associated with a Markovian feedback protocol.

### 5.2.4 Solid-state quantum computing with RF-SET readout

In this section we study the possibility of applying this error correction technique to a particular quantum computing architecture.

Several schemes for solid-state quantum computing have been proposed [42, 58, 82, 63]. These use the charge or spin degree of freedom of single particles to represent logical qubits,

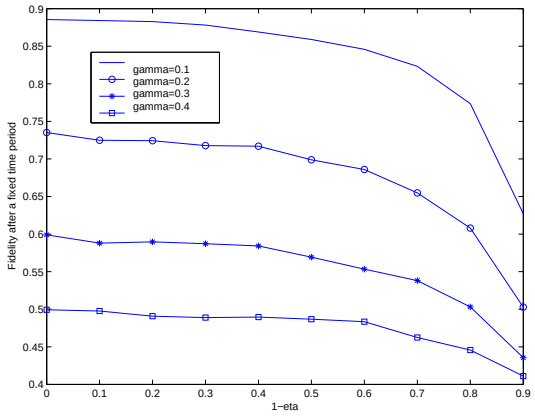



Figure 5.3: Average fidelity after a fixed amount of time as a function of $1-\eta$ for several error rates (parameters used: $dt = 10^{-4}, \kappa = 50, \lambda = 50, r = 10, T = 1500 \times dt$).

and measurement involves probing this degree of freedom.

Here we examine the weak measurement of one such proposal that uses coherently coupled quantum dots (CQDs) and an electron that tunnels between the dots [41]. The dots are formed by two P donors in Si, separated by a distance of about 50nm. Surface gates are used to remove one electron from the double donor system leaving a single electron on the P-$P^+$ system. This system can be regarded as a double well potential. Surface gates can then be used to control the barrier between the wells as well as the relative depth of the two wells. Using surface gates, the wells can be biased so that the electron can be well localized on either the left $|L\rangle$ or the right $|R\rangle$ of the barrier. These (almost) orthogonal localized states are taken as the logical basis for the qubit, $|0\rangle = |L\rangle, \quad |1\rangle = |R\rangle$. It is possible to design the double well system so that, when the well depths are equal, there are only two energy eigenstates below the barrier. These states are the symmetric ground state $|+\rangle$ and the antisymmetric first excited state $|-\rangle$. A state localized on the left (right) of the barrier is then well approximated as a linear superposition of these two states,

$$|L\rangle = \frac{1}{\sqrt{2}}(|+\rangle + |-\rangle) \tag{5.21}$$

$$|R\rangle = \frac{1}{\sqrt{2}}(|+\rangle - |-\rangle) \tag{5.22}$$

An initial state localized in one well will then tunnel to the other well at the frequency

$\Delta = (\epsilon_+ - \epsilon_-)/\hbar$ where $\epsilon_\pm$ are the two energy eigenstates below the barrier.

The Pauli matrix, $Z = |L\rangle\langle L| - |R\rangle\langle R|$, is diagonal in this localized state basis. The Hamiltonian for the system can be well approximated by

$$H = \hbar\frac{\omega(t)}{2}Z + \hbar\frac{\Delta(t)}{2}X \tag{5.23}$$

where $X = |L\rangle\langle R| + |R\rangle\langle L|$. Surface gates control the relative well depth $\hbar\omega(t)$ (a bias gate control) and the tunnelling rate $\Delta(t)$, (a barrier gate control) which are therefore time dependent. For non-zero bias the energy gap between the ground state and the first excited state is $E(t) = \hbar\sqrt{\omega(t)^2 + \Delta(t)^2}$. Further details on the validity of this Hamiltonian and how well it can be realised in the $PP^+$ in Si system can be found in [8].

A number of authors have discussed the sources of decoherence in a charge qubit system such as this one[8, 24, 41]. For appropriate donor separation, phonons can be neglected as a source of decoherence. The dominant sources of decoherence then arise from fluctuations in voltages on the surface gates controlling the Hamiltonian and electrons moving in and out of trap states in the vicinity of the dot. This latter source of decoherence is expected to occur on a longer time scale and is largely responsible for $1/f$ noise in these systems. In any case both sources of decoherence can be modelled using the well known spin-boson model [90]. The key element of this model for the discussion here is that the interaction energy between the qubit and the reservoir is a function of $Z$.

If the tunnelling term proportional to $\Delta(t)X$ in Eq. (5.23) were not present, decoherence of this kind would lead to pure dephasing. However, in a general single qubit gate operation, both dephasing and bit-flip errors can arise in the spin-boson model. We can thus use the decoherence rate calculated for this model as the bit-flip error rate in our feedback error correction model. We will use the result from the detailed model of Hollenberg et al. [41] for a device operating at 10K, and set the error rate $\gamma = 1.4 \times 10^6 \mathrm{s}^{-1}$. This rate could be made a factor of ten smaller by operating at lower temperatures and improving the electronics controlling the gates.

We now turn to estimating the measurement rate, $\kappa$, for the $PP^+$ system. In order to readout the qubit in the logical basis we need to distinguish a single electron in the left or the right well quickly and with high probability of success (efficiency). The technique of

choice is currently based on radio frequency single electron transistors (RF-SET)[75]. We will use the twin SET implementation of Buehler et al. [12].

In an RF SET the Ohmic load in a tuned tank circuit comprises a single electron transistor with the qubit acting as a gate bias. The two different charge states of the qubit provide two different bias conditions for the SET, producing two different resistive loads, and thus two levels of power transmitted through the tank circuit. The electronic signal carries a number of noise components: for example, the Johnson-Nyquist noise of the circuit, random changes in the SET bias conditions due to fluctuating trap states in the SET, etc. The measurement must be operated in such a way that the charge state of the qubit can be quickly discerned as a departure of the signal from some fiducial setting, despite the noise. Clearly it takes some minimum time interval, $t_M$, to discriminate a qubit signal change from a random noisy fluctuation. We need to keep the measurement time as short as possible. However if the measurement time is too short, one may mistake a large fluctuation, due to a non-qubit based change in bias conditions, for the real signal. In other words one may mistake a 1 for a 0, and vice versa. The probability of this happening is the efficiency of the measurement, $\eta(t_M)$, which depends on the measurement time. The key performance parameters are (i) the measurement time, $t_M$, and (ii) the efficiency $\eta(t_M)$. An additional parameter that is often quoted is the minimum charge sensitivity per root hertz, $S$. Given $t_M$, $S$ determines a minimum change in the charge, $\Delta q$, that can be seen by the RF-SET at a given bias condition. In [12], a measurement time of $t_M = 6 \times 10^{-6}$s was found for a signal of $\Delta q = 0.2e$ and an efficiency of $10^{-6}$. We now need to relate this measurement time to the measurement decoherence rate parameter, $\kappa$, of our ideal feedback model.

If the measurement were truly quantum limited (that is to say, the signal-to-noise ratio is determined only by the decoherence rate $\kappa$), the inverse measurement time would be of the same order of magnitude as the decoherence rate (see [31]). The measurement described in Buehler et al.[12] will almost certainly not be quantum limited. However, here we will assume the measurement to be quantum limited, so as to obtain a lower limit to the measurement decoherence rate. Thus we take $\kappa = 10^6 \mathrm{s}^{-1}$.

We next need to estimate typical values for the feedback strength. From Eq. (5.6) we see that the feedback Hamiltonian is proportional to an $X$ operator. In the charge qubit example, this corresponds to changing the tunneling rate for each of the double dot systems

that comprise each qubit. The biggest tunneling rate ($\Delta$) occurs when the bias of the double wells makes it symmetric. In [8], the maximum tunneling rate is about $10^9$ s$^{-1}$, for a donor separation of 40nm. A large tunneling rate makes for a fast gate, and thus a fast correction operation. Thus the maximum value of $\lambda$ can be taken to be $10^9$ s$^{-1}$.

To summarize, in the PP$^+$ based charge qubit, with RF-SET readout, we have $\gamma \approx \kappa \approx 10^6 \text{s}^{-1}$, and $\lambda \approx 10^9 \text{s}^{-1}$.

The fact that the measurement strength and the error rate are of the same order of magnitude for this architecture is a problem for our error correction scheme. This means that the rate at which we gain information is about the same as the rate at which errors happen, and it is difficult to operate a feedback correction protocol in such a regime. Although it is unlikely that the measurement rate could be made significantly larger in the near future, as mentioned above it is possible that the error rate could be made smaller by improvements in the controlling electronics. Thus it is interesting to consider how low the error rate would have to be pushed before our error control scheme becomes effective. To answer this question we ran the three qubit bit-flip code simulation using the parameters stated above and lowered the error rate until the error control performance was acceptable. We found that the fidelity after 1ms could be kept above 0.8 on average if the error rate, $\gamma$, is below $10^2$ s$^{-1}$ (with $\kappa = 10^6$ s$^{-1}$, and $\lambda = 10^7$ s$^{-1}$). So we see that a difference in order of magnitude of four between the measurement and feedback strengths, and the error rate, is about what this protocol (using the three qubit code) requires for reasonable performance. That is, we require

$$\frac{\kappa}{\gamma} \approx \frac{\lambda}{\gamma} \approx 10^4 \tag{5.24}$$

Of course, depending on the performance requirements this ratio may be larger or smaller. Also, a full optimization of the filter used in the scheme is likely to drive this ratio down by up to an order of magnitude.

We can compare the requirements of the three-qubit code with the one-qubit version. Given the same measurement and feedback parameters ($\kappa = 10^6$ s$^{-1}$, $\lambda = 10^7$ s$^{-1}$), the one-qubit 'code' can keep the fidelity above 0.8 after 1 ms when $\kappa/\gamma \approx \lambda/\gamma \approx 10$. That is, only one order of magnitude difference is required between the error rate and the measurement and feedback rates. This suggests that a key issue with feedback based error correction

schemes is *scalability*. The ratio between measurement and feedback rates and error rate has to increase along with the error correcting code size (in qubits).

## 5.3 Discussion and conclusion

We have described a practical scheme for implementing error correction using continuous measurement and Hamiltonian feedback and have demonstrated the validity of the scheme by simulating it for a simple error correction scenario. The simulations show that this error control scheme can be made very effective if the operational parameters (measurement strength, feedback strength, filter parameters) are well matched to the error rate of a given sytem. At the same time, the scheme uses relatively modest resources and thus is easy to implement, as well as robust in the face of measurement inefficiencies.

We also studied a solid-state quantum computing architecture with RF-SET readout and the feasibility of implementing this error correction protocol on it. Although the measurement and feedback rates currently possible on this architecture do not allow for error correction via this feedback scheme with the intrinsic error rate, it is foreseeable that as the controlling technology improves, this error control scheme will become possible on this architecture. From numerical simulations, we found the approximate parameter regime where the three qubit code using this scheme becomes effective – that is, exactly how much improvement is necessary before the scheme becomes feasible. It would be interesting to investigate this further and explore more rigorously how values of $\kappa/\gamma$ and $\lambda/\gamma$ dictate protocol performance.

In general, further exploration of the parameter space for various codes is of considerable interest. One direction that could be taken is to optimize the various parameters. For example, one might imagine that the optimum decay time of the filter depends somewhat on factors such as the inefficiency of the measurement.

Analytical models for both this scheme and the one given in Chapter 6 would be of interest. It is possible to design a rough model for the one-qubit case in which the finite-time response due to non-instantaneous correction is the main source of degradation of the state; this ought to be a good assumption when $\kappa$ is large. However, numerical simulations do not support this model; in particular, the one qubit simulations seem to show a steady

state, a feature not present in a model in which there is always some rate of degradation [74]. This result suggests more sophisticated behavior on the part of the correction protocol than we are currently able to understand, and an analytical model able to reproduce the behavior in the simulations is still unknown.

The numerical simulations seem to show a threshold error rate above which this particular error correction scheme does not work. More work could be done in identifying such a threshold, especially its dependence on $\kappa/\gamma$ and $\lambda/\gamma$. Another important consideration is the scalability of the protocol: how large do $\kappa$ and $\lambda$ need to be for good results as the size of the code grows larger? Preliminary simulation results suggest that the good values of $\kappa/\gamma$ and $\lambda/\gamma$ grow exponentially with the size of the code. Investigating this dependence further would give insight into how truly practical this proposal is.

# Chapter 6

# Quantum error correction for continuously detected errors

## 6.1 Introduction

The work in Chapters 4 and 5 presupposed that classical processing of currents could be done arbitrarily quickly, so the feedback was allowed to be an extremely complicated function of the entire measurement record in Chapter 4 and a rather less complicated one in 5. These sorts of complicated feedback can be modeled only by numerical simulations. In this chapter, by contrast, the feedback will be restricted to be directly proportional to measured currents, thus removing any need for classical post-processing. In the Markovian limit, this allows an analytical treatment. This simplification is possible because in this chapter we will assume that the errors are *detected*. That is, the experimenter knows precisely what sort of error has occurred because the environment that caused the errors is being continuously measured. Since the environment is thus acting as part of the measurement apparatus, the errors it produces could be considered measurement-induced errors.

There are a number of implementations in which measurement-induced errors of this sort may be significant. In the efficient linear optics scheme of Knill *et al.* [51], gates are implemented by nondeterministic teleportation. Failure of the teleportation corresponds to a gate error in which one of the qubits is measured in the computational basis with known result. In a number of solid state schemes, the readout device is always present and might make an accidental measurement of a qubit, even if the readout apparatus is in a quiescent state. An example is the use of RF single electron transistors to readout a charge

transfer event in the Kane proposal. Such a measurement is modelled as a weak continuous measurement [100]. While one supposes that the SET is biased in its low conductance state during qubit processing, it is useful to know that even if the device does accidentally make a measurement, the resulting error can be corrected.

In this chapter, we show that for certain error models and codes, Markovian feedback plus an additional constant Hamiltonian (a "driving Hamiltonian") can protect an unknown quantum state encoded in a particular codespace. Using the stabilizer formalism, we show that if there is one sort of error per physical qubit, and the error is detected perfectly, then it is always possible to store $n-1$ logical qubits in $n$ physical qubits. This works whether the detector record consists of discrete spikes (Poisson noise) or a continuous current (white noise). This suggest that if the dominant decoherence process can be monitored, then using that information to control the system Hamiltonian may be the key to preventing such decoherence (see also the example in [20]).

As a salient application of this formalism, we consider the special case of spontaneous emission. Stabilizing states against spontaneous emission by using quantum error correcting codes has been studied by several groups [59, 69, 4, 3]. Here we demonstrate that a simple $n$-qubit error correcting code, Markovian quantum feedback, and a driving Hamiltonian, is sufficient to correct spontaneous emissions on $n-1$ qubits. The result of encoding $n-1$ logical qubits in $n$ physical qubits has been recently independently derived by [43] for the special case of spontaneous emission; however, our scheme differs in a number of respects. We also show that spontaneous emission error correction by feedback can be incorporated within the framework of canonical quantum error correction, which can correct arbitrary errors.

Recall that the usual quantum error correction protocol has several steps: after the encoding and error, the error syndrome is measured (in a stabilizer code, this corresponds to measuring the stabilizer generators), and a correction is performed based on that error syndrome. In this chapter we will use a modified version of this protocol. In particular, we will not measure stabilizer elements. Instead, we will assume that a limited class of errors occurs on the system and that these errors are detectable: we know when an error has happened and what the error is. The correction back to the codespace can still be performed by a unitary recovery operator based on the information from the error measurement. Fig.

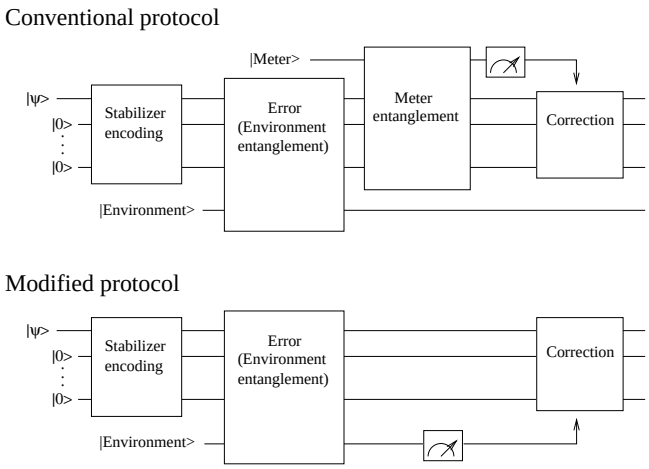



Figure 6.1: The top diagram shows the conventional stabilizer error correction protocol. After the state is encoded, an error occurs through coupling with the environment. To correct this error, the encoded state is entangled with a meter in order to measure the stabilizer generators, and then feedback is applied on the basis of those measurements. The bottom diagram shows our modified protocol, in which the error and measurement steps are the same. To correct the error in this protocol, the environment qubits are measured, and we feedback on the results of the environment measurement.

6.1 shows the difference between the conventional protocol and our modified protocol.

In this chapter, we will also consider operators of the form

$$T = T_1 \otimes \cdots \otimes T_n, \tag{6.1}$$

where $T_i$ is an arbitrary traceless one-qubit operator normalized such that its eigenvalues are $\{-1, 1\}$. Operators of this form are not generally Pauli-group stabilizers as presented in [38], as $T$ is not in general a member of $P_n$. However, because of the special form of $T$, $T$ is equivalent to a Pauli operator up to conjugation by a unitary that is a product of one-qubit unitaries, *i.e.*, there exists some $U = \bigotimes_{i=1}^{n} U_i$ such that $UTU^\dagger$ is a member of $P_n$. Therefore, choosing $T$ as the sole stabilizer generator for a code is equivalent, up to conjugation by a unitary, to choosing a member of the Pauli group as the stabilizer generator. (Note that additional constraints are necessary if $T$ is not the only stabilizer generator.)

## 6.2 Example: Spontaneous-emission correction

A particular example of a Poisson process error is spontaneous emission, in which the jump operator is proportional to $|0\rangle\langle 1|$, so that the state simply decays from $|1\rangle$ to $|0\rangle$ at random times. Indeed, if the decay is observed (say by emitting a photon which is then detected), this may be regarded as a destructive measurement of the operator $|1\rangle\langle 1|$.

Stabilizing states against the important decay process of spontaneous emission through application of error correcting codes has been studied by several groups [59, 69, 4, 3]. In [69] Plenio, Vedral and Knight considered the structure of quantum error correction codes and addressed the problem that spontaneous emission implies continuous evolution of the state even when no emission has occurred. They developed an eight-qubit code that both corrects one general error and corrects the no-emission evolution to arbitrary order.

More recently, in several papers Alber *et al.* [4, 3] have addressed a somewhat more specific problem relating to spontaneous emission from statistically independent resevoirs. In this formulation, the only errors possible are spontaneous emission errors, and the time and position of a particular spontaneous emission is known. They showed that given these constraints, a reduction of the redundancy in [69] was possible, and constructed a four-qubit code which corrects for one spontaneous emission error.

In fact, for the case considered in [4, 3], a very simple error correcting code consisting of just two qubits with feedback is sufficient to correct spontaneous emissions for a single logical qubit. A crucial difference from Refs. [4, 3] is that we call for a constant driving Hamiltonian in addition to the feedback Hamiltonian. Moreover, a simple code of $n$ qubits, with the appropriate feedback and driving Hamiltonians, can encode $n-1$ qubits and correct for spontaneous emissions when the position (*i.e.*, which qubit) and time of the jump are known. We also show that an equally effective protocol can be found for a diffusive unraveling of the spontaneous emission (as in homodyne detection).

