

Continuous Error Correction in Quantum Dot Qubits

Stasiu Wolanski, Jesus College

2022 - 2023

Abstract

Woop!

Contents

0.1	Introduction	1
1	Background	2
1.1	Quantum error correction	2
1.1.1	Quantum computing	2
1.1.2	Error correction	2
1.1.3	Continuous error correction	2
1.2	Quantum Dot Qubits	2
1.2.1	Semiconductor Heterostructures	2
2	Theory	3
2.1	Rapidly repeated error correction schemes	3
2.2	Truly continuous error correction in quantum dot systems	3
2.2.1	Scheme summary	3
2.2.2	Theoretical treatment of scheme with second order perturbation theory	4
2.2.3	Further questions related to tunnelling physics	6
3	Methods	7
3.1	Simulating the hubbard model	7
3.1.1	Python extension: <code>FastHamiltoniser</code>	7
3.1.2	Python class for Hubbard model construction: <code>Hamiltonian</code>	9
3.1.3	Python class for time evolution of statevector: <code>Evolution</code>	9
3.2	Truly continuous error correction prototype	9
3.2.1	Hubbard Hamiltonian setup	10
3.2.2	Energy shifts due to hopping coupling	11
3.2.3	Demonstrating the error propagation	11
4	Feasibility of the scheme	13
4.1	Conclusions	13
	Appendices	15
A	Decoherence theory	16
B	Detecting errors in ancilla qubits.	18
C	Locked states	19

0.1 Introduction

Chapter 1

Background

1.1 Quantum error correction

1.1.1 Quantum computing

1.1.2 Error correction

1.1.3 Continuous error correction

1.2 Quantum Dot Qubits

1.2.1 Semiconductor Heterostructures

The Hubbard Model

Chapter 2

Theory

2.1 Rapidly repeated error correction schemes

This section develops a framework for analysing error correction schemes that are based on discrete, but rapidly repeated, parity measurement and correction. It allows for the efficient and exact calculation of the expected fidelity of such a scheme given certain characteristic parameters: the rate of errors, the frequency of error correction cycles, and the fidelity of the error syndrome measurements. The crucial insight is that all standard decoherence processes have the property that, when they are applied to a diagonal density matrix, the result is another diagonal density matrix. This means that by choosing a basis where the initial (codespace) state of the system is one of the basis vectors, we can analyse the system density matrix as a vector, and apply markovian evolution by calculating the relevant transition matrix.

2.2 Truly continuous error correction in quantum dot systems

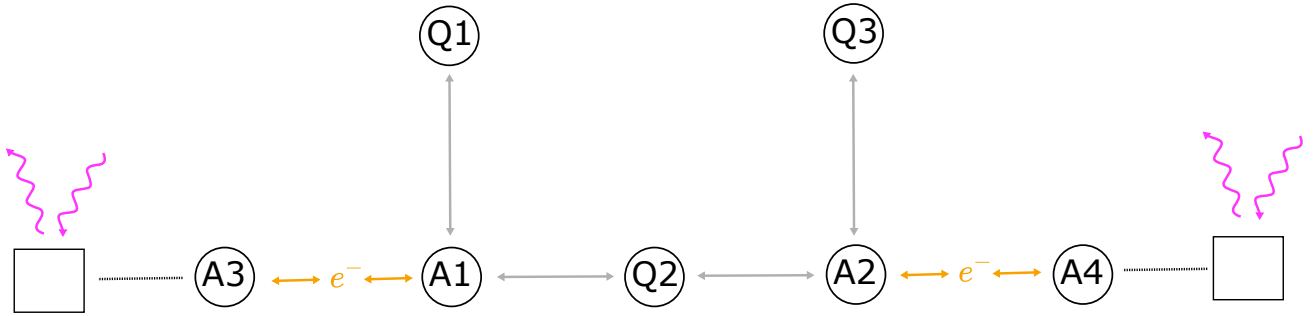


Figure 2.1: Schematic of the layout of qubits for the continuous error correction scheme. Q1, Q2 and Q3 comprise a single logical qubit. A1 to A4 are ancilla qubits used to detect bit flip errors in the logical qubits. Q1 and Q2 are continuously weakly coupled to A1 through a tunnel barrier, and likewise Q2 and Q3 to A2. R1 and R2 are resonators, which are additional quantum dots held at a higher electron occupancy than the qubits. These are probed using RF signals, and their coupling to the charge on the ancillas then allows the singlet/triplet state of the ancillas, as in [5].

2.2.1 Scheme summary

In this section a proposal for truly continuous error correction is developed. I thank Chris Long, Dr Mertig, Giovanni Oakes and my supervisor Dr Arvidsson-Shukur for contributing variously to the development of the ideas presented here. Recent developments in the field of RF reflectometry measurements enable the measurement of the singlet/triplet state of two adjacent qubits coupled via a tunnel barrier, by monitoring the rf reflection characteristics of a neighbouring electron reservoir [5]. The codespace qubits are to be coupled pairwise to two ancillary qubits whose function is to

perform parity measurements, in such a way that any deviation from the codespace will result in a change of state of one or both of the ancilla qubits, depending on the error syndrome. The rate at which the error propagates to the ancilla qubit(s) is in the tens of nanoseconds, a faster timescale than either gate times (hundreds of nanoseconds) or readout (the state of the art [5] achieves 6ps).

This scheme is potentially more efficient than a repeated error correction scheme like those analysed in section 2.1. One reason for this is that the ‘dead time’ after an error has occurred during which the system is vulnerable to a second error is limited by the time for the ancilla to be reset, which is also on the order of microseconds (e.g. see [3]). A second, more significant advantage is that the scheme does not require any quantum gates to be performed for the purposes of error syndrome detection, as such detection happens continuously and simultaneously with circuit gates. This reduces circuit complexity, and avoids the error accumulation associated with performing the syndrome measurements.

TODO flesh this out.

The proposed qubit layout is given in figure 2.1. The nature of the circuit requires a 2D array of quantum dot structures, of which there is not yet any published experimental demonstration, but a proposed architecture was proposed in [Tadokoro2021’2]. In the scheme, the logical qubit is encoded in the *odd parity subspace* of three qubits Q1, Q2, and Q3. There are two pairs of ancilla qubits: A1, A3, and A2, A4. Each of these is initially prepared in the T_0 triplet state. As shown in the figure, each pair of qubits Q1 and Q2, Q2 and Q3, is weakly coupled to one ancilla qubit through a voltage-controlled tunnel barrier. If the couplings are sufficiently well-tuned, the ancilla qubits will remain as triplets and no tunneling will occur. When a bit-flip error occurs on one of Q1, Q2 or Q3, this will cause one or both (depending on the error syndrome) of the ancilla pairs to rotate into a singlet state. This allows (subject to complexities discussed in section 2.2.3) tunneling between the electrons in the affected pair, which is registered as a change in dispersive response of the nearby resonator. Thus the error syndrome is immediately detected as a signal in the continuously monitored reflectometry circuit.

2.2.2 Theoretical treatment of scheme with second order perturbation theory

Suppose the system is initially in an arbitrary codespace state, with each of the ancilla pairs starting in triplet states:

$$|\psi(t=0)\rangle = (\alpha|\uparrow_{Q1}\downarrow_{Q2}\uparrow_{Q3}\rangle + \beta|\downarrow_{Q1}\uparrow_{Q2}\downarrow_{Q3}\rangle)(|\uparrow_{A1}\downarrow_{