### 6.2.1 Two-qubit code: Jump unraveling

The simplest system for which we can protect against detected spontaneous emissions is a system of two qubits. We consider the model in which the only decoherence process is due to spontaneous emission from statistically independent reservoirs. We will show that a

simple code, used in conjunction with a driving Hamiltonian, protects the codespace when the time and location of a spontaneous emission is known and a correcting unitary is applied instantaneously; the codespace suffers no decoherence.

The codewords of the code are given by the following:

$$\begin{aligned} |\bar{0}\rangle &\equiv (|00\rangle + |11\rangle)/\sqrt{2} \\ |\bar{1}\rangle &\equiv (|01\rangle + |10\rangle)/\sqrt{2}. \end{aligned} \tag{6.2}$$

In the stabilizer notation, this is a stabilizer code with stabilizer generator $XX$. Both codewords are $+1$ eigenstates of $XX$.

Following the presentation in Chapter 3, the jump operators for spontaneous emission of the $j$th qubit are

$$\Omega_j = \sqrt{\kappa_j dt}(X_j - iY_j) \equiv \sqrt{\kappa_j dt}a_j, \tag{6.3}$$

where $4\kappa_j$ is the decay rate for that qubit. In the absence of any feedback, the master equation is

$$\dot{\rho} = \sum_{j=1,2} \kappa_j \mathcal{D}[X_j - iY_j]\rho - i[H, \rho]. \tag{6.4}$$

If the emission is detected, such that the qubit $j$ from which it originated is known, it is possible to correct back to the codespace without knowing the state. This is because the code and error fulfill the necessary and sufficient conditions for appropriate recovery operations [50]:

$$\langle \psi_\mu | E^\dagger E | \psi_\nu \rangle = \Lambda_E \delta_{\mu\nu}. \tag{6.5}$$

Here $E$ is the operator for the measurement (error) that has occurred and $\Lambda_E$ is a constant. The states $|\psi_\mu\rangle, |\psi_\nu\rangle$ are the encoded states in Eq. (6.2) with $\langle \psi_\mu | \psi_\nu \rangle = \delta_{\mu\nu}$. These conditions differ from the usual condition (2.23) only by taking into account that we *know* a particular error $E = \Omega_j$ has occurred.

More explicitly, if a spontaneous emission on the first qubit occurs, $|\bar{0}\rangle \to |10\rangle$ and $|\bar{1}\rangle \to |11\rangle$, and similarly for spontaneous emission on the second qubit. Since these are orthogonal states, this fulfills the condition given in (6.5), so a unitary exists that will

correct this spontaneous emission error. One choice for the correcting unitary is

$$\begin{aligned} U_1 &= (XI - ZX)/\sqrt{2} \\ U_2 &= (IX - XZ)/\sqrt{2}. \end{aligned} \tag{6.6}$$

As pointed out in [69], a further complication is the nontrivial evolution of the state in the time between spontaneous emissions. From Chapter 3, this is described by the measurement operator

$$\begin{aligned} \Omega_0 &= II\,(1 - (\kappa_1 + \kappa_2)dt) - \kappa_1 dt ZI \\ &\quad -\kappa_2 dt IZ - iH dt. \end{aligned} \tag{6.7}$$

The non-unitary part of this evolution can be corrected by assuming a driving Hamiltonian of the form

$$H = -(\kappa_1 YX + \kappa_2 XY). \tag{6.8}$$

This result can easily be seen by plugging (6.8) into (6.7) with a suitable rearrangement of terms:

$$\begin{aligned} \Omega_0 &= II\,(1 - (\kappa_1 + \kappa_2)dt) - \kappa_1 dt ZI(II - XX) \\ &\quad - \kappa_2 dt IZ(II - XX), \end{aligned} \tag{6.9}$$

and since $II - XX$ acts to annihilate the codespace, $\Omega_0$ acts trivially on the codespace.

We then have the following master equation for the evolution of the system:

$$d\rho = \Omega_0 \rho \Omega_0^\dagger - \rho + dt \sum_{j=\{1,2\}} \kappa_j U_j a_j \rho a_j^\dagger U_j^\dagger, \tag{6.10}$$

where $U_j$ is the recovery operator for a spontaneous emission from qubit $j$. From Sec. 3, these unitaries can be achieved by the feedback Hamiltonian

$$H_{fb} = \sum_{j=1,2} \frac{dN_j(t)}{dt} V_j, \tag{6.11}$$

where $N_j(t)$ is the spontaneous emission count for qubit $j$, and $U_j = \exp(-iV_j)$. Here, we can see from the simple form of (6.6) that $V_j$ can be chosen as proportional to $U_j$. Since $U_j a_j \rho a_j^\dagger U_j^\dagger$ acts as the identity on the codespace by definition, and since we have shown that $\Omega_0 \rho \Omega_0^\dagger$ preserves the codespace, (6.10) must preserve the codespace.

Such a code is optimal in the sense that it uses the smallest possible number of qubits required to perform the task of correcting a spontaneous emission error, as we know that the information stored in one unencoded qubit is destroyed by spontaneous emission.

### 6.2.2 Two-qubit code: Diffusive unraveling

A similar situation applies for feedback of a continuous measurement record with white noise, as from homodyne detection of the emission. We use the same codewords, and choose $\phi_j = -\pi/2$ for the measurement. Then (6.6) suggests using the following feedback operators:

$$\begin{aligned} F_1 &= \sqrt{\kappa_1}(XI - ZX) \\ F_2 &= \sqrt{\kappa_2}(IX - XZ). \end{aligned} \tag{6.12}$$

If we use these feedback Hamiltonians with the same driving Hamiltonian (6.8) as in the jump case, the resulting master equation is, using (3.19),

$$\dot{\rho} = \kappa_1 \mathcal{D}[YI - iZX]\rho + \kappa_2 \mathcal{D}[IY - iXZ]\rho \tag{6.13}$$

We can see that this master equation preserves the codespace, by again noting that $YI - iZX = YI(II - XX)$, and similarly for $IY - iXZ$. The operator $II - XX$ of course acts to annihilate the codespace. This insight will be used in the next section to derive a feedback procedure for a more general measurement operator.

### 6.2.3 Generalizations to $n$ qubits

We will now demonstrate a simple $n$-qubit code that corrects for spontaneous emission errors only, while encoding $n-1$ qubits. Both of the above calculations (jump and diffusion) generalize. The master equation is the same as (6.4), but now the sum runs from 1 to $n$.

Again we need only a single stabilizer generator, namely $X^{\otimes n}$. The number of codewords is thus $2^{n-1}$, enabling $n-1$ logical qubits to be encoded. Since it uses only one physical qubit in excess of the number of logical qubits, this is again obviously an optimal code.

First, we consider the jump case. As in Sec. 6.2.1, a spontaneous emission jump fulfills the error-correction condition (6.5) (see Sec. 6.3.1 below). Therefore, there exists a unitary that will correct for the spontaneous-emission jump. Additionally, it is easy to see by analogy with (6.9) that

$$H = \kappa_j \sum_j X^{\otimes j-1} Y X^{\otimes n-j} \tag{6.14}$$

protects against the nontrivial no-emission evolution. Therefore the codespace is protected.

Next, for a diffusive unraveling, we again choose $\phi_j = -\pi/2$, as in Sec. 6.2.2. The same driving Hamiltonian (6.14) is again required, and the feedback operators generalize to

$$F_j = \sqrt{\kappa_j}\left(I^{\otimes j-1} X I^{\otimes n-j} + X^{\otimes j-1} Z X^{\otimes n-j}\right). \tag{6.15}$$

The master equation becomes

$$\dot{\rho} = \sum_j \kappa_j \mathcal{D}[I^{\otimes j-1} Y I^{\otimes n-j}(I^{\otimes n} - X^{\otimes n})]. \tag{6.16}$$

These schemes with a driving Hamiltonian do not have the admittedly desirable property of the codes given in [69, 4, 3] that if there is a time delay between the occurrence of the error and the application of the correction, the effective no-emission evolution does not lead to additional errors. Nevertheless, as pointed out in [3], the time delay for those codes must still be short so as to prevent two successive spontaneous emissions between correction; they numerically show that the fidelity decays roughly exponentially as a function of delay time. Therefore, we believe that this drawback of our protocol is not significant.

## 6.3 One-qubit general measurement operators

The form of the above example strongly indicates that there is a nice generalization to be obtained by considering stabilizer generators in more detail. In this section, we consider an arbitrary measurement operator operating on each qubit. We find the condition that

the stabilizers of the codespace must satisfy. We show that it is always possible to find an optimal codespace (with a single stabilizer group generator). We work out the case of diffusive feedback in detail and derive it as the limit of a jump process.

### 6.3.1 General unraveling

Different unravelings of the master equation (3.4) may be usefully parameterized by $\gamma$. In Sec. 3, we have seen that a simple jump unraveling has $\gamma = 0$, while the diffusive unraveling is characterized by $|\gamma| \to \infty$. We will now address the question of when a unitary correction operator exists for arbitrary $\gamma$, *i.e.*, when a measurement scheme with a given $\gamma$ works to correct the error.

Consider a Hilbert space of $n$ qubits with a stabilizer group $\{S_l\}$. Let us consider a single jump operator $c$ acting on a single qubit. We may then write $c$ in terms of Hermitian operators $A$ and $B$ as

$$\begin{aligned} e^{-i\phi}c &= \chi I + A + iB & (6.17)\\ &\equiv \chi I + \vec{a}\cdot\vec{\sigma} + i\vec{b}\cdot\vec{\sigma} & (6.18)\end{aligned}$$

where $\chi$ is a complex number, $\vec{a}$ and $\vec{b}$ are real vectors, and $\vec{\sigma} = (X, Y, Z)^T$.

We now use the standard condition (6.5), where here we take $E = c + \gamma$. Henceforth, $\gamma$ is to be understood as real and positive, since the relevant phase $\phi$ has been taken into account in the definition (6.17). The relevant term is

$$\begin{aligned} E^\dagger E &= (|\chi+\gamma|^2 + \vec{a}^2 + \vec{b}^2)I \\ &\quad + \mathrm{Re}(\chi+\gamma)A + \mathrm{Im}(\chi+\gamma)^* iB + (\vec{a}\times\vec{b})\cdot\vec{\sigma}) \\ &\equiv (|\chi+\gamma|^2 + \vec{a}^2 + \vec{b}^2)I + D, & (6.19)\end{aligned}$$

where $D$ is Hermitian.

Now we can use the familiar sufficient condition for a stabilizer code [38]: the stabilizer should anticommute with the traceless part of $E^\dagger E$. This condition becomes explicitly

$$0 = \{S, D\}. \quad (6.20)$$

As long as this is satisfied, there is some feedback unitary $e^{-iV}$ which will correct the error.

Normalization implies that when $E$ does not occur, there may still be nontrivial evolution. In the continuous time paradigm, where one Kraus operator is given by $E\sqrt{dt}$, the transform (3.10) tells us that the no-jump normalization Kraus operator is given by

$$\Omega_0 = 1 - \frac{1}{2}E^\dagger E dt - \frac{\gamma}{2}(e^{-i\phi}c - e^{i\phi}c^\dagger)dt - iHdt. \tag{6.21}$$

Now we choose the driving Hamiltonian

$$H = \frac{i}{2}DS + \frac{i\gamma}{2}(e^{-i\phi}c - e^{i\phi}c^\dagger). \tag{6.22}$$

This is a Hermitian operator because of (6.20). Then the total evolution due to $\Omega_0$ is just the identity, apart from a term proportional to $D(1-S)$, which annihilates the codespace. Thus for a state initially in the codespace, the condition (6.20) suffices for correction of both the jump and no-jump evolution.

A nice generalization may now be found for a set $\{c_j\}$ of errors such that $c_j$ [with associated operator $D_j$ as defined in (6.19)] acts on the $j$th qubit alone. Since $D_j$ is traceless, it is always possible to find some other Hermitian traceless one-qubit operator $s_j$ such that $\{s_j, D_j\} = 0$. Then we may pick the stabilizer group by choosing the single stabilizer generator

$$S = s_1 \otimes \cdots \otimes s_n \tag{6.23}$$

so that the stabilizer group is $\{1, S\}$. As noted in Sec. 6.1, this is not strictly a stabilizer group, as $S$ may not be in the Pauli group, but this does not change the analysis. Choosing $H$ according to this $S$ such that

$$H = \sum_j \frac{i}{2}D_j S + \frac{i\gamma_j}{2}(e^{-i\phi_j}c_j - e^{i\phi_j}c_j^\dagger) \tag{6.24}$$

will, by our analysis above, provide a total evolution that protects the codespace, and the errors will be correctable; furthermore, this codespace encodes $n-1$ qubits in $n$.

Note that we can now easily understand the $n$-qubit jump process error of spontaneous emission considered in Sec. 6.2. Here, $\gamma = 0$, $S = X^{\otimes n}$, and $D_j = 2\kappa_j Z_j$. Thus (6.20) is

satisfied, and the Hamiltonian (6.14) is derived directly from (6.24).

Moreover, one is not restricted to the case of one stabilizer; it is possible to choose a different $S_j$ for each individual error $c_j$. For example, for the spontaneous emission errors $c_j = X_j - iY_j$ we could choose $S_j$ as different stabilizers of the five-qubit code. This choice is easily made, as the usual generators of the five-qubit code are $\{XZZXI, IXZZX, XIXZZ, ZXIXZ\}$, as we saw in Chapter 2. For each qubit $j$, we may pick a stabilizer $S_j$ from this set which acts as $X$ on that qubit, and $X$ anticommutes with $D_j = Z_j$. This procedure would be useful in a system where spontaneous emission is the dominant error process; it would have the virtue of both correcting spontaneous emission errors by means of feedback as well as correcting other (rarer) errors by additionally using canonical error correction.

We note that the work in this section can very easily be modified to generalize the results of [43]. That work has the same error model as ours: known jumps occuring on separate qubits so that the time and location of each jump is known; but [43] postulates fast unitary pulses instead of a driving Hamiltonian. Their scheme for spontaneous emission depends on applying the unitary $X^{\otimes n}$ at intervals $T_c/2$ that are small compared to the rate of spontaneous emission jumps. They show that after a full $T_c$ period, the no-jump evolution becomes

$$U = e^{-iT_c H_c/2} X^{\otimes n} e^{-iT_c H_c/2} X^{\otimes n} = e^{-T_c/2 \sum_{i=1}^{N} \kappa_i} 1. \tag{6.25}$$

Thus the application of these pulses acts, as does our driving Hamiltonian, to correct the no-jump evolution. The generalization from spontaneous emission to general jump operator $c_j$ for their case is simple: the code is the same as in the above one-stabilizer protocol, with single stabilizer equal to (6.23). The fast unitary pulses are in this case also simply equal to (6.23).

### 6.3.2 Diffusive unraveling

The case of white-noise feedback, where $\gamma \to \infty$, is easily treated by recalling the master equation (3.18) for white-noise measurement and feedback. It is clear that the first term in (3.18) can be eliminated by choosing the constant driving Hamiltonian

$$H = -(e^{i\phi} c^\dagger F + e^{-i\phi} F c)/2 \tag{6.26}$$

which is automatically Hermitian. The problem then becomes choosing a feedback Hamiltonian $F$ such that $c - iF$ annihilates the codespace. The choice for $F$ can be made simply by noting that if the codespace is stabilized by some stabilizer $S$, we can choose

$$F = B - iAS. \tag{6.27}$$

Now, note that the decoherence superoperator $\mathcal{D}$ acts such that

$$\mathcal{D}[\chi I + L]\rho = \mathcal{D}[L]. \tag{6.28}$$

Then we know that $\mathcal{D}[c - iF] = \mathcal{D}[\chi I + A(I - S)]$ annihilates the codespace.

The only caveat is that $F$ is a Hamiltonian and therefore must be Hermitian. Then the choice (6.27) for $F$ is only possible if the anticommutator of $S$ and $A$ is zero:

$$\{S, A\} = 0. \tag{6.29}$$

Therefore, if we are given the measurement operator $e^{-i\phi}c = \chi + A + iB$, we must choose a code with some stabilizer such that condition (6.29) applies; then it is possible to find a feedback and a driving Hamiltonian such that the total evolution protects the codespace.

At first glance, it may seem odd that the condition for feedback does not depend at all upon $B$. This independence has to do with the measurement unraveling: the diffusive measurement record (3.13) depends only upon $e^{-i\phi}c + e^{i\phi}c^\dagger = 2(A + \chi)$.

### 6.3.3 Diffusion as the limit of jumps

It is instructive to show that the diffusive feedback process can be derived by taking the limit of a jump feedback process using the transformation (3.10). This takes several steps, and we use the treatment in [95] as a guide. But to begin, note that the condition (6.29) follows by considering Eq. (6.20) in the limit $\gamma \to \infty$, as the leading order term in $D$ is proportional to $A$.

Consider the jump unraveling picture with jump operator $c + \gamma$ for $\gamma$ large (but not infinite). Recall that in the error-correction picture given in Sec. 6.2, we postulated a feedback Hamiltonian $(dN/dt)V$ that produces a unitary correction $e^{-iV}$ that acts instantaneously

after the jump. In addition we will postulate a driving Hamiltonian $K$ that acts when no jump happens. In this picture, we will show that given the condition (6.29), it is possible to find asymptotic expressions for $V$ and $K$ so that the deterministic equation for the system preserves the stabilizer codespace. Finally, we will show that taking the limit $\gamma \to \infty$ leads to the expression for the feedback and driving Hamiltonians (6.26) and (6.27).

Let us consider the measurement operators for the unraveling with large $\gamma$ and $H = 0$. Following (3.10) these are

$$\begin{aligned} \Omega_1 &= \sqrt{dt}(c+\gamma) \\ \Omega_0 &= 1 - \frac{dt}{2}[c\gamma - c^\dagger\gamma + (c+\gamma)^\dagger(c+\gamma)], \end{aligned} \tag{6.30}$$

where we have assumed for simplicity that $\gamma$ is real. Now, including the feedback and driving Hamiltonians modifies these to

$$\begin{aligned} \Omega_1'(dt) &= \sqrt{dt}e^{-iV}(c+\gamma) \\ \Omega_0'(dt) &= e^{-iKdt}\Omega_0(dt) \\ &= 1 - iKdt - \frac{dt}{2}(c^\dagger c + 2\gamma c + \gamma^2). \end{aligned} \tag{6.31}$$

Following Ref. [95], expand $V$ in terms of $1/\gamma$ to second order: $V = V_1/\gamma + V_2/\gamma^2$ where the $V_i$ are Hermitian. Then expanding the exponential in (6.31) we get to second order

$$\begin{aligned} \Omega_1'(dt) &= \sqrt{dt}\left[1 - i\left(\frac{V_1}{\gamma} + \frac{V_2}{\gamma^2}\right) - \frac{1}{2}\frac{V_1^2}{\gamma^2}\right](A + iB + \gamma) \\ &= \sqrt{dt}\gamma\left[1 + \frac{\chi}{\gamma} + \frac{1}{\gamma}(A + iB - iV_1)\right. \\ &\quad \left. + \frac{1}{\gamma^2}(V_1^2/2 - iV_2 - i(A+iB)V_1)\right]. \end{aligned} \tag{6.32}$$

A reasonable choice for $V_1$, by analogy to (6.27), is $B - iAS$. Following [95], we also use (6.26) and (6.27) to choose $V_2$ and $K$; note that (6.27) is exactly the expression we would expect for $K$ from (6.22) in the limit as $\gamma$ is taken to infinity. We will proceed to show that

the choice for $V$ and $K$,

$$\begin{aligned} V_1 &= B - iAS && (6.33)\\ V_2 &= -(c^\dagger F + Fc)/2 && (6.34)\\ K &= -\gamma(B - iAS), && (6.35) \end{aligned}$$

leads to the correct evolution to second order in $\gamma$.

Now, the deterministic evolution is given by

$$d\rho = \Omega_0'\rho\Omega_0' + \Omega_1'\rho\Omega_1' - \rho. \tag{6.36}$$

Substituting (6.31)–(6.35) into (6.36) to second order in $\gamma$, after some algebra, gives the deterministic jump equation

$$d\rho = \mathcal{D}[A(1-S)]\rho \tag{6.37}$$

which of course acts as zero on the codespace.

Now we will show that taking the limit as $\gamma \to \infty$ leads to the feedback operators given in (6.26) and (6.27). We saw in (6.33) and (6.34) that the feedback Hamiltonian needed to undo the effect of the jump operator $c + \gamma$ was just

$$H_{fb} = \frac{dN(t)}{dt}\left(\frac{B - iAS}{\gamma} - \frac{c^\dagger F + Fc}{2}\right). \tag{6.38}$$

Keeping terms of two orders in $\gamma$ gives

$$\begin{aligned} H_{fb} &= \gamma(B - iAS) - \frac{c^\dagger F + Fc}{2} \\ &\quad + \frac{dN(t) - \gamma^2 dt}{\gamma dt}(B - iAS). \end{aligned} \tag{6.39}$$

The last term just becomes the current $\dot{Q}(t)$ as $\gamma$ approaches infinity, as in equation (3.12). Furthermore, we have not yet added in the driving Hamiltonian to the expression for the

feedback. Doing so yields

$$\begin{aligned} H_{\text{total}}(t) &= H_{fb} + K \\ &= \dot{Q}(t)(B - iAS) - \frac{c^\dagger F + Fc}{2} \end{aligned} \tag{6.40}$$

which is just what we obtained in the previous section. Thus we can see that this continuous current feedback can be thought of as an appropriate limit of a jump plus unitary correction process.

### 6.3.4 Relaxing assumptions: imperfect knowledge of measurement rate and imperfect detection

One implicit assumption the above analysis has made is that the measurement strength $\kappa$ must be known perfectly in order to apply the Hamiltonian 6.8. In a realistic situation, $\kappa$ may not be known precisely. Let us assume for simplicity that the error rates on both qubits are the same ($\kappa$): furthermore, imagine that our value for $\kappa$ is slightly in error by some fraction $\epsilon$, so that instead of correctly applying a driving Hamiltonian proportional to $\kappa$, we instead apply a Hamiltonian proportional to $\kappa(1+\epsilon)$. In this case, the differential equations describing the density matrix can be easily solved to yield a steady state that differs from the original state by an amount that is proportional to $\epsilon$ to first order. This analysis was done for the two-qubit system considered in Sec. 6.2.1, and also holds when the error rates on the two qubits are different. The principles of the analysis ought to hold for arbitrary numbers of qubits as well.

In addition, these results for feedback were obtained by assuming unit efficiency, *i.e.*, perfect detection. Realistically, of course, the efficiency $\eta$ will be less than unity. This results in extra terms in the feedback master equations we have derived [93]. In the jump case, the extra term is

$$\dot{\rho} = (1-\eta)\sum_j (c_j \rho c_j^\dagger - U_j c_j \rho c_j^\dagger U_j^\dagger). \tag{6.41}$$

In the diffusion case it is

$$\dot{\rho} = \frac{1-\eta}{\eta}\sum_j \mathcal{D}[F_j]. \tag{6.42}$$

In both cases this results in exponential decay of coherence in the codespace. This is because

the error correction protocol here relies absolutely upon detecting the error when it occurs. If the error is missed (jump case), or imperfectly known (diffusion case), then it cannot be corrected. This behavior is, of course, a property of any continuous-time error correction protocol that depends on correcting each error instantaneously (*e.g.*, [4, 3, 43]).

On the other hand, such behavior for Markovian feedback is in contrast to the state-estimation procedures used in Chapters 4 and 5. The latter procedures are much more robust under non-unit efficiency; indeed, given non-unit efficiency, we saw that they still worked to protect an unknown quantum state without exponential loss. This difference in performance occurs because state-estimation is a function of the entire measurement record, not just instantaneous measurement results, and thus does not propagate errors to the same extent that a Markovian feedback system does. Thus we can see that there is a certain tradeoff. Our Markovian feedback scheme relies upon calculational simplicity, but at the expense of robustness. The state-estimation procedure, conversely, is designed to be robust, but at the cost of computational complexity.

## 6.4 Universal quantum gates

Given a protected code subspace, one interesting question, as in [43], is to investigate what kinds of unitary gates are possible on such a subspace. For universal quantum computation on the subspace— the ability to build up arbitrary unitary gates on $k$ qubits— it suffices to be able to perform arbitrary one-qubit gates for all $k$ encoded qubits and a two-qubit entangling gate such as controlled-NOT for all encoded qubits $\mu, \nu$. Indeed, as is noted in [43], it is enough to be able to perform the *Hamiltonians* $\bar{X}_\mu, \bar{Z}_\mu$, and $\bar{X}_\mu \bar{X}_\nu$ for all $\mu, \nu$ [19]. We will demonstrate that performing these Hamiltonians with our protocol is possible for the spontaneous emission scheme given in Sec. 6.2, and then we will show how that construction generalizes for an arbitrary jump operator.

Recall that the example in Sec. 6.2 has single stabilizer $X^{\otimes n}$ and encodes $n-1$ logical qubits in $n$ physical qubits. To find the $2(n-1)$ encoded operations, we must find operators that together with the stabilizer generate the normalizer of $X^{\otimes n}$ [64]. In addition, if these operators are to act as encoded $X$ and $Z$ operations, they must satisfy the usual

commutation relations for these operators:

$$\begin{aligned} \{X_\mu, Z_\mu\} &= 0 \\ [X_\mu, Z_\nu] &= 0, \mu \neq \nu \\ [X_\mu, X_\nu] &= [Z_\mu, Z_\nu] = 0. \end{aligned} \tag{6.43}$$

Operators satisfying these constraints are easily found for this code:

$$\begin{aligned} \bar{X}_\mu &= I^{\otimes \mu-1} X I^{\otimes n-\mu} \\ \bar{Z}_\mu &= I^{\otimes \mu-1} Z I^{\otimes n-\mu-1} Z \\ \bar{X}_\mu \bar{X}_\nu &= I^{\otimes \mu-1} X I^{\otimes \nu-\mu-1} X I^{\otimes n-\nu}, \end{aligned} \tag{6.44}$$

where we assume $1 \leq \mu < \nu \leq n-1$. If we apply a Hamiltonian $H_{enc}$ given by any linear combination of the operators in (6.44), the resulting evolution is encapsulated in the expression for $\Omega_0$, from (6.7):

$$\begin{aligned} \Omega_0 &= (1 - \sum_j \kappa_j dt)1 \\ &\quad - \sum_j \kappa_j Z_j (1 - X^{\otimes n}) dt - i H_{enc} dt. \end{aligned} \tag{6.45}$$

As the first term is proportional to the identity and the second term acts as zero on the codespace, the effective evolution is given solely by $H_{enc}$ as long as the state remains in the codespace under that evolution. But because the encoded operations are elements of the normalizer, as we saw Chapter 2, applying $H_{enc}$ does not take the state out of the codespace. Furthermore, our protocol assumes that spontaneous emission jumps are corrected immediately and perfectly, so jumps during the gate operation will also not take the state out of the codespace. Thus we can perform universal quantum computation without having to worry about competing effects from the driving Hamiltonian.

The generalization to the scheme given in Sec. 6.3.1 to encode $n-1$ logical qubits in $n$ physical qubits is easily done. First we note that for the stabilizer $S$ given in the general scheme, we know that

$$S = U X^{\otimes n} U^\dagger \tag{6.46}$$

for some unitary $U = \bigotimes_{i=1}^{n} U_i$, so the encoded operations for that code are similarly given by

$$\begin{aligned} \bar{X}_\mu &= I^{\otimes \mu-1} U_\mu X U_\mu^\dagger I^{\otimes n-\mu} \\ \bar{Z}_\mu &= I^{\otimes \mu-1} U_\mu Z U_\mu^\dagger I^{\otimes n-\mu-1} U_n Z U_n^\dagger \\ \bar{X}_\mu \bar{X}_\nu &= I^{\otimes \mu-1} U_\mu X U_\mu^\dagger I^{\otimes \nu-\mu-1} U_\nu X U_\nu^\dagger I^{\otimes n-\nu}. \end{aligned} \tag{6.47}$$

Now, from (6.22) we can see that the generalization of (6.45) is

$$\Omega_0 = (1 - f dt)1 - \sum_j g_j D_j (1 - S) dt - i H_{enc} dt, \tag{6.48}$$

for real numbers $f, g_i$ given by expanding the expression (6.21). Again, since $D(1 - S)$ annihilates the codespace, the effective evolution is given solely by $H_{enc}$ as long as the state remains in the codespace under that evolution. Again, $H_{enc}$ is made up of normalizer elements, which do not take the state out of the codespace; and again jumps that occur while the gate is being applied are immediately corrected and thus do not affect the gate operation. Therefore, universal quantum computation is possible under our general scheme.

## 6.5 Multiple channels

The previous analysis was for one error channel per qubit. In this section we consider the following obvious generalization: What happens if there are multiple error channels $E^{(j),\alpha}$ per single qubit, all of which can be detected? (Here $j$ denotes the qubit on which the channel acts, and $\alpha$ indexes which channel it is.) Given a certain number of channels per qubit, what is the smallest number of stabilizers needed to be able to use our protocol? Equivalently, given $n$ physical qubits, how many logical qubits can be encoded?

For multiple channels (denoted by $\alpha$) on a given qubit, the expressions in this previous work can easily be generalized. We are assuming that the time scale of correction is fast compared to the time scale of decoherence; therefore, different errors do not interfere with one another, and all the expressions in our paper behave well (*i.e.*, linearly). We should also note here that implicit in the idea that all errors are *detected* is the assumption

that, therefore, given such a detection we know not only when and where ($j$) the error has occurred, but also what the error is ($\alpha$). In other words, given a detection we can determine the error Kraus operator $\Omega_j^\alpha$ that has been applied.

Given the above assumptions, to generalize to multiple-channel protocols we must merely check whether for all $\alpha$ and $j$ it is true that

$$\langle\psi_i|D^{(j),\alpha}|\psi_k\rangle = 0. \tag{6.49}$$

If (6.49) holds, the corresponding errors $E^{(j),\alpha}$ will be correctable, and we will see that this condition also makes it possible to find a driving Hamiltonian such that the no-jump errors are also corrected.

Let us first consider the case when there are two channels on a single qubit: $\alpha = 1, 2$. When there are two channels, a Bloch-sphere analysis shows that it is possible to find a single $S$ such that $\{S, D^{(j),\alpha}\} = 0$. Let us consider qubit 1: since $D^{(1),1}$ and $D^{(1),2}$ are traceless, they can be represented by two vectors on the Bloch sphere. In fact, $D^{(1),1}$ and $D^{(1),2}$ define a plane intersecting the Bloch sphere; now we pick $s^{(1)}$ to be the operator that corresponds to the vector on the Bloch sphere that is orthogonal to that plane. Since it is possible to find a unitary rotation that takes $s^{(1)}$ to $\sigma_Z$ as well as $D^{(1),1}$ and $D^{(1),2}$ to linear combinations of $\sigma_X$ and $\sigma_Y$, this operator must anticommute with $D^{(1),1}$ and $D^{(1),2}$. Doing the same for the other physical qubits, we pick the single stabilizer generator

$$S = s^{(1)} \otimes \cdots \otimes s^{(n)} \tag{6.50}$$

so that the stabilizer group is $\{1, S\}$ as before. Again, this procedure encodes $n - 1$ qubits in $n$.

The next step is to consider three channels. Unfortunately, for three channels on a single qubit, it is not in general possible to find a single $s$ which anticommutes with all the $D$ operators of the channels; this is reflected by the fact that the Bloch sphere is three-dimensional, and so given three arbitrary vectors, it is not possible in general to find a fourth vector perpendicular to all three.

However, we can do almost as well. Let us return to (6.49) again. In fact for (6.49) to

be true, it suffices to decompose any given error operator, $D$, as $D = \vec{d} \cdot \vec{\sigma}$ and to require

$$\langle \psi_i | d_l \sigma_l | \psi_k \rangle = 0 \; \forall \; l. \tag{6.51}$$

If our stabilizers are the two stabilizers of the familiar four-qubit code for the erasure channel [39],

$$\begin{aligned} S_1 &= XXXX, \\ S_2 &= ZZZZ, \end{aligned} \tag{6.52}$$

we can see that for any $l$ one of these two, call it $S_{j(l)}$, will satisfy

$$\{S_{j(l)}, \sigma_l\} = 0 \tag{6.53}$$

no matter what $D$ is, and thus (6.51) holds.

In this case, with $a = 1 + O(dt)$ as before, we have

$$\Omega_0 = a1 - \frac{D}{2}dt - \frac{\gamma}{2}(e^{-i\phi}E - e^{i\phi}E^\dagger)dt - iHdt. \tag{6.54}$$

Let

$$H = \sum_l \frac{i}{2}(d_l \sigma_l) S_{j(l)} + \frac{i\gamma}{2}(e^{-i\phi}E - e^{i\phi}E^\dagger), \tag{6.55}$$

where $S_{j(l)}$ is defined as in (6.53). Then

$$\Omega_0 = a1 - \frac{1}{2}\sum_l d_l \sigma_l (1 - S_{j(l)}), \tag{6.56}$$

which leaves the codespace invariant. This analysis is true for each additional error channel we introduce. Thus no matter how many error channels there are, as long as we can detect all of them and know which error has happened and where the error has happened, we can correct for the error and the no-error evolution. This code encodes two logical qubits in four physical ones.

In fact, this reasoning applies for $n$ qubits, where $n$ is even, given the two stabilizers

$$\begin{aligned} S_1 &= X^{\otimes n} \\ S_2 &= Z^{\otimes n}. \end{aligned} \tag{6.57}$$

Using these stabilizers with the constant Hamiltonian found above, it is possible to encode $n-2$ qubits in $n$.

This protocol, of course, borrows heavily from the stabilizer formalism of the quantum erasure code. Indeed, the quantum erasure code can be generalized using the stabilizers in (6.57) in the same way, with the same scaling of $n-2$ logical qubits in $n$ physical ones; as far as we know this scaling has not been explicitly noted in the literature. On the other hand, our protocol differs from the erasure code in that we have made a different and more restrictive assumption about the error model; as a result, we only need to perform local measurements instead of a highly nonlocal stabilizer measurement. To elaborate, the quantum erasure code makes the same assumption that the position and time of the error are both known. In the protocol given here, we make the further assumption that we know *what error* has occurred in that measuring the error tells us what error has occurred. This information about the error comes precisely from the detection of the local measurements performed by the environment. Again, these results indicate that if dominant error processes can be monitored, using that information can be the key to correcting them, and that the overhead in encoding is minimal (just two physical qubits).

## 6.6 Looking at a spin 1 system

The protocols in this chapter have been heavily dependent on the use of a two-state quantum system. What happens when we consider embedding the two-dimensional Hilbert space of a qubit in a spin $J$ state? A realistic quantum architecture may well make use of such spin $J$ systems in which the operators $e^{-i\hat{n}\cdot\hat{J}}$ are the control operations that can be applied to the system. Furthermore, understanding this case may help in applying this protocol to coherent-state computing [73].

Unfortunately, for most values of $J$ and for coherent states I do not know how to design a suitable protocol. In this section I will sketch a protocol for $J = 1$. In this section alone

I will *not* be using the usual stabilizer notation, in the sense that when I have a tensor product I will *always* explicitly write $X \otimes X$ and not $XX$.

A first guess might be to let the states of the qubit be $J_z$ eigenstates, *e.g.*,

$$\begin{aligned} |0_L\rangle &= |1,1\rangle_z \\ |1_L\rangle &= |1,0\rangle_z \\ |E\rangle &= |1,-1\rangle_z, \end{aligned} \tag{6.58}$$

where $|E\rangle$ will be some leakage (error) state. Since the no-jump evolution is given by the operator $J_+J_- = J^2 - J_z^2$, and of course the above are eigenvalues of both $J^2$ and $J_z^2$, the codespace is fixed by the no-jump evolution (one might think of it as a sort of decoherence-free subspace under that evolution). The $J_-$ errors can, therefore, be simply fixed by doing a $J_+$ operator.

Unfortunately this is a bit *too* simple to be true, as $J_+$ operators do not fall in the category of operators we are allowed to perform. Instead let the states of the qubit be given by $J_x$ eigenstates (one might also motivate this encoding as an analogy to the "cat" states in coherent-state encoding [73]). It turns out as well that matters are better if we use the $|\pm_L\rangle$ notation, that is,

$$\begin{aligned} |+_L\rangle &= |1,1\rangle_x + |1,-1\rangle_x \\ |-_L\rangle &= |1,1\rangle_x - |1,-1\rangle_x \\ |E\rangle &= |1,0\rangle_x, \end{aligned} \tag{6.59}$$

where $|\pm_L\rangle$ will be the logical states, and again $|E\rangle$ will be some leakage (error) state.

The nice thing about this encoding is that processing turns out to be (relatively) nice: $J_x$ acts simply on the two-dimensional space as the operation $X_{qubit}$ (and annihilates the error state), and $Z_{qubit}$ is also somewhat simple: $Z_{qubit} = 1 - 2J_y^2$.

This allows us to decompose the no-jump error $J_+J_-$ as

$$J_+J_- = J_x^2 + J_y^2 + 2J_z. \tag{6.60}$$

The first term, $J_x^2$, obviously acts as the identity on the two-dimensional space, so we do not

need to worry about that term further. The third term, $2J_z$, cannot be canceled directly by a $J_z$ term in the Hamiltonian (recall that this would give a non-Hermitian and therefore dissipative evolution), but fortunately *on the two-dimensional space* it can be canceled by $H_1 = J_x J_y + J_y J_x$.

The second term is a bit of a problem, but now we can use the same trick used above for detected errors. If we encode the state in the same way as before, such that

$$\begin{aligned} |\tilde{0}\rangle &= |+_L\rangle|+_L\rangle + |-_L\rangle|-_L\rangle \\ |\tilde{1}\rangle &= |+_L\rangle|-_L\rangle + |+_L\rangle|-_L\rangle, \end{aligned} \tag{6.61}$$

the codespace given by $\{|\tilde{0}\rangle, |\tilde{1}\rangle$ is stabilized by $J_x \otimes J_x = X_{qubit} \otimes X_{qubit}$.

So now (assuming for now that the $J_+ J_-$ evolution is happening on the first qubit; can be fairly easily generalized if we get this to work) we define

$$H_2 = -iDS = -iZ_{qubit}X_{qubit} \otimes X_{qubit} = (1 - 2J_y^2)J_x \otimes J_x = J_x \otimes J_x - 2J_y^2 J_x \otimes J_x. \tag{6.62}$$

And now we just apply the total Hamiltonian $H = H_1 + H_2$.

The jump evolution is similarly easy to correct; if we are in the codespace to begin with, it turns out that the effect of $J_-$ is the following:

$$J_-(\alpha|+_L\rangle + \beta|-_L\rangle) = \beta|E\rangle + \beta|+_L\rangle + \alpha|-_L\rangle \tag{6.63}$$

Since, as we remarked before, $J_x$ acts as $X$ on the $|\pm_L\rangle$ basis and annihilates the error state, $J_x$ is the operator that corrects for this error.

This nicely solves the problem of being able to do all the necessary operations. Unfortunately, in a real physical system, performing complicated operators like $Jy^2 J_x \otimes J_x$ may be quite difficult. So this procedure, although theoretically elegant, has a fundamental drawback as well. Perhaps it is possible to decompose the $J_+ J_-$ term in a different way that leads to a protocol that retains the elegance but is easier to perform.

## 6.7 Conclusion

It is possible to understand a particular variant of quantum control as quantum error correction. This method is very general in that it can correct any single qubit detected errors, while requiring only $n$ physical qubits to encode $n-1$ logical qubits. As a particular example, we have shown how to correct for spontaneous emission evolution using feedback and a driving Hamiltonian, which allows less redundancy than has previously been obtained. We have additionally shown that universal quantum computation is possible under our method and generalized the method for multiple error channels. Perhaps this work will provide a starting point for practically implementable feedback schemes to protect unknown states. The fact that only two qubits are required for a demonstration should be particularly appealing.

This work, together with the work from previous chapters, suggests that there are certain tradeoffs present in all these protocols. In the last three chapters we have seen three protocols, all of which have different strengths and weaknesses. The protocol given in Chapter 4 was able to perform inefficiency-robust optimal control, but at the expense of a large amount of calculational power. The protocol in Chapter 5 retained the property of robustness with much less post-processing, but at the expense of only working when the measurement and feedback strengths were much stronger than that of decoherence. In contrast, the protocol given here requires no post-processing at all, but it cannot correct arbitrary errors and is not robust to measurement inefficiency. Is there a nice way of characterizing the tradeoffs for these different protocols? Perhaps if that question could be answered, one could construct a whole range of intermediate protocols that could be tailored for the specific requirements of a given physical system.

# Chapter 7

# Cost of quantum fault tolerance

## 7.1 Towards Pippenger's conjecture

In Chapters 4-6 we explored the consequences of changing the physical tools available to perform quantum error correction. We now ask what happens when the physical tools remain projective measurements and unitary gates, but the structure of the code becomes different: information is encoded in topological degrees of freedom. Here we shall apply these toric codes to the problem of calculating the cost of quantum fault tolerance.

We saw in Chapter 1 that Pippenger has conjectured that the quantum blowup to produce a fault-tolerant circuit is given by

$$\begin{aligned} L^* &= O(L \log^2 L) \ , \\ D^* &= O(D) \ . \end{aligned} \tag{7.1}$$

In this chapter, we will use topological coding methods to prove the following:

**Theorem 1** *A quantum circuit of size $L$ and depth $D$ can be accurately simulated by a circuit of noisy quantum gates, provided the noise is sufficiently weak, with blowup in size and depth*

$$\begin{aligned} L^* &= O(L \log^{3+\alpha} L) \ , \\ D^* &= O(D \cdot \log \log L) \ , \end{aligned} \tag{7.2}$$

for any positive $\alpha$.

As for the classical analysis, in Theorem 1 we are assuming that a quantum gate can be executed acting on any pair of qubits with a fixed fidelity. Geometry is ignored — we won't worry about how the qubits are arranged in space. In the quantum case, we also assume that an inexhaustible supply of ancilla qubits (with fixed fidelity) is available; since quantum gates are reversible, ancilla qubits are needed to provide a sink for dumping the entropy introduced by errors.

An interesting alternative computational model is one in which qubits can be measured, and classical postprocessing of measurement outcomes that is polylogarithmic in $L$ is assumed to be instantaneous. In this model, the blow up in *quantum* depth and size can be further reduced:

**Theorem 2** *A quantum circuit of size $L$ and depth $D$ can be accurately simulated by a circuit of noisy quantum gates, measurements, and reliable classical gates. If classical postprocessing of measurement outcomes that is polylogarithmic in $L$ is regarded as instantaneous, then provided that noise in the quantum gates is sufficiently weak, the* quantum *blowup in size and depth is*

$$\begin{aligned} L^*_{qu} &= O(L \log^{2+\beta} L) \ , \\ D^*_{qu} &= O(D) \end{aligned} \tag{7.3}$$

for any positive $\beta$.

Here we prove this result using a topological coding method, improving the power of $\log L$ in the blowup of the size over previous results. Thus in this model, we can come arbitrarily close to realizing the blowup envisioned by Pippenger.

The chapter is organized as follows: Section 7.2 gives some background on the classical version of the topological code being used here, and uses formalism from the classical case to prove an important lemma about failure rates for encoded gates. Section 7.3 uses the lemma to prove Theorem 1 above, and section 7.4 proves Theorem 2. Section 7.5 concludes.

## 7.2 The probability of failure for an encoded gate

Chapter 2 describes a quantum memory, discussed in more detail in [16], in which qubits are arranged in a hypercubic lattice on a four-dimensional torus, such that (six) encoded qubits can be protected from storage errors through the action of noisy local gates. It was a source of embarrassment in [16] that four dimensions were required to allow all gates to act locally, but now we are disregarding geometry and need not be apologetic. This four-dimensional quantum memory based on toric coding [48, 44] is a natural quantum generalization of a two-dimensional classical memory based on repetition coding, which can be stabilized by local gates as described by Toom [83]. In this classical system, each bit is encoded in a repetition code on a two-dimensional torus with side length $m$: that is, the initial state of the system is either all zeros or all ones. This state can be preserved under the application of a simple local cellular automaton transition rule at each (discrete) time step, even in the presence of noise at each time step, if the amount of noise is small enough. In particular, let us define a spacetime cell to be in error if its value obtained from the noiseless evolution (*e.g.*, the value 0 for the encoded bit 0) differs from the actual value (*e.g.*, the cell has value 1 when the encoded bit is 0). Under the action of the transition rule at every time step, even if the rule is applied with error $\epsilon$ every time, as the size of the torus gets very large, at arbitrary time the probability of any one cell being in error is still at most a constant factor times $\epsilon$.

Such a simple transition rule that preserves the state under noise on a two-dimensional lattice is given by the following procedure. For a given cell $p$, draw a string between $p$ and its neighbor if at the current time step the value of the neighbor is different than $p$. If $p$ has string on both its north and east edges, flip the value of $p$ in the next time step (from 0 to 1 or from 1 to 0); otherwise, the value of $p$ will remain the same. This rule is called Toom's rule after A. Toom, who was the first to give a proof of the fault-tolerance of this rule [83]. The intuition behind the rule comes from the observation that given a finite (error) droplet of 1's in a sea of 0's (or vice versa, if a 1 is to be protected), this rule will invariably eat away at the droplet in a time that is roughly proportional to the linear boundary of the droplet. Note that such a droplet will have its boundary delimited by string.

In the proofs of Theorems 1 and 2, it will be important to show that given qubits

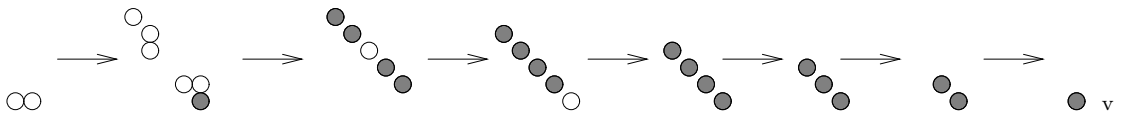



Figure 7.1: Here is shown a possible error path for the cell $v$ having been in error. White cells are in Noise, while grey circles are in error but not in Noise.

encoded in the four-dimensional topological coding method in lattices of linear size $s$, the probability of error in an encoded gate goes like $\exp[-O(s)]$. In order to do so, we shall borrow formalism from a proof of the fault-tolerance of Toom's rule due to Gacs [30].

### 7.2.1 An overview of Gacs' proof of the fault-tolerance of Toom's rule

The basic idea of Ref. [30] is the following: given a particular cell, in order to find the probability that it is in error, we must add up all the probabilities corresponding to the different ways the cell could have come to be in error. Let us define Noise as the set of cells in spacetime that did not obey the transition rule at that time step. Then one way the cell could have been in error is to simply have been in Noise at the current time step, with probability $\epsilon$. Another possible route could have been two neighboring cells in Noise in the preceding time step that evolved correctly (obeyed the transition rule) in the current time step to result in the current erroneous cell, with probability $\epsilon^2$. A more complicated route is shown in Fig. 7.1.

Counting all of these possibilities is the main subject of [30]; to do so, Gacs associates each of these error "paths" in spacetime with what he calls an *Explanation Tree.* Cells in error are the nodes of these Explanation Trees. A given node is connected by an edge only to cells that are adjacent to it in space and/or time. Edges connecting cells adjacent in space but with the same time coordinate will be called *forks*, and edges connecting cells with different time coordinates will be called *arrows.* These trees are constructed in such a way that the number of possible "Explanations" with $n$ points in Noise is no more than exponential in $n$. This construction is made possible with two main lemmas from [30]:

**Explanation Tree Lemma**: Let $u$ be a cell in error but outside Noise. Then there is an Explanation Tree $Expl$ rooted at $u$ such that if $n$ nodes of $Expl$ belong to Noise then the number of edges of $Expl$ is at most $4n - 4$.

The second lemma is useful because the construction of the above trees implies a maximum node degree:

**Tree Counting Lemma**: In a graph of maximum node degree $r$, the number of weighted[1] subtrees rooted at a given node and having $k$ edges is $2r \cdot (2r^2)^k$.

Since the probability of $n$ points being in Noise goes as $\epsilon^n$, adding up all the probabilities associated with those Explanation Trees results in a converging geometric series, as long as $\epsilon$ is below a particular threshold value. Explicitly, for some cell $v$,

$$\begin{aligned} Prob(v \text{ in error}) &\leq \sum_{n=m}^{\infty} |E_n| \epsilon^n \\ &\leq \frac{2r}{16r^8} \sum_{n=1}^{\infty} (16r^8\epsilon)^n = \frac{2r\epsilon}{1-16r^8\epsilon}. \end{aligned} \tag{7.4}$$

where $|E_n|$ denotes the number of trees with $n$ edges.

The above is true for a system acting as a quantum memory that is on an infinite lattice. Changing the proof to accomodate finite-size lattices is simple: replacing the plane by the torus causes an error no worse than the probability that the Explanation Tree has more nodes than the linear size of the torus, which is exponentially small. Moreover, incorporating computation in these models is fairly simple. In the classical case, computation involving two encoded bits is done by encoding each bit in a two-dimensional lattice and then performing the computation between each pair of corresponding bits. This complication does nothing more to the proof by Gacs than add another dimension to the formalism and add to the maximum degree $r$ of each node. Gacs' theorem becomes the following: For error rate $\epsilon$ below a certain threshold value and finite lattice size $m$,

$$Prob(v \text{ in error}) \leq 24tm^2N(2 \cdot (144)\epsilon^{1/12})^m + 24\epsilon, \tag{7.5}$$

where $N$ is the size of the computational register and $t$ is the time taken to do the calculation. The second term comes from the analysis in the previous paragraphs, and the first term is the correction for finite-size lattices.

[1]The term "weighted" refers to an extra bit associated with each node; it is a technicality of the proof that will not be further discussed here, as it carries over without change in the toric code case. For elaboration see [30].

The proof of the Tree Counting Lemma is fairly straightforward and simply involves counting the number of ways the tree can be formed. The Explanation Tree Lemma, however, is rather more convoluted in its construction of Explanation Trees.

The Explanation Tree in spacetime is built by considering "excuses" for each point in error. That is, if a spacetime cell $P = (a, b, t)$ is in error but not in Noise, there must have been at least two points in the preceding time step that were in error and that resulted in the cell $P$ being in error. In particular, at least two out of the three elements of the set $\{(a, b, t-1), (a, b+1, t-1), (a+1, b, t-1)$ must be in error. Choose two elements that are in error, and denote this two-element set as $Excuse(P)$. (This is, of course, not necessarily a unique choice, but this turns out not to matter.) A preliminary *investigation graph* for a particular error path is constructed by connecting each cell $P$ in error (if it is not in Noise) to the cells in $Excuse(P)$ by arrows, and for each $P$ connecting the two cells in $Excuse(P)$ to each other by forks. Fig. 7.2 shows a sample investigation graph.

How large the graphs are in space is quantified by what Gacs calls the *span* and *size* of the tree. The span is computed by defining the following functionals: for $v = (x, y, t)$,

$$\begin{aligned} L_1(v) &= -x \\ L_2(v) &= -y \\ L_3(v) &= x + y. \end{aligned} \tag{7.6}$$

A *spanned set* $(S, v_1, v_2, v_3)$ is a set of cells $S$ together with an ordered triplet of cells (or *poles*) $(v_1, v_2, v_3), v_i \in S$, and the span of a spanned set is simply

$$Span(S, v_1, v_2, v_3) = \sum_{i=1}^{3} L_i(v_i). \tag{7.7}$$

Note that the span of a set consisting of a single point $v$ (which forces the poles to be $(v, v, v)$) is zero.

One special case of the span of $(S, v_1, v_2, v_3)$ is that in which the $v_i \in S$ are picked such that $L_i(v_i)$ is maximal over the set $S$ for all $i$. A moment's thought will convince you that

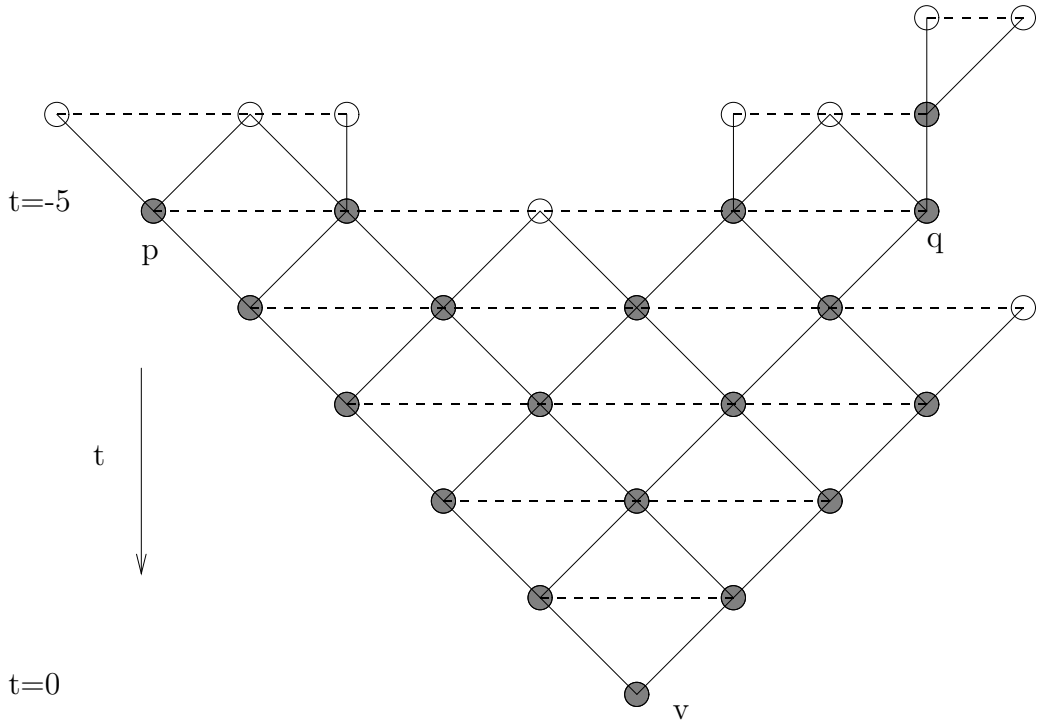



Figure 7.2: A sample investigation graph for the evolution shown in Fig. 7.1. Two droplets at $t = -5$ evolve to become the a single cell, $v$, in error at $t = 0$. Nodes are cells in error, and adjacent nodes are connected by edges. Grey circles are cells in error but not in Noise, and white circles are cells in Noise. Solid lines are arrows, and dotted lines are forks. Earlier times are higher. Cells $p$ and $q$ are in different Histories at $t = -5$ (see Fig. 7.4).

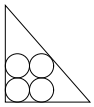


Figure 7.3: The size of the set $S$, where $S$ is four cells arranged as shown, is 2. The isoceles triangle shown has a short side of length 3.

given a set $S$ of cells in an "error droplet" at a particular time, the quantity

$$Size(S) = \sum_{i=1}^{3} \max_{v \in S} L_i(v) \tag{7.8}$$

corresponds to one less than the length of the shortest side of the smallest isoceles right triangle (with right angle at the lower left of the triangle) that covers the set $S$ . See Fig. 7.3. Additionally, given such an error droplet, applying Toom's rule perfectly will result in the next time step in an error droplet with size that is one less than it was previously. Thus given a set $S$, $Size(\cup_{v \in S} Excuse(v)) = Size(S) + 1$. This is a general principle in regard to spanned sets: if $\mathbf{S} = (S, v_1, v_2, v_3)$ is a spanned set and the $v_i$ are not in Noise, the spanned set $\mathbf{S'} = (\cup_{v \in S} Excuse(v), Excuse_1(v_1), Excuse_2(v_2), Excuse_3(v_3))$ has the property that $Span(\mathbf{S'}) = Span(\mathbf{S}) + 1$. The span, then, is a way of quantifying the observation that applying Toom's rule shrinks an error droplet.

The investigation graph is pruned through a process Gacs calls refinement. The basic idea behind the refinement procedure is that it retains all the important structural behavior both about the points in Noise necessary to the Explanation as well as about how they propagate (encoded in the concept of the span), while removing any extraneous edges or nodes of the tree.

In order to retain this structural information, more formalism is needed to address the case of two separated droplets coalescing into one droplet at a later time, as in Fig. 7.2. This situation leads to the concept of the *spanned History*: given a spacetime cell $p$ with time coordinate $T$, consider the graph formed by deleting all forks and all parts of the investigation graph with time coordinate greater than $T$. The connected part of the resulting graph containing $p$ is $History(p)$, and the time coordinate of the History, $Time(History(p))$, is defined to be $T$. A spanned History is an object $(History(p), p_1, p_2, p_3)$, where $p_1, p_2, p_3$

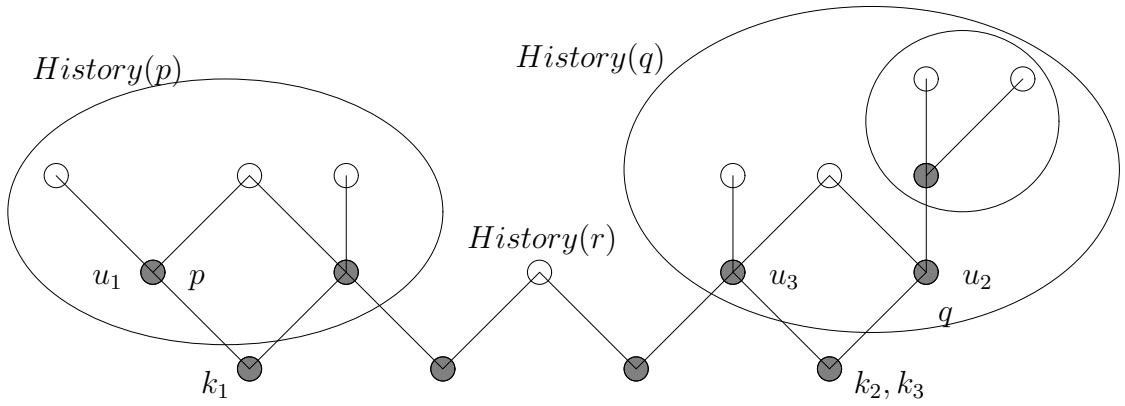



Figure 7.4: A spanned History $K$ (one of the two Histories associated with $t = -4$ in the investigation graph in Fig. 7.2; the other is the single point at the right) with poles $(u_1, u_2, u_3)$ and with all sub-Histories drawn. Here we can say $M_1 = History(p), M_2 = History(r) = r, M_3 = History(q)$. White circles denote Histories, including the one-cell Histories that are points in Noise. $p$ and $u_1$ refer to the same cell, as do the pair $k_2, k_3$ and the pair $q, u_2$.

are three (not necessarily distinct) poles contained in $History(p)$ with time coordinate $T$. These poles, again, provide a way to keep track of the spatial size of the Histories. Fig. 7.2 gives an investigation graph in which the points $p$ and $q$ are in different Histories, and Fig. 7.4 gives a subset of the graph on which all Histories are marked.

The refinement operation proceeds as follows. We start with a node $v$ in error but not in Noise; this node has a nontrivial History. We pick the three poles associated with this History to be $(v, v, v)$. We will proceed recursively by considering each spanned History at each time $T$, moving backwards in time. Given a particular spanned History $K$ with poles that we will denote as $k_1, k_2, k_3$, consider all the distinct (sub-)Histories $M_j$ that can be formed with $Time(M_j) = Time(K) - 1 = T - 1$. (These sub-Histories $M_j$ will all be subsets of $K$ and will partition the cells in $K$ with time coordinate less than $T$. See Fig. 7.4). Delete any existing forks with time coordinate $T - 1$ as well as arrows connecting cells with time coordinate $T - 1$ to cells with time coordinate $T$. Now define the points $u_i = Excuse_i(k_i)$. (Note that $\sum_i L_i(u_i) = Span(K, k_1, k_2, k_3) + 1$, as was discussed earlier.) The only edges added back in are the (at most) three arrows connecting $k_i$ to $u_i$, and forks connecting separate histories at time $T - 1$ together. Gacs has proved a "Stokes" theorem

that says, when applied to this situation, that we can discard extraneous sub-Histories in a way such that the remaining sub-Histories $M_{j'}$ (now indexed by $j'$ instead of $j$ to account for the fact that there are fewer of them) can be turned into spanned Histories by associating poles to each of them. The $u_i$ give some of the poles; the others are given by the endpoints of the forks connecting the Histories together. Furthermore, the theorem states, this construction has the property that $\sum_{j'} span(M_{j'}) + \text{number of forks} = \sum_i L_i(u_i)$. This means that given this construction, we can consistently define the span of the tree at a given time $T-1$ to be the span of all the histories $M$ such that $Time(M) = T-1$ plus the number of forks connecting histories together in the tree. Also, since we saw that $\sum_i L_i(u_i) = Span(K, k_1, k_2, k_3) + 1$, one step of the refinement has the effect of adding at most three arrows and raising the span by one.

An example of this refinement procedure is given in Fig. 7.4. Here, poles $p_1, p_2, p_3$ and $q_1, q_2, q_3$ for $History(p)$ and $History(q)$ respectively can be chosen from the $u_i$ and the fork endpoints in such a way that $Span(History(p)) = 1$ and $Span(History(q)) = 1$. Note that there will be two forks drawn in this example to connect $History(p)$ to $r$ and $History(q)$ to $r$; then

$$\begin{aligned} Span(K, k_1, k_2, k_3) + 1 &= 3 + 1 \\ &= Span(History(p) \cup History(q) \cup History(r), u_1, u_2, u_3) \\ &= Span(History(p)) + Span(History(q)) + Span(History(r)) \\ &\quad +\# \text{ forks} \\ &= 1 + 1 + 0 + 2 \end{aligned} \tag{7.9}$$

It is possible to prove via inductive reasoning that after carrying out this refinement procedure for each spanned History at each time, the resulting Explanation Tree is connected. When the tree has been wholly refined the only Histories left will be single points in Noise with span 0, so the number of forks is equal to the span, which is equal to the total number of points in Noise minus one (because any extra forks would have been removed during refinement). Fig. 7.5 shows the Explanation Tree resulting from the investigation graph given in Fig. 7.2.

This removal of extraneous edges and nodes is what allows the tree size to remain small

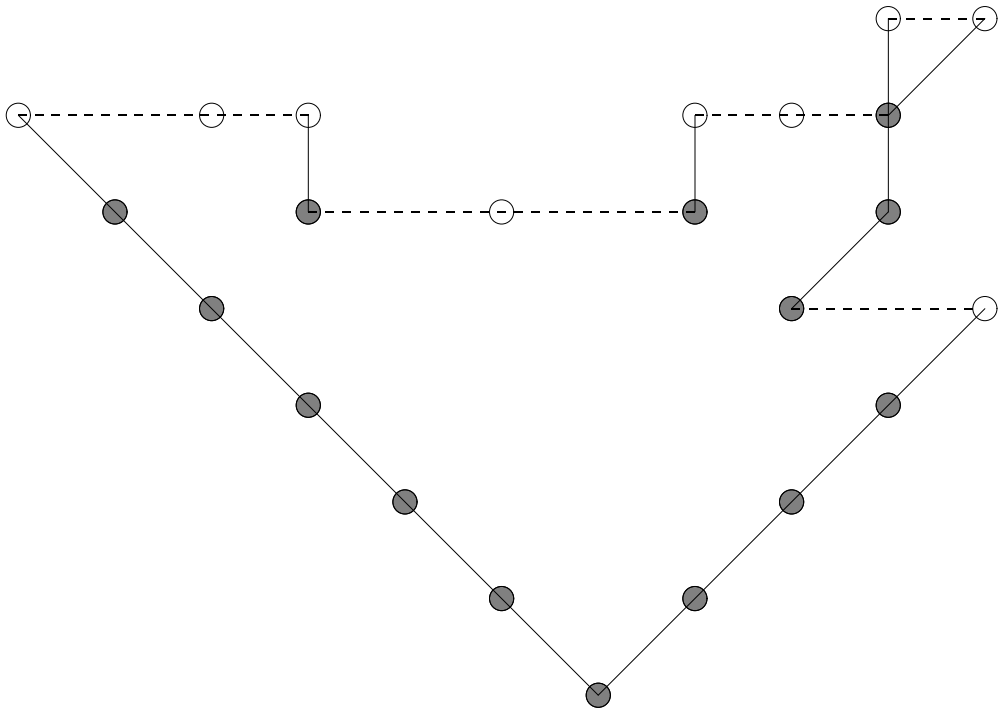


Figure 7.5: The Explanation Tree associated with the investigation graph in Fig. 7.2. White circles denote points in Noise, and grey points denote points in error but not in Noise. Note that the number of horizontal edges is one less than the number of points in Noise.

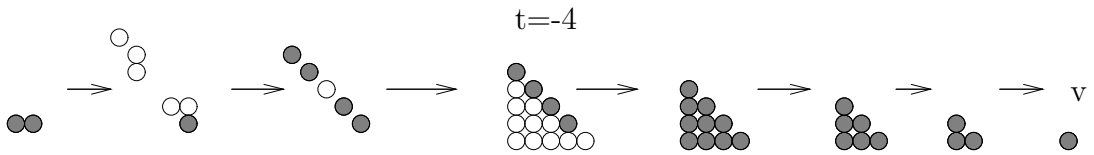



Figure 7.6: Another error path that has the same Explanation Tree as that in Fig. 7.1.

and is intimately tied to the concepts of the span and size, which are extremely important in Gacs' arguments. In his construction of the explanation trees in [30], as we have seen, he only considers trees in which the number of cells in Noise ($k$) is almost the *same* as the span of the explanation tree ($k-1$).

The subtleties of discarding extraneous trees are discarded may be better understood by means of an example. Let us consider the error path shown in Fig. 7.6. This path leads to $v$ being in error; however, there are many more cells in Noise in this path than in that shown in Fig. 7.1. However, when the Explanation Tree for this error path is drawn, it looks (or can look; there is some ambiguity in making the tree) exactly like the Explanation Tree given in Fig. 7.5. Essentially, the intuition is that Toom's rule does not care about any of the cells in that triangle except the cells shown in Fig. 7.1.

Even though configurations such as Fig. 7.6 are not explicitly counted, the probability counting actually includes *all* explanations. Gacs is counting sets of the form Fig. 7.1 as Explanations with the important stipulation that one does not *know* whether or not the interior cells at *e.g.* $t=-4$ are in Noise or not (hence the sum in Eq. (7.4) has a term $\epsilon^n$ instead of, say, $\epsilon^n(1-\epsilon)^m$). Therefore, Gacs is actually also counting sets of the form Fig. 7.6. In fact, any given cell in the interior of the triangle at $t=-4$ in Fig. 7.6 could be in Noise (with probability $\epsilon$) or not (with probability $1-\epsilon$). Since those two probabilities sum to 1, Gacs needs not count Fig. 7.6 separately. In general, the intuition behind Gacs' argument tells us that given a set with a certain *Size*, we require a cell in Noise to have given rise to each element of the *Size*, and any other cells can be ignored.

The preceding procedure, of course, is only a heuristic outline. Gacs' description is rigorous, and the reader who would like to understand the details of the construction is referred to [30].

### 7.2.2 Suppression of error droplets

Before applying the formalism to the four-dimensional quantum case, let us consider a variant of Gacs' analysis: Since the toric code fails when large enough error droplets are formed, we wish to compute the probability that a droplet of $Size\ m$ (under the definition of $Size$ given above) is present at a given time. In particular, we will show that the probability of getting a droplet error in a two-dimensional CA system that is subject to Toom's rule at every time step is suppressed exponentially by the $Size$ of the droplet, that is, roughly by the length of the boundary of the droplet.

The crucial idea of the proof is that the droplet itself can be regarded as the Explanation for a single-cell error with no additional Noise. The intuition behind this claim is the simple observation that after enough applications of Toom's rule, a connected droplet will evolve into a single cell; furthermore, at all time steps until then the evolution of the droplet will result in another connected droplet. (Formally, the statement is the following: Take the investigation graph generated by applying Toom's rule noiselessly to the droplet until it becomes a single cell. Delete all forks. For all $t < 0$, deleting all areas connecting cells at all times $t' < t$ should result in a graph that is a single History.) We will call the tree representing the decay of the droplet to a single cell the "original tree" and define the times $0, T$ by saying that the tree encompasses time 0 to time $T$. The idea is that we can construct a Gacs-like tree by imagining a model in which we turn off the noise after time 0.

Now we consider an explanation tree that is a concatenation of the "Original tree" and some Explanation of the droplet. We will assume that this Explanation encompasses time $T$ to time $-T'$; it is, of course, also a valid Explanation Tree for the single-cell root of the original tree. We will call this tree the "whole tree." Now consider deleting all parts of the whole tree between times 1 to $T$, so that the remaining parts encompass times 0 to $-T'$. We will call this construction the "truncated tree." The goal will be to sum over all the truncated trees, keeping the original tree fixed. In this way all Explanations of the droplet will be considered. In Fig. 7.7 and Fig. 7.8, examples of these terms are shown for a droplet consisting of a four-by-four triangle; the error path shown is the same as that given in Fig. 7.6.

The Tree Counting Lemma still holds here: until time 0 there has been only one history, and the construction in Gacs' paper ensures that histories are always connected to one

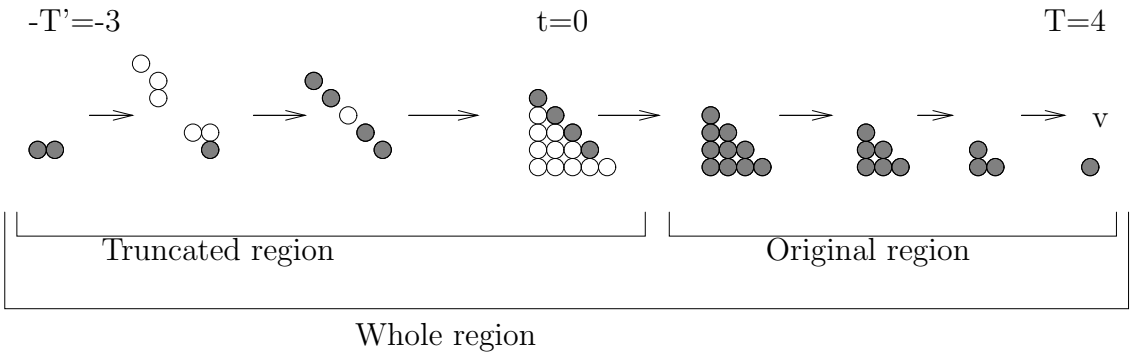



Figure 7.7: An error path for a droplet at $t = 0$, analogous to that given for a single cell in Fig. 7.6. Note that the time labels are different from those in Fig. 7.6.

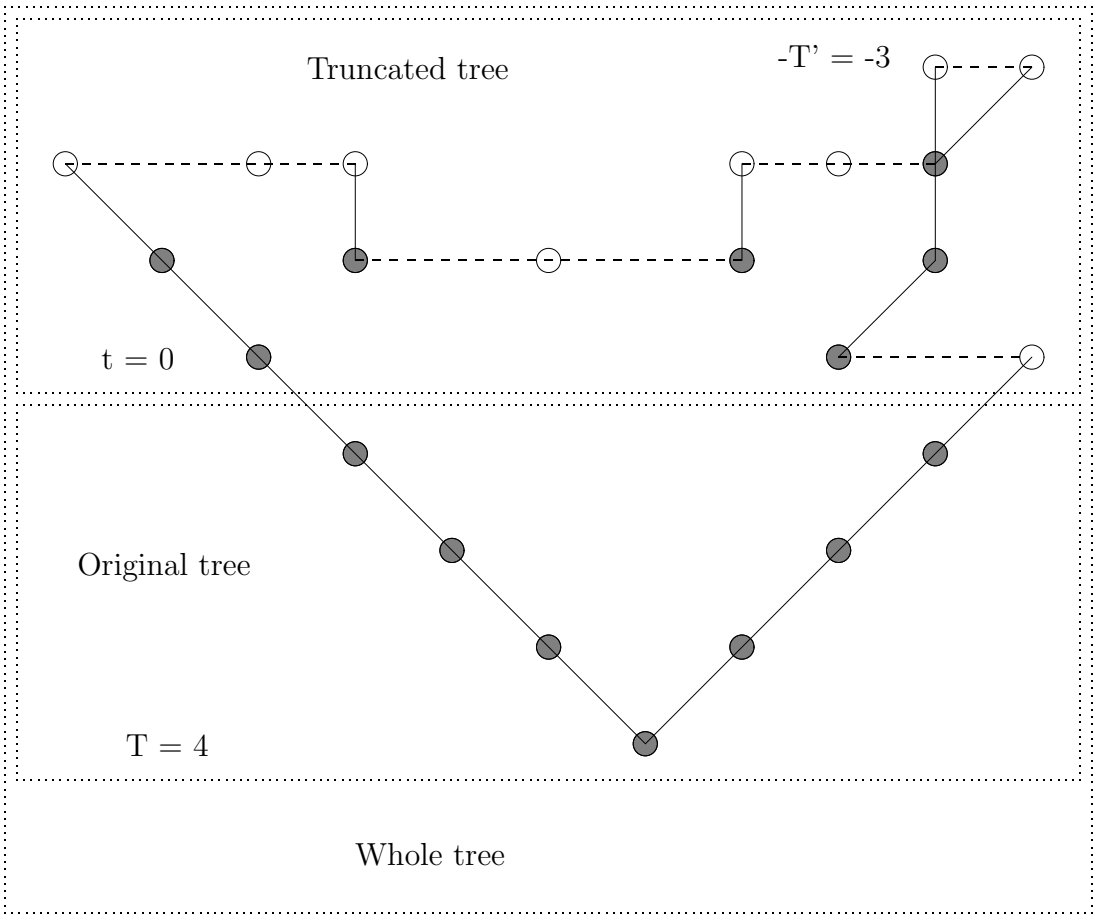



Figure 7.8: The whole, original, and truncated trees for the error path in Fig. 7.7.

another. Then it follows that the truncated tree will still be connected, and his Tree Counting Lemma holds for any connected graph.

Let $\kappa$ be the number of cells in Noise and $\gamma$ be the number of edges of the whole tree. Then we know by Gacs' construction that $\gamma \leq 4(\kappa - 1)$. We also know, using Gacs' construction, that if the Original tree has size $s$, the number of edges of the Original tree is at least $a \geq s$, as at least one edge is introduced every time the span is increased by 1.

Now let $\kappa, g'$ be the number of cells in Noise and edges of the truncated tree, respectively. (Since the whole tree is constructed such that there are no cells in Noise before time 0, the number of cells in Noise is the same for the whole and truncated trees.) Then

$$\begin{aligned} g' &= \gamma - a & (7.10) \\ &\leq 4(\kappa - 1) - s. & (7.11) \end{aligned}$$

Again,

$$Prob(\text{droplet of size } s) \leq \sum_{n=m}^{\infty} |E_n| \epsilon^n, \tag{7.12}$$

where $|E_n|$ is the set of possible explanation trees with $n$ cells in Noise, and $m$ is the smallest number of errors that could result in the droplet. (Note that $m = 1$ in Gacs.)

If the maximum node degree is $r$ ($r = 12$ for 2 dimensions) and the error rate is less than a certain threshold, the Tree Counting Lemma then implies

$$|E_n| \leq 2r(2r^2)^{4n-4}, \tag{7.13}$$

so

$$Prob(\text{droplet}) \leq (2r^2)^{-s} \frac{2r}{16r^8} \sum_{n=m}^{\infty} (16r^8\epsilon)^n = \frac{(16r^8\epsilon)^m}{8r^7(1 - 16r^8\epsilon)} (2r^2)^{-s}. \tag{7.14}$$

The trick now is to figure out $m$. One might believe that $m$ would be equal to the *area* of the droplet, but this is not correct. In fact it is easy to demonstrate an example in which $4s$ cells in Noise suffice to build a connected droplet of boundary $s$ (and thus area $\sim s^2$) [62]. Careful reading of Gacs' arguments shows that $m$ must be at least $s$. Here is perhaps a more intuitive way of seeing that: For a droplet with boundary $s$, consider an arbitrary cell on the North side of such a droplet. Either this cell can be in error through

itself being in Noise, or else by evolution from other errors. Now, the second choice is not valid: For a cell to be in error by evolution, without having started out in Noise, both its north and east neighbors must be in error. It is possible for the east cell to have become in error in previous time steps, but since we are at the north boundary of the droplet, by definition there is no way that something "more northward" could have become in error without another independent error. The same analysis holds for the east side of the droplet.

Then we have

$$Prob(\text{droplet of size } s) \leq \frac{1}{8r^7(1-16r^8\epsilon)(2r^2)^s}(16r^8\epsilon)^s. \tag{7.15}$$

We can see that in two dimensions, the suppression of error droplets goes as $\exp[-O(s)]$.

Similarly, in $d$ dimensions, where cells are centered in $d$-dimensional solids with a $(d-1)$-dimensional boundary, we can define a similar rule. Define a positive direction for each axis. For a given $d$-dimensional solid, there are $d$ faces that have more positive coordinates than the other $d$ faces. Define string as before to be placed on faces that are shared by two cells with differing values. If there is string on all plus faces, flip the value of the cell; otherwise do nothing to that cell. The above argument for finding $m$ generalizes to $d$ dimensions. In the $d$-dimensional case, the probability of obtaining a droplet with boundary size $s^{d-1}$ becomes

$$Prob(\text{droplet}(s^{d-1}, d)) \leq \frac{1}{8r^7(1-16r^8\epsilon)(2r^2)^s}(16r^8\epsilon)^{s^{d-1}}. \tag{7.16}$$

### 7.2.3 The quantum problem: two-dimensional error droplets in four dimensions

Here we will show that much of Gacs' formalism is applicable to the quantum problem. In some ways, of course, the quantum (4D) version is more complicated, and in those cases we will show that the probability counting can still be done in a similar way, by constructing correspondences to the two-dimensional problem.

We now consider a system where spins reside on two-dimensional plaquettes in a four-dimensional lattice. Given some edge, we will measure the product of $X$'s on the spins residing on all six plaquettes which share that edge. We say there is "string" on that edge if that product is equal to $-1$. Errors on some connected set of plaquettes then manifest

as a closed loop of string. The $Z$ operators work the same way on the dual lattice, but here we will restrict ourselves to considering the $X$ operators for simplicity. This "string" description is equivalent to the description given before for the two-dimensional classical lattice, where the string resided on edges bordering two plaquettes with different values.

We now need a Toom-like deterministic rule to get rid of such loops. We propose the following simple rule: Define the *plus* and *minus* edges as in the previous section. If there is string on both of the plus sides of the plaquette, then flip the spin on the plaquette. In two dimensions this rule reduces to Toom's rule.

Now we make the appropriate changes in the Gacs definitions. The $L$ functional becomes

$$\begin{aligned} L_1(v) &= -x \\ L_2(v) &= -y \\ L_3(v) &= -z \\ L_4(v) &= -w \\ L_5(v) &= x + y + z + w, \end{aligned} \tag{7.17}$$

where now $v$ denotes a *plaquette* that is parametrized by the coordinates of its center. We use five poles; given this redefinition, it is fairly easy to see that Gacs' "Stokes" theorem (one of the ways in which he gets rid of extraneous nodes) still holds.

Let us denote a similar functional, $L^{vertex}$, that acts on *vertices* instead of on plaquettes. This new definition allows us to see see that any loop of string resulting from a "droplet" of flipped plaquettes eventually disappears through application of a noiseless Toom rule. Characterize the loop of string $S$ by the two numbers $\max_{v\in S} L_5^{vertex}(v)$ and $\sum_{i=1}^{4} \min_{v\in S}(-L_i^{vertex}(v))$. The rule will always act to decrease $\max_{v\in S} L_5(v)$ by 1 in each time step, while $\sum_{i=1}^{4} \min_{v\in S}(-L_i^{vertex}(v))$ must remain constant.

$Excuse(v)$ is now at least two members of the 7-element set given by the original plaquette $v$ and the six plaquettes that share an edge $e$ containing string with $v$ such that $e$ is a minus edge of that plaquette and a plus edge of $v$. Given a set of cells $S$ that are in error but not in Noise, the set $\cup_{v\in S} Excuses(v)$ will always have size at least one greater than $Size(S)$ (where we take the distance between the center of one plaquette and a neighboring one in the same plane to be two). Then all the other definitions follow from the ones I have

already given (with a different maximum node degree, of course). Many of Gacs' arguments follow directly; however, there are several issues that must be addressed.

One potential problem with moving the two-dimensional analysis to four dimensions comes from the fact that sometimes a droplet does not result in a single plaquette after many applications of the 4-D Toom's rule. In fact, one can show a three-dimensional example in which a connected droplet, after a suitable number of applications of the modified Toom's rule, results in a number of disconnected droplets that is quadratic in the $Size$ of the droplet. However, this is a technical problem, and the solution involves tedious manipulation of the existing formalism rather than lending any additional insight; as such, a modified counting argument fixing this problem is given in the Appendix.

A more fundamental potential problem is the fact that there are many different "error surfaces" that give rise to the same error loop. For a given loop of string at a particular time, for example, we can consider a variety of closed two-dimensional surface such that the loop divides the surface into two pieces. Either of these pieces, when considered as a set of plaquettes in Noise, could have produced that loop of string. We will call the set of plaquettes arrived at by using the Excuses shown above an "obedient" explanation, and any other Explanation possibly using a closed two-dimensional surface complement as a "disobedient" explanation.

We need to show that every disobedient explanation has a corresponding obedient explanation, and that the correspondence is such that the counting arguments retain their validity. The form of this obedient explanation is found by taking the set of loops of string corresponding to the error path in spacetime and considering their evolution by Toom's rule, keeping track of the spacetime coordinates of all the plaquettes that would be flipped in such a process. Fig. 7.9 gives an example of building an obedient Explanation.

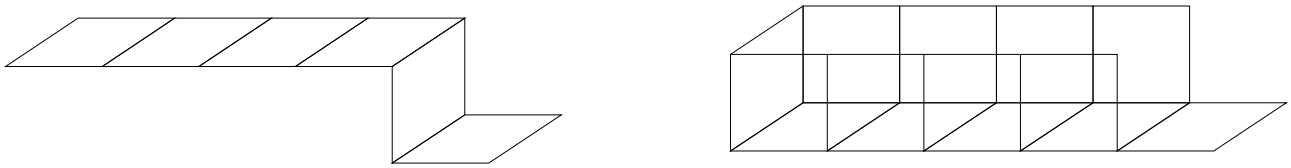
The explicit construction of the obedient Explanation is as follows:

1. Starting with the earliest time step:

   (a) Draw all the cells (in space) in the Explanation from the previous time steps

   (b) Perform Toom's rule on the loop of string at the current time step and keep track of which cells Toom's rule flips. Draw in those cells.

   (c) Erase any cell that would have string on both its plus edges if it were erased.



(a) (b)

Figure 7.9: (a) shows a set of plaquettes in Noise leading to a disobedient explanation; (b) illustrates a set of plaquettes in Noise leading to a corresponding obedient explanation found by keeping track of which plaquettes flipped through application of Toom's rule.

(d) Erase any cell that has its center more than a distance of one away from the loop. These cells will represent the points in Noise in the successive time step.

2. Repeat for all time steps in the Explanation Tree.

3. Get rid of extraneous nodes (and edges) by performing the Gacs refinement operation on the resulting investigation graph.

By construction, plaquettes in Noise are only generated in the obedient Explanation in the same time steps in which there are cells in Noise in the disobedient Explanation. Steps 1c and 1d guarantee that the only cells that remain in the obedient explanation at a given time step share at least one edge with the loop of string at that time step. As a consequence, cells in Noise in the obedient explanation share at least one edge with at least one cell in Noise in the disobedient explanation.

The number of disobedient explanations associated with a given obedient one is thus given by the number of ways that cells in Noise arranged in a disobedient way could mimic the effect of the obedient explanation. Since each additional cell in Noise adds at most three links to the error loop of string and extends the span by at least one, the problem reduces to finding the ways that a disobedient set of cells could give rise to those links. The disobedient error surface could hypothetically be quite large; however, as we saw in 7.2.1, Toom's rule only cares about the string and not about the parts of the surface that do not contribute to the string. Therefore, we only have to consider the cells adjacent to the relevant links. In $d$ dimensions (with $(d/2)$-dimensional plaquettes), the number of plaquettes that shares an

edge with a given plaquette is $d-1$, and the number of edges of a plaquette is $d$, so the total number of disobedient explanations needed to be counted for a given obedient explanation of span $s$ can be at most $((d-1)\cdot 2(d/2-1))^{s+1}$.

It could be possible that the disobedient Explanation takes *fewer* points in Noise than the obedient one. For example, in Fig. 7.9(b) there are more points in Noise than in Fig. 7.9(a). It is easy to see that even though the disobedient explanation might contain fewer points in Noise than the obedient one, the number of points in Noise for the disobedient explanation must still be linearly related to that of the obedient one. Given a loop of size $s$, there are guaranteed to be at least $s$ links in the loops comprising the Explanation that must be shared by any equivalent explanation, that is, any explanation must have the property that each of these $s$ links must be the border of some plaquette in the explanation. Since a plaquette has three sides that could be used as this kind of border, the number of plaquettes used in a disobedient explanation can be no less than $s/3$. For $d$ dimensions (with $d/2$-dimensional plaquettes) we can then see that the number of plaquettes used in a disobedient explanation can be no less than $s/(d-1)$.

In this case, the effect on the counting is that when counting trees with $s$ points in Noise, one must count not only the obedient explanations with $s$ points in Noise, but also those with $s+1$ points, $s+2$,... and all the way up to $s(d-1)$ points in Noise; in this way we count all the disobedient explanations with $s$ points in Noise but whose obedient explanations have more than $s$ points in Noise.

If we define $q = 16r^8(d-1)(d-2)$, then the probability of obtaining a droplet of size $s$ becomes

$$\begin{aligned}
Prob(\text{droplet}) &\leq (2r^2)^{-s}\frac{2r}{16r^8}\sum_{n=s}^{\infty}(\sum_{j=n}^{n(d-1)} q^j)\epsilon^n \\
&= (2r^2)^{-s}\frac{2r}{16r^8}\sum_{n=s}^{\infty}\frac{1}{q-1}(q^{n(d-1)+1}-q^n)\epsilon^n \\
&< (2r^2)^{-s}\frac{1}{8r^7}\frac{q}{q-1}\frac{(q^{d-1}\epsilon)^s}{1-q^{d-1}\epsilon} \\
&< (2r^2)^{-s}\frac{1}{8r^7}\frac{2q}{q-1}(q^{d-1}\epsilon)^s \\
&\equiv (2r^2)^{-s}Q(q^{d-1}\epsilon)^s. \qquad (7.18)
\end{aligned}$$

Finding $r$ for $d$-dimensional plaquettes in $2d$ dimensions is simply done by counting the number of edges possibly emanating from a particular node; it turns out to be

$$r < 4(d-1)^2 \frac{d(d/2+1)}{4} \sim \frac{d^4}{2}. \tag{7.19}$$

Finally, we have showed all the above for a system acting as a quantum memory and that is on an infinite lattice. As before, incorporating computation and finite-size lattices is not hard. The computation is carried out via transversal implementation of quantum gates, and the proof carries over directly.

We have shown that the probability of loops growing too large is very small. However, one problem we have not yet adequately addressed is the following: what if many errors occur in a single time step, so that a nontrivial surface occurs without any large loop of string ever appearing? This probability should be extremely small, as such a nontrivial surface involves a number of plaquettes on the order of $s^2$ for a two-dimensional torus.

We need to be able to say the following: Let us assume at time $t$ that there are no globs (collections of loops of string) so large that the probability of them appearing is less than $Q(q^{4(d-1)}\epsilon)^{s/16}$, and let us also assume that correcting all errors leads to a homologically trivial surface. Let us further assume that at time $t+1$ there are no globs with probability less than $Q(q^{4(d-1)}\epsilon)^{s/16}$, as before. Then the probability that in the next time step so many errors occur that doing the correction leads to a homologically nontrivial surface is roughly proportional to $\epsilon^{s^2/32}$.

Such a nontrivial surface is given by a two-dimensional surface that wraps around the torus, *i.e.*, is roughly of size $s$ by $s$. Let us consider a two-dimensional (flat for simplicity) subsurface $S$ of a nontrivial surface, where $S$ has side length $L = s/4$. On this surface, there will be some loops of string. Given such a loop of string $l$, there are many surfaces with boundary given by this loop of string. Let us for this argument consider the minimal-area surface with boundary $l$; if that minimal surface is not in $S$, replace whatever plaquettes in $S$ have the boundary $l$ with the minimal surface.

The first case is that in which more than $1/4$ of the plaquettes in $S$ are in these minimal surfaces. Then we can add fewer than $3L/4$ more cells in Noise to form some droplet of size $L$. This means that the probability of such a configuration (the "glob") can be computed

by considering it as a droplet of size $L$ and then dividing out the probability that the extra $3L/4$ cells were in Noise, which is just $\epsilon^{3L/4}$. That is,

$$\begin{aligned} Prob(\text{first case}) &\leq Q\frac{(q^{d-1}\epsilon)^L}{\epsilon^{3L/4}} \\ &= Q(q^{4(d-1)}\epsilon)^{s/16}. \end{aligned} \tag{7.20}$$

The second case occurs if fewer than $1/4$ of the plaquettes in $S$ are contained in these minimal surfaces. Now we have to bound the probability that many plaquettes flip, resulting in small droplets that are not very dense, that nevertheless will be corrected in a way that leads to a nontrivial surface. (Again, if there are large droplets, or even many small droplets that are dense enough that their minimal surfaces cover $L/4$ of the area of $S$, the probability is bounded by the above.) That is, at least $3L/4$ of the surface must have been flipped (since the minimal surfaces of the droplets are allowed to cover $L/4$ of the surface). This gives that the number of plaquettes that need to be flipped for this to happen is at least $L^2/2$, so the probability this happens (for a given surface and given configuration at time $t+1$) is just $\epsilon^{L^2/2} = \epsilon^{s^2/32}$.

Finally, to finish the second case we must be sure that we are counting the number of such surfaces and the number of configurations possible at time $t+1$. The number of ending configurations cannot be more than $2^{L^2}$. Moreover, counting surfaces is a well-known problem. A variant of the technique given by (*e.g.*) Whittington and Soteros [91] (who themselved used techniques due to Klarner [49]) (see Appendix B) yields

$$a_n \leq 2^{d(d+1)n} \tag{7.21}$$

for the number of surfaces $a_n$ able to be made from $n$ $d/2$-dimensional plaquettes in $d$ dimensions. Thus the probability of the second case occuring is just

$$Prob(\text{second case}) = (2^{d(d+1)+1}\epsilon^{1/32})^{s^2} \tag{7.22}$$

which is extremely small as $s$ becomes large.

We know that uncorrectable errors for the quantum toric code with linear size $s$ in

dimension $d$ correspond to $d/2$-dimensional droplets that have $d/2-1$-dimensional boundary that is larger than $s^{d/2-1}$ (and are thus susceptible to being corrected in an erroneous way resulting in a homologially nontrivial loop). We have then shown, for a large torus of size $s$ in dimension $d$,

$$\begin{aligned} prob(\text{encoded gate error}(s,d)) &\leq prob(\text{droplet of } size > s/16) + (2^{d(d+1)+1}\epsilon^{1/32})^{s^2} \\ &< 2\sum_{m=s/16}^{\infty} \frac{q}{8r^7(q-1)(1-q^{d-1}\epsilon)} \frac{(q^{d-1}\epsilon)^{m^{d/2-1}}}{(2r^2)^m} \\ &= \frac{q(q^{d-1}\epsilon^{1/16})^{s^{d/2-1}}}{4r^7(q-1)(1-q^{d-1}\epsilon)(1-\frac{q^{d-1}\epsilon}{2r^2})}. \end{aligned} \tag{7.23}$$

We can now state the chief lemma of this section:

Lemma: For a $d$-dimensional toric code with $d/2$-dimensional error chains and linear dimension $s$, when the combined error probability of each quantum gate and associated local correction rule $\epsilon$ is fixed and sufficiently small, the probability of error per encoded gate is $\exp[-O(s^{d/2-1})]$.

## 7.3 Quantum blowup without fast classical computation

The extension of this 4-torus code to the $d$-torus with $d > 4$ will be used in our proof of Theorem 1. For clarity, we will first discuss the $d = 4$ case, and then the extension to $d > 4$ will be obvious.

So consider a fault-tolerant circuit that processes the protected qubits of the 4-torus code. After implementation of each quantum gate, the (local) error recovery circuit is executed a constant number of times. As shown in Sec. 7.2, if the "error probability" $\epsilon$ of the quantum gates is fixed and sufficiently small, then the probability of error per encoded gate is $\exp[-O(s)]$ where $s$ is the linear size of the lattice. In our simulation of a size $L$ circuit with error probability $\delta$, this failure rate per gate must be $O(\delta/L)$, or $s = O(\log(L/\delta)/\log(1/\epsilon))$. The block size of the code is $O(s^4)$, so that

$$\text{block size} = O(\log^4 L)\ , \tag{7.24}$$

with $\epsilon$ and $\delta$ fixed.

For the $d$-torus code the block size is of order $s^d$, the defects in the code block are $(d/2-1)$-dimensional, and hence the error rate is $\exp\left[-O(s^{d/2-1})\right]$. Thus to achieve a failure probability per gate of order $1/L$, we chose

$$\text{block size} = O(\log^\gamma L) \ , \quad \gamma = \frac{2}{1-2/d} \ ; \tag{7.25}$$

the power $\gamma$ approaches 2 from above as $d \to \infty$.

Sec. IX of [16] describes the implementation of fault-tolerant encoded gates for the 2-torus code of [48, 44]. Similar considerations apply to the 4-torus code or the $d$-torus code. The quantum gates in the normalizer group [35] (those whose action by conjugation takes tensor products of Pauli matrices to tensor products of Pauli matrices) can be implemented *transversally* producing a constant blowup in depth and a blowup in size of order the block size of the code. (A further slowdown in the implementation of the Hadamard gate is discussed in [16]. This additional slowdown arises if we insist on using local gates, since the transversal Hadamard gate produces a rearrangement of the qubits in the code block. However, this slowdown does not arise if nonlocal gates are allowed.)

To complete a universal set of quantum gates, the *quantum software* strategy is invoked [77, 37]. We prepare offline a suitable encoded ancilla state; then (transversal) encoded normalizer gates are performed, and encoded blocks are measured. Conditioned on the measurement outcomes, further normalizer gates are applied. Measurements are not really necessary, though, since we can replace operations conditioned on classical measurement outcomes with conditional quantum gates, at a constant cost in both size and depth.

Nevertheless, we must consider the complexity of the measurement of an encoded block, and we will argue that the slowdown due to this measurement dominates the depth of our simulation (contributing the $\log\log L$ factor in $D^*$ appearing in eq. (7.2)). The offline preparation of quantum software does not affect the depth, but we must consider whether the software preparation dominates the size.

### 7.3.1 Fault-tolerant measurement

First let's consider how we would destructively measure an encoded qubit if we could measure all the qubits in the (4-torus) code block, and then process the measurement outcomes

with a *flawless* classical computer.

If there were no errors in the block, the outcome of the encoded measurement could be determined by measuring the parity of the $s^2$ qubits lying on a minimal homologically nontrivial closed surface. But the parity cannot be accurately determined until we remove all the errors. A fault-tolerant procedure is to first measure all of the $O(s^4)$ qubits in the code block. Larger error droplets are destroyed by a coarse-graining procedure which works as follows:

We apply Toom's rule. Now, we partition the $d$-dimensional lattice into $d$-dimensional cubes of edge length 2 (volume $2^d$). Consider one of these coarse-grained cubes. It will contain various $d/2$-dimensional plaquettes of area $2^{d/2}$, both in the interior and on the surface of the coarse-grained cube. We get rid of all the $(d/2-1)$-dimensional string in the interior of all of these $d/2$-dimensional coarse-grained plaquettes; we then replace each $d$-dimensional block by the value given by the cells in the plaquette. (Since there is no string in the interior of these coarse-grained plaquettes, the value of all cells in the coarse-grained plaquette is the same.) After $\log s$ repetitions of this coarse-graining procedure, applying Toom's rule will get rid of droplets that are of size on the order of $s$.

The only thing left to check is that it is possible to do the string removal from the plaquettes as described above, which is not immediately obvious. Here is a recursive procedure for doing so: For each $d$-dimensional coarse-hypercube with string running through it, construct a string on the exterior of the coarse-hypercube that connects with the interior string to make a $(d/2-1)$-dimensional hyperloop. This takes no more than $d$ operations. Then, do a "baby" Toom's rule on the new hyperloop. Since the new hyperloop is contained within the coarse-hypercube, Toom's rule requires no more than $4d$ repetitions to eat away at the loop. Now, repeat the procedure for the $(d-1)$-dimensional hypercubes that make up the boundary of the $d$-dimensional hypercube, and so on until we get to plaquettes of dimension $d/2$. The cost in depth for this step alone is therefore at most

$$\mathit{depth}(\text{string removal}) \leq \frac{5(d+d/2)d/2}{2} = \frac{15}{4}d^2. \tag{7.26}$$

The total cost in depth for this procedure is of order $O(d^2 \log s) = O(\log s)$.

For smaller errors, it is sufficient to apply the robust classical version of our local "an-

nealing" algorithm a *constant* number of times. With probability of success exponentially close to one, all of the errors in the classical measured bits are removed. (Long strings that take a long time to anneal and cannot be quickly taken care of by the above proedure are exponentially rare, as we saw in Sec. 7.2.) Finally we measure the parity of the bits on a homologically nontrivial surface with a classical circuit of depth $\log s^2 = O(\log \log L)$.

Although we don't have a classical computer to process the measurement outcomes, we can simulate the classical circuit with a quantum circuit. Furthermore, we can use Von Neumann's fault-tolerant simulation to make this classical circuit robust.

There is a problem: Von Neumann's circuit processes classical information that is already encoded. To execute it, we must first fanout each of the $O(s^4)$ classical bits many times. Because we will want to compute the parity with probability of error $e^{-O(s)}$, we'll want to use a length $s$ repetition code, so that the fanout requires depth $\log s = O(\log \log L)$. However, a naive fanout will propagate errors badly — we need to achieve the fanout fault-tolerantly. This is possible, if we apply a two-dimensional classical Toom rule a constant number of times between fanouts, as we will see in 7.3.2. Then we can encode each measured classical bit using the repetition code with a probability of an encoding error of order $\epsilon$, and proceed to anneal all the errors with the (now robust) four-dimensional local annealing algorithm. Fanout and annealing together require depth $O(\log s)$ and size $O(s^5)$ (since each of $s^4$ bits is represented by a length $s$ block). Then the parity of $s^2$ encoded bits is computed in depth $O(\log s)$ and size $O(s^5)$.

After the encoded measurement, we need to apply an encoded normalizer gate conditioned on the outcome. To do that, we first apply further fanouts to the measurement outcome, extending the repetition code from length $s$ to length $s^4$ (in depth $O(\log s^4) = O(\log s)$ and size $O(s^4)$), and then we apply the conditional operation transversally (in constant depth and size $O(s^4)$), with each gate acting on a qubit of the 4-torus block conditioned on the value of the corresponding bit in the classical repetition code block.

Altogether, for the 4-torus code our fault-tolerant measurement circuit has size $O(\log^5 L)$ and depth $O(\log \log L)$. Therefore, implementing fault-tolerant gates that consume quantum software produces blowups in size and depth by these factors.

If we use the $d$-torus code, the blowup in depth is still a factor $O(\log \log L)$, but the blowup in size depends on $d$. To achieve an error probability per measurement $\exp[O(s^{d/2-1})]$,

we choose the length of the classical repetition code to be $O(s^{d/2-1})$. Apart from the blowup eq. (7.25) in size of the toric block, the classical repetition code produces a further blowup by a factor $O(\log L)$. Altogether, the blowup in the size of the measurement circuit is

$$O(\log^{\sigma} L) \ , \quad \sigma = \gamma + 1 = 3 \cdot \frac{1 - 2/(3d)}{1 - 2/d} \ ; \tag{7.27}$$

the power $\sigma$ approaches 3 from above as $d \to \infty$.

### 7.3.2 Fault-tolerant fanout

In the fault-tolerant measurement of the torus above, it is important to encode the classical bits corresponding to the measurement outcomes on the torus qubits into a length $s$ repetition code by a depth $\log s$ fanout. However, a naive fanout will propagate errors badly; we want to fanout fault-tolerantly. This is possible by applying a two-dimensional classical Toom rule a constant number of times at all fanout levels.

We can characterize this protocol by an inductive procedure, as for the case of concatenated codes [50], that achieves fault-tolerance in much the same way as in [50]. In particular, we denote by $C_1$ the protocol in which

1. One bit is encoded in four to make a two-by-two square, with some error at most $\epsilon_e$;
2. (a) Local errors are taking place with some error $\epsilon_d$;
   (b) Toom's rule is applied to the four-bit block with some error at most $\epsilon_t$;
3. The four bits are "decoded."

In [50], the decoding actually resulted in a smaller Hilbert space. Here, since we are interested in the whole space of physical qubits, "decoding" will be a purely formal step, requiring no operations, which just means that we will treat the four qubits as one logical qubit in future applications of Toom's rule (what this means will become more clear in the $C_h$ protocol).

We denote the protocol $C_h$ inductively as follows:

1. (a) One bit is encoded in four to make a two-by-two square, with some error at most $\epsilon_e$;

(b) The protocol $C_{h-1}$ is applied to the four bits;

2. (a) Local errors are taking place with some error $\epsilon_d$;

   (b) Toom's rule is applied to the four-bit block with some error at most $\epsilon_t$;

3. The four bits are "decoded."

Now we can see the purpose of the decoding step: for the protocol $C_h$, after the first step we actually have a $2^h \times 2^h$ number of bits. But because of the "decoding" step, in the second step we are only considering operations between the four $2^{h-1} \times 2^{h-1}$ code blocks. Note that we can use transversal operations to perform Toom's rule between code blocks that keep the Toom's rule step at constant depth.

Now we must assume that the error in applying Toom's rule, $\epsilon_t$, and the local error $\epsilon_d$, are sufficiently small that there exists some $r$ such that the total recovery error $\epsilon_c \leq \epsilon_t + O(\epsilon_d^2)$ has the property that $r\epsilon_c \leq \epsilon_d$. We will also assume that the encoding error $\epsilon_e$ is less than $\epsilon_c$.

In order to apply Von Neumann's fault-tolerant circuit to our block of size $s$, we need to be able to say that after the fanout the probability of more than $ts$ errors is at most proportional to $\epsilon_d$, where $t$ is some constant fraction. The key to doing so is to inductively consider probabilities that *blocks* of bits have become independently corrupted at each level of the hierarchical protocol.

We will define the $n$-block size $b_n \equiv 4^n$ as the number of cells present after performing the protocol $C_n$. Now we will consider the probability $pb_n$ that the block has been corrupted. A corrupted block is one that has errors in at least half of the cells of the block. Once a block has been diagnosed as corrupted, all cells in the block will be assumed to be in error. Thus we will be overcounting the corrupted-block probability of error and undercounting the probabilities of error for smaller amounts of corruption; however, we will always be overestimating the total probability of error at every stage of the protocol, so we will be able to find an upper bound using this coarse diagnosis procedure.

In order to complete the induction, we will also define $p_n(q)$, the probability that there are between $2^q$ and $2^{q+1}$ errors in the $n$-block, where $0 < q < 2n-1$. If we define a suitable cutoff hierarchy level $\eta$ and error rate $\epsilon_d$ (*e.g.*, $\eta = 4$ and $\epsilon_d < 0.001$ will work), and $\epsilon_t$ is sufficiently small compared to $\epsilon_d$, it is easy to calculate the base case probabilities directly

for $n \leq \eta$ and show that $p_n(q) < \epsilon_d^q$ and $pb_n < \epsilon_d$ for $n \leq \eta$. For $n > \eta$, we write $n = \eta + m$, where $m$ is greater than zero, and we would like to show

$$\begin{aligned} \sum_{q=1}^{m} p_{\eta+m}(q), &< 1, q \leq m & (7.28) \\ p_{\eta+m}(q) &< \epsilon^{q-m+1}, q > m & (7.29) \\ pb_n &< \epsilon_d. & (7.30) \end{aligned}$$

Eq. (7.28) is straightforward. To show Eq. (7.29) we assume the hypothesis true for $n-1$ and note that the principal contributions to $p_{\eta+m}(q)$ are the following: one corrupted $n-1$ block and at least one other $n-1$ block with $2^{q-1}$ errors; and one corrupted $n-1$ block and two other $n-1$ blocks with $2^{q-2}$ errors (choosing sufficiently large $\eta$ ensures that all other contributions, *e.g.*, from Toom errors at the correction step, are higher-order for $n > \eta$). We can thus write

$$\begin{aligned} p_{\eta+m} &\leq 12p_{\eta+m-1}(q-1)\epsilon_d + 12p^2_{\eta+m-1}(q-2)\epsilon_d + O(\epsilon^{q-m+1}) \\ &< 12\epsilon_d^{q-1-(m-1)-1} + 12\epsilon_d^{2(q-m-1)+1}\epsilon_d + O(\epsilon^{q-m+1}) \\ &< \epsilon_d^{q-m+1}. & (7.31) \end{aligned}$$

Similarly, for the inductive step to prove (7.30) we consider the following principal contributions to $pb_n$: an error in the encoding, which leads directly to the corruption of the entire block; an error due to at least two corrupted $n-1$-blocks; errors due to one corrupted $n-1$ block and one other $b(n-1)/2$-error block that persisted from the $n-1$-level protocol; and errors due to one corrupted $n-1$ block and two other $b(n-1)/4$-error blocks. Defining positive $n'$ such that $n = \eta + n'$, this gives

$$\begin{aligned} pb_n &= \epsilon_c + 6pb_{n-1} + 12p_{n-1}(2n-4)\epsilon_d + 12p_{n-1}(2n-5)\epsilon_d + O(\epsilon^{2\eta-2+n'}) \\ &< \epsilon_c + 6\epsilon_d^2 + 12\epsilon_d\epsilon_d^{2\eta+n'-3} + 12\epsilon_d\epsilon_d^{2(2\eta-4+n')} + O(\epsilon^{2\eta-2+n'}) & (7.32) \end{aligned}$$

Given our assumptions on $\epsilon_d$, we can see that the inductive step holds true. Finally, we can count up all the errors after we have done the fanout to a block of size $s$: the probability

of at least $ts$ errors is

$$p(\text{at least } ts \text{ errors}) < \epsilon_d + \sum_{q=\log ts}^{\log(s/2)} p_{\log s}(q). \tag{7.33}$$

Substituting in the expressions in (7.29), and letting $q = \log ts + q'$ gives

$$p(\text{at least } ts \text{ errors}) < \epsilon_d + \sum_{q'=0} \epsilon_d^{\log t+\eta+1} \epsilon_d^{q'}. \tag{7.34}$$

If $\eta$ is chosen so that it is larger than $-\log t$, and if $\epsilon_d$ is sufficiently small, this expression will converge to be no more than $3\epsilon_d$, which will let us apply Von Neumann's fault-tolerant circuit.

### 7.3.3 Software preparation

The preparation of high-fidelity encoded quantum software cannot be carried out directly on a large encoded block. Rather, we gradually build up the block, via a "quantum fanout procedure" with depth $O(\log s)$. If we use 4-torus coding, we want the probability of an encoded error in the software to be $\exp[-O(s)]$, so each time $s$ doubles we must square the error probability. This "purification" of the software is achieved through a protocol, described by Dennis [15] and by Kitaev [46], that generalizes the "recurrence" protocol [11] for purifying entangled pairs of qubits. The software purification protocol squares the probability of an encoded error through a constant number of normalizer gates and measurements of encoded blocks (which are then discarded).

Since the software fans out $O(\log s)$ times, and at each stage a constant number of measurements requiring depth $O(\log s)$ are executed, the software is prepared in depth $O(\log^2 s)$. However, the depth of the software preparation does not affect the depth of our fault-tolerant quantum circuit — the preparation is "offline" and causes no delay. The size of the purification circuit, dominated by the measurements that are required, is $O(s^5) = O(\log^5 L)$. If we use the $d$-torus code, the size of the purification circuit is as in eq. (7.27). (With $d$-torus coding, each time we double the linear size of the block, we raise the error probability to the power $2^{d/2-1}$ in a constant (for fixed $d$) number of purification steps.)

Now we have assembled all the elements needed for fault-tolerant simulation of a quan-

tum circuit using the 4-torus code or the $d$-torus code. The size and depth of the simulation are dominated by the measurement of encoded blocks. (We can replace measurements by conditional quantum gates, which causes a further blowup in size and depth by only a constant.) For any positive $\alpha$, we may choose $d$ large enough so that the cost is as in eq. (7.2). This proves Theorem 1.

Note that if the repertoire of our noisy computer does not include all of the gates used in the fault-tolerant simulation, a "Solovay-Kitaev" slowdown will arise in the construction of the necessary gates from the available gates. This further blowup is just a further constant factor in size and depth, and so has no affect on Theorem 1.

## 7.4 Quantum blowup when classical computation is fast

Note that the size and depth of our quantum circuits are dominated by computations that are essentially classical, needed to extract the outcomes of measurements of encoded blocks. Consider a different computational model, in which measurements yield classical outcomes, and classical processing is considered to be essentially instantaneous. In this model, the size and depth of the classical postprocessing of measurement outcomes is not included in our estimates of the size and depth of the circuit of *quantum* gates. At a cost in classical processing that is polylogarithmic in $L$, we can reduce the quantum blowup in depth to a constant. The cost in size is of order the blowup eq. (7.25) in the block size. For any positive $\beta$, we may choose $d$ large enough so that the cost is as in eq. (7.3). This proves Theorem 2.

With Steane's highly parallelized syndrome computation [81], the quantum processing needed to extract the syndrome of a concatenated code also has constant depth and polylog($L$) size [70]; thus a result analogous to Theorem 2 can be established using concatenation, but we have improved on the power of $\log L$ in the size blowup realized by concatenated coding.

The postprocessing of the measured syndrome is carried out level-by-level in the concatenated code. If noisy gates are used, then fault-tolerant processing at each level is built from fault-tolerant gadgets constructed at the next level down, resulting in a blowup in depth by $(\text{constant})^\ell = \text{polylog}(L)$ (where $\ell$ is the number of levels of concatenation). Thus, with

concatenated codes we do not know how to achieve the $\log \log L$ blowup in depth asserted in Theorem 1.

## 7.5 Status of Pippenger's conjecture

In the model in which size polylog($L$) classical processing is regarded as effectively instantaneous, we have come arbitrarily close to realizing the blowup envisioned by Pippenger. Can we improve the result to $\beta = 0$ in eq. (7.3)?

As discussed in [16], with size polylog($L$) reliable classical postprocessing of measurement outcomes, we can achieve an error probability per gate of order $1/L$ using a 2-torus code with block size $\log^2 L$, ostensibly the block size needed to realize the Pippenger blowup in size. However, the recovery method described in [16] involves $O(s)$ repetitions of the syndrome measurement between successive encoded gates (where $s$ is the linear size of the torus), resulting in a blowup by a factor $O(\log L)$ in depth and $O(\log^3 L)$ is size.

With the 4-torus code (or the $d$-torus code for $d > 4$), it suffices to measure the syndrome a constant number of times between encoded gates. Furthermore, by processing the measured syndrome optimally, we can achieve an error probability per gate $\exp[-O(s^2)]$, an improvement over the $\exp[-O(s)]$ established for the local recovery method used in the proofs of Theorems 1 and 2. The catch is that the optimal recovery algorithm may require too much computation. Given the syndrome (the locations of the string loops), the optimal recovery method involves flipping qubits on the minimal-area 2-surface bounded by the strings, and finding the minimal surface with a specified boundary is NP-hard [6] (it is equivalent to finding the ground state of a four-dimensional Ising spin glass). Perhaps we can show that finding the minimal surface (up to a homologically trivial surface) is an easy problem when the error rate is small and the loops of string typically have small length and are dilute. Then we can improve Theorem 2 to an exact rather than approximate $O(\log^2 L)$ blowup in size.

And in the model considered in Theorem 1 (no reliable classical gates), can the cost be reduced to a constant factor in depth and a factor $O(\log^2 L)$ in size? The blowup found in eq. (7.2) was dominated by the simulated measurement of encoded blocks, a feature of the implementation via quantum software of universal quantum gates for topological codes.

Some sort of quantum software strategy is a necessary part of quantum fault tolerance, as Gottesman has shown [36]. Closing the gap between Theorem 1 and Pippenger's conjecture might be achieved by eliminating the (simulated) measurement step in the implementation of fault-tolerant encoded gates. Can we prepare quantum software offline (with a quantum circuit of sufficiently small size) that interacts with encoded blocks and is then discarded without ever being "measured"? It is not clear whether such a construction of universal fault-tolerant gates is possible.

# Appendix A

# One droplet becoming many

As mentioned in section 7.2.3, the proof given there has the flaw that sometimes a droplet does not result in a single plaquette after many applications of the modified Toom's rule because of the multiple dimensions. So we cannot yet directly use the results on a single-plaquette tree given in section 7.2.1; in particular, there can be more than one history in the original tree. However, it is still possible to suitably rewire Gacs' protocol so that everything works.

Now, let $V_0'$ — analogous to $V_0$ in Gacs— be the set of (individual, disconnected) cells obtained by evolving the droplet forward, as in the previous section. In the previous section this set was one cell; now that is no longer the case. (We can define $V_0'$ precisely by the following: Construct the "original" tree as in the 2D Toom's rule, by applying Toom's rule without noise until all errors are gone. For $Time(u) \geq 0$, consider the subgraph of $(V_{original}, Arrows)$ (the graph induced by the Original tree) formed by the cells $\{v \in V_{original} | Time(v) \geq Time(u)\}$. By analogy to Gacs' histories, let's call the connected component of this subgraph containing $u$ the *extrapolation* of $u$. Any $u$ with $Time(u) \geq 0$ such that $Extrapolation(u) = \{u\}$ will now be in $V_0'$.) For a given tree, define $G_K$ as in Gacs; it is still connected, but may not now encompass the entire explanation for $K = History(u), u \in V_0'$, as was true when $V_0'$ was a single cell.

Let us define $G_{drop}$ in the obvious way consistent with Gacs' notation: if $t = 0$ is the time at which the droplet is present, $G_{drop}$ is the subgraph of $(V, Arrows)$ induced by taking all the cells $v \in V$ with $Time(v) \leq 0$. If $V_0'$ is a two-element set $\{a, b\}$, there are two cases.

The first case is that in which $History(a) = History(b)$ is the whole graph. Arrange the

arrows for the tree for $t > 0$ by choosing the poles of the droplet such that $Span(droplet) = Size(droplet)$ and working backwards using the Gacs construction backwards. The number of edges here is precisely as in the Gacs construction.

The second case is that in which $History(a)$ is distinct from $History(b)$. Then the number of edges can be found by treating the explanation as two distinct explanation trees: the number of edges is simply less than $4(n_1 - 1) + 4(n_2 - 1) \leq 4(n - 1)$, where $n_1, n_2$ are the number of cells in Noise for the explanation trees for $a, b$ respectively.

In the case where $V_0'$ has $q$ elements, the number of cases goes like $2^q$. So we need to multiply the probablity found in the previous section by $2^q$. But each cell in $V_0'$ adds at least one to the boundary, because a different history can only happen when two different orientations come together, and the different orientation necessitates at least one extra plaquette in Noise to Explain it. So we effectively only need to multiply by $2^{boundary}$.

# Appendix B

# Lattice animals

Here we give a proof based closely on [91] that the number of connected $m$-dimensional surfaces made up of $n$ $k$-dimensional plaquettes in a square lattice in $d$ dimensions is less than $2^{2m(2d-2m+1)n}$.

First, note that a $m$-dimensional "plaquette" on a $d$-dimensional integer square lattice can be parametrized by its center, which has some coordinates $(x_1, x_2, ...x_d)$. These coordinates are such that $k$ of the coordinates are half-integers and $d-m$ coordinates are integers.

Let $S_0$ be the set of centers in the surface of interest. We will define the *top* of a set $S_0$ of such centers by the following. Define the set $S_1 \subset S_0$ as the set of centers for which the first coordinate $x_1$ is the maximum over $S_0$. Define the set $S_2 \subset S_1$ as the set of centers in $S_1$ for which the second coordinate $x_2$ is the maximum over $S_2$. Define $S_j$ recusrively in this way until $S_j$ contains only one element, which will be the *top* of the set. Define the bottom in the same way, replacing the word "maximum" above with "minimum."

Now we will give a unique ordering to the points in $S_0$. For each of the $2m$ edges of a $k$-dimensional plaquette, there are $2(d-(m-1))-1$ ways another plaquette in $d$ dimensions could attach to the given plaquette. This means that there are a total of $2m(2d-2m+1)$ edges one could draw from one center to another center (though this will lead to some overcounting later, as only $2m$ edges at a time are actually possible). Let us specify a canonical ordering of these $2m(2d-2m+1)$ edges, $\{l_1, l_2, ...l_{2m(2d-2m+1)}\}$. Let $v_1$ be the bottom point. We number the $k$ edges incident on $v_1$ $1...k$ in the order of the canonical ordering, and the vertices connected to $v_1$ through these edges $2...k+1$. Number the edges

of $v_2$ in an order corresponding to the canonical ordering, and so on.

We will now place the edges in the surface into a vector that will uniquely define the surface. For $v_k$, $k > 1$, there exists at least one vertex $v_j$ connected to $v_k$ through an edge such that $j < k$. Let $j$ be the smallest such value, and let $r$ be the order of the edge in the canonical ordering given above. Now we number the $i$th edge emanating from $v_k$ (where $i$ is given by the canonical ordering subscript $\{l_i\}$) with the number $s_i = (i - r) mod 2m(2d - 2m + 1)$ for each $i = 1...2m(2d - 2m + 1), i \neq r$. We place a zero in the $(2m(2d - 2m + 1) + s_i)$th location of a $2m(2d - 2m + 1)n$ - length vector if the $i$th edge is not in the surface or if it connects to $v_p$ for $p < k$; otherwise we place a one in that position. (In the special case $k = 1$, assume the edge $(v_1 - \hat{x}_2, v_1)$ is in the surface and proceed as above. This proceudre gives a $2m(2d - 2m + 1)n$-length vector for any surface. There are $2^{2m(2d-2m+1)n}$ such vectors, and thus the number of surfaces that can be made with $n$ plaquettes, $a_n$, satisfies

$$a_n \leq 2^{2m(2d-2m+1)n}. \tag{B.1}$$

(In fact this is overcounting by a great deal, and it's possible that further work could be done to reduce the threshold. Nevertheless this is good enough for our purposes.)

# Bibliography

[1] D. Aharonov and M. Ben-Or. Fault-tolerant quantum computation with constant error. In *Proc. 29th Ann. ACM Symp. on Theory of Computing*, New York, 1998. ACM. quant-ph/9611025; D. Aharonov and M. Ben-Or, "Fault-tolerant quantum computation with constant error rate," quant-ph/9906129 (1999).

[2] C. Ahn, A. C. Doherty, and A. J. Landahl. Continuous quantum error correction via quantum feedback control. *Physical Review A*, 65:042301, 2002. quant-ph/0110111.

[3] G. Alber, Th. Beth, Ch. Charnes, A. Delgado, M. Grassl, and M. Mussinger. Detected jump-error correcting quantum codes, quantum error designs and quantum computation. quant-ph/0208140.

[4] G. Alber, Th. Beth, Ch. Charnes, A. Delgado, M. Grassl, and M. Mussinger. Stabilizing distinguishable qubits against spontaneous decay by detected-jump correcting quantum codes. *Physical Review Letters*, 86:4402–4405, 2001. quant-ph/0103042.

[5] M. A. Armen, J. K. Au, J. K. Stockton, A. C. Doherty, and H. Mabuchi. Adaptive homodyne measurement of optical phase. *Phys. Rev. Lett.*, 89:133602, 2002.

[6] F. Barahona. On the computational complexity of ising spin-glass models. *J. Phys. A*, 15:3241–3253, 1982.

[7] A. Barenco, T. A. Brun, R. Schack, and T. Spille. Effects of noise on quantum error correction algorithms. *Physical Review A*, 56:1177–1188, 1997. quant-ph/9612047.

[8] S. D. Barrett and G. J. Milburn. Measuring the decoherence rate in a semiconductor charge qubit. *Phys. Rev. B*, 68:155307, 2003.

[9] C. H. Bennett and G. Brassard. Quantum cryptography: Public-key distribution and coin tossing. In *Proceedings of IEEE International Conference on Computers, Systems and Signal Processing*, pages 175–179, Bangalore, India, 1984. C. H. Bennett and G. Brassard, "Quantum public key distribution," IBM Technical Disclosure Bulletin **28**, 3153–3163 (1985).

[10] C. H. Bennett, G. Brassard, C. Crepeau, R.Jozsa, A. Peres, and W.Wootters. Teleporting an unknown quantum state via dual classical and epr channels. *Phys. Rev. Lett.*, 70:1895–1899, 1993.

[11] C. H. Bennett, D. P. Divincenzo, J. A. Smolin, and W. K. Wootters. Mixed state entanglement and quantum error correction. *Physical Review A*, 54:3824, 1996. quant-ph/9701015.

[12] T. M. Buehler, D. J. Reilly, R. P. Starrett, A. Greentree, A. R. Hamilton, A. S. Dzurak, and R. G. Clark. Efficient readout with the radio frequency single electron transistor in the presence of charge noise. *Phys. Rev. B (Rapid). Submitted.*, 2003.

[13] H. J. Carmichael. *An Open Systems Approach to Quantum Optics*. Springer-Verlag, Berlin, 1993.

[14] I. L. Chuang and Y. Yamamoto. The persistent qubit. quant-ph/9604030.

[15] E. Dennis. Fault-tolerant computation without concatenatio. *Phys. Rev. A*, 63:052314, 2001. quant-ph/9905027.

[16] E. Dennis, A. Kitaev, A. Landahl, and J. Preskill. Topological quantum memory. *J. Math. Phys.*, 43:4452–4505, 2002. quant-ph/0110143.

[17] D. Deutsch. Quantum theory, the church-turing principle and the universal quantum computer. *Proc. R. Soc. Lond. A*, 400:97, 1985.

[18] R. L. Dobrushin and S. I. Ortyakov. Upper bound for the redundancy of self-correcting arrangements of unreliable functional elements. *Problems Inform. Transmission*, 13:203–218, 1977.

[19] Jennifer L. Dodd, Michael A. Nielsen, Michael J. Bremner, and Robert T. Thew. Universal quantum computation and simulation using any entangling hamiltonian and local unitaries. *Physical Review A*, 65:040301, 2002.

[20] A. C. Doherty. *Motion, Measurement and Control in an Open Quantum System.* Ph.D. thesis, The University of Auckland, 1999.

[21] A. C. Doherty, S. Habib, K. Jacobs, H. Mabuchi, and S. M. Tan. Quantum feedback control and classical control theory. quant-ph/9912107.

[22] A. C. Doherty and K. Jacobs. Feedback-control of quantum systems using continuous state-estimation. *Physical Review A*, 60:2700, 1999. quant-ph/9812004.

[23] W. H. Zurek E Knill, R. Laflamme. Resilient quantum computation: error models and thresholds. *Proc. Roy. Soc. London, Ser. A*, 454:365, 1998. quant-ph/9702058.

[24] L. Fedichkin and A. Fedorov. Decoherence rate of semiconductor charge qubit coupled to acoustic phonon reservoir. *Phys. Rev. A*, 69:032311, 2004. quant-ph/0309024.

[25] R. P. Feynman. Simulating physics with computers. *Int. J. Theor. Phys.*, 21:467, 1982.

[26] D. F.Walls and G. J.Milburn. *Quantum Optics.* Springer-Verlag, Berlin, 1994.

[27] P. Gacs. Reliable computation with cellular automata. *Journal of Computer System Science*, 32:15–78, 1986.

[28] P. Gacs. Reliable cellular automata with self-organization. *J. Statist. Phys*, 103:45–267, 2001. arXiv:math.PR/0003117.

[29] C. W. Gardiner. *Handbook of Stochastic Methods.* Springer, Berlin, 1985.

[30] Peter Gács. A new version of toom's proof, 1995. Tech. report, Boston University, Department of Computer Science, Boston, MA 02215.

[31] H. S. Goan. An analysis of reading out the state of a charge quantum bit. *Quant. Inf. and Comp.*, 3:121, 2003.

[32] Hsi-Sheng Goan and G. J. Milburn. Dynamics of a mesoscopic charge quantum bit under continuous quantum measurement. *Phys. Rev. B*, 64:235307, 2001.

[33] Hsi-Sheng Goan, G. J. Milburn, H. M. Wiseman, and He Bi Sun. Continuous quantum measurement of two coupled quantum dots using a point contact: A quantum trajectory approach. *Physical Review B*, 63:125326, 2001.

[34] D. Gottesman. *Stabilizer codes and quantum error correction*. Ph.D. thesis, Caltech, 1997. quant-ph/9705052.

[35] D. Gottesman. A theory of fault-tolerant quantum computation. *Phys. Rev. A*, 57:127, 1998. quant-ph/9702029.

[36] D. Gottesman, 2000. Private communication.

[37] D. Gottesman and I. L. Chuang. Quantum teleportation is a universal computational primitive. *Nature*, 402:390–392, 1999. quant-ph/9908010.

[38] Daniel Gottesman. Class of quantum error-correcting codes saturating the quantum hamming bound. *Physical Review A*, 54:1862, 1996.

[39] M. Grassl, T. Beth, and T. Pellizzari. Codes for the quantum erasure channel. *Physical Review A*, 56:33–38, 1997.

[40] J. Harrington and C. Ahn. In preparation.

[41] L. C. L. Hollenberg, A. S. Dzurak, C. Wellard, A. R. Hamilton, D. J. Reilly, G. J. Milburn, and R. G. Clark. Charge-based quantum computing using single donors in semiconductors. *Phys. Rev. B (Brief Reports)*, 69:113301, 2003.

[42] B. E. Kane. A silicon-based nuclear spin quantum computer. *Nature (London)*, 393:133, 1998.

[43] K. Khodjasteh and D. A. Lidar. Quantum computing in the presence of spontaneous emission by a combined dynamical decoupling and quantum-error-correction strategy. *Phys. Rev. A*, 68:022322, 2003. quant-ph/0301105.

[44] A. Y. Kitaev. Fault-tolerant quantum computation by anyons, 1997. quant-ph/9707021.

[45] A. Y. Kitaev. Quantum computations: algorithms and error correction. *Russ. Math. Surv.*, 52(6):1191–1249, 1997.

[46] A. Yu. Kitaev. unpublished.

[47] A. Yu. Kitaev. Quantum computations: algorithms and error correction. *Russian Math. Surveys*, 52:1191–1249, 1997.

[48] A. Yu. Kitaev. Quantum error correction with imperfect gates. In O. Hirota, A. S. Holevo, and C. M. Caves, editors, *Proceedings of the Third International Conference on Quantum Communication and Measurement*, New York, 1997. Plenum.

[49] D. A. Klarner. Cell growth problems. *Canadian Journal of Mathematics*, 25:585–602, 1965.

[50] E. Knill and R. Laflamme. A theory of quantum error-correcting codes. *Physical Review A*, 55:900–911, 1997. quant-ph/9604034.

[51] E. Knill, R. Laflamme, and G. Milburn. Efficient linear optics quantum computation. *Nature*, 409:46, 2001. quant-ph/0006088.

[52] P. L. Koleden, E. Platen, and H. Schurz. *Numerical solution of SDE through computer experiments*. Springer-Verlag, Berlin, 1994.

[53] A. N. Korotkov. Selective evolution of a qubit state due to continuous measurement. *Physical Review B*, 63:115403, 2001. cond-mat/0008461.

[54] K. Kraus. *States, Effects, and Operations: Fundamental Notions of Quantum Theory*, volume 190 of *Lecture Notes in Physics*. Springer-Verlag, Berlin, 1983.

[55] R. Laflamme, C. Miquel, J.-P. Paz, and W. H. Zurek. Perfect quantum error correction code. *Phys. Rev. Lett*, 77:198, 1996. quant-ph/9602019.

[56] D. A. Lidar, I. L. Chuang, and K. B. Whaley. Decoherence free subspaces for quantum computation. *Phys. Rev. Lett.*, 81:2594, 1998. quant-ph/9807004.

[57] H. Lo and H. F. Chau. Unconditional security of quantum key distribution over arbitrarily long distances. *Science*, 283:2050–2056, 1999.

[58] D. Loss and D. P. DiVincenzo. Quantum computation with quantum dots. *Phys. Rev. A*, 57:120, 1998.

[59] H. Mabuchi and P. Zoller. Inversion of quantum jumps in quantum optical systems under continuous observation. *Physical Review Letters*, 76:3108–3111, 1996.

[60] D. Mayers. Unconditional security in quantum cryptography, 1998. quant-ph/9802025.

[61] G. J. Milburn and He Bi Sun. Quantum open-systems approach to current noise in resonant tunneling junctions. *Phys. Rev. B*, 59:10748, 1999.

[62] C. Mochon. Private communication.

[63] Y. Nakumura, Y. A. Pashkin, and J. S. Tsai. Coherent control of macroscopic quantum states in a single-cooper-pair box. *Nature*, 398:786, 1999.

[64] M. A. Nielsen and I. L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, Cambridge, 2000.

[65] A. V. Oppenheim, A. S. Willsky, and S. Hamid Nawab. *Signal and Systems*. Prentice Hall, 2nd edition, 1996.

[66] J. P. Paz and W. Zurek. Continuous error correction. *Proc. Trans. R. Soc. Lond. A*, 454:355–364, 1998. quant-ph/9707049.

[67] N. Pippenger. On networks of noisy gates. In *IEEE Symposium on Foundations of Computer Science, vol. 26*, pages 30–38, New York, 1985. IEEE Press.

[68] N. Pippenger, 2002. private communication.

[69] M. B. Plenio, V. Vedral, and P. L. Knight. Quantum error correction in the presence of spontaneous emission. *Physical Review A*, 55:67, 1997. quant-ph/9603022.

[70] J. Preskill. Fault-tolerant quantum computation. In H.-K. Lo, T. Spiller, and S. Popescu, editors, *Quantum Information and Computation*. World Scientific, Singapore, 1998. quant-ph/9712048.

[71] J. Preskill. Lecture notes for caltech course ph 219: Quantum information and computation, 1998. http://www.theory.caltech.edu/~preskill/ph219/.

[72] J. Preskill. Reliable quantum computers. *Proc. Roy. Soc. Lond. A*, 454:385–410, 1998. quant-ph/9705031.

[73] T.C. Ralph, A. Gilchrist, G.J. Milburn, W.J. Munro, and S. Glancy. Quantum computation with optical coherent states. *Physical Review A*, 68:042319, 2003. quant-ph/0306004.

[74] M. Sarovar. Private communication.

[75] R. J. Schoelkopf, P. Wahlgren, A. A. Kozhevnikov, P. Delsing, and D. E. Prober. The radio-frequency single-electron transistor (rf-set): A fast and ultrasensitive electrometer. *Science*, 280:1238, 1998.

[76] P. W. Shor. Scheme for reducing decoherence in quantum computer memory. *Physical Review A*, 52:2493, 1995.

[77] P. W. Shor. Fault-tolerant quantum computation. In *Proceedings, 37th Annual Symposium on Foundations of Computer Science*, pages 56–65, Los Alamitos, CA, 1996. IEEE Press. quant-ph/9605011.

[78] P. W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J.Sci.Statist.Comput.*, 26:1484, 1997. quant-ph/9508027; see also Proceedings of the 35th Annual Symposium on Foundations of Computer Science, Santa Fe, NM, Nov. 20–22, 1994.

[79] P. W. Shor and J. Preskill. Simple proof of security of the bb84 quantum key distribution protocol. *Phys. Rev. Lett.*, 85:441–444, 2000. quant-ph/0003004.

[80] A. Steane. Error-correcting codes in quantum theory. *Phys. Rev. Lett.*, 77:793, 1996.

[81] A. Steane. Active stabilization, quantum computation, and quantum state synthesis. *Phys. Rev. Lett.*, 78:2252, 1997. quant-ph/9611027.

[82] T. Tanamoto. Coherent control of macroscopic quantum states in a single-cooper-pair box. *Phys. Rev. A*, 61:22305, 2000.

[83] A. I. Toom. Stable and attractive trajectories in multicomponent systems. In R. I. Dobrushin, editor, *Multicomponent Systems, Advances in Probability, vol. 6*, pages 549–575. Dekker, New York, 1980.

[84] R. van Handel, J. K. Stockton, and H. Mabuchi. Feedback control of quantum state reduction, 2004. quant-ph/0402136.

[85] L. Viola, E. Knill, and S. Lloyd. Dynamical decoupling of open quantum systems. *Phys. Rev. Lett.*, 82:2147–2421, 1999. quant-ph/9809071.

[86] L. Viola and S. Lloyd. Dynamical suppression of decoherence in two-state quantum systems. *Physical Review A*, 58:2733, 1998. quant-ph/9803057.

[87] J. von Neumann. Probabilistic logics and the synthesis of reliable organisms from unreliable components. In C. E. Shannon and J. McCarthy, editors, *Automata studies*. Princeton University Press, Princeton, 1956.

[88] I. Walmsley and H. Rabitz. Quantum physics under control. *Physics Today*, 56:43–49, 2003.

[89] J. Wang and H. M. Wiseman. Feedback-stabilization of an arbitrary pure state of a two-level atom: a quantum trajectory treatment. *Physical Review A*, 64:063810, 2001. quant-ph/0008003.

[90] U. Weiss. *Quantum Dissipative Systems*. World Scientific, 1999.

[91] S. G. Whittington and C. E. Soteros. Lattice animals: Rigorous results and wild guesses. In Geoffrey Grimmett and Dominic Welsh, editors, *Disorder in physical Systems: A Volume in Honour of John M. Hammersley*. Oxford University Press, 1990.

[92] D. J. Wineland, C. Monroe, W. M. Itano, D. Leibfried, B. E. King, and D. M. Meekhof. i don't know. *Journal of Research of the National Institute of Standards and Technology*, 103:259, 1998.

[93] H. M. Wiseman. Quantum theory of continuous feedback. *Physical Review A*, 49:2133–2150, 1994. Errata in Phys. Rev. A **49**, 5159 (1994), Phys. Rev. A **50**, 4428 (1994).

[94] H. M. Wiseman. Adaptive phase measurements of optical modes: Going beyond the marginal q distribution. *Phys. Rev. Lett.*, 75:4587, 1995.

[95] H. M. Wiseman. Using feedback to eliminate back-action in quantum measurements. *Physical Review A*, 51:2459, 1995.

[96] H. M. Wiseman. Quantum trajectories and quantum measurement theory. *Quantum Semiclassical Optics*, 8:205–222, 1996.

[97] H. M. Wiseman and R. B. Killip. Adaptive phase measurements of optical modes: Going beyond the marginal q distribution. *Phys. Rev. A*, 57:2169, 1998.

[98] H. M. Wiseman and G. J. Milburn. Quantum theory of field-quadrature measurements. *Physical Review A*, 47:642, 1993.

[99] H. M. Wiseman and G. J. Milburn. Quantum theory of optical feedback via homodyne detection. *Phys. Rev. Lett.*, 70:548, 1993.

[100] H. M. Wiseman, Dian Wahyu Utami, He Bi Sun, G. J.Milburn, B. E. Kane, A. Dzurak, and R. G.Clark. Quantum measurement of coherent tunneling between quantum dots. *Phys. Rev. B*, 63:235308, 2001.

[101] K. Zhou, K. Glover, and J. C. Doyle. *Robust and Optimal Control*. Prentice-Hall, Upper Saddle River, N.J., 1996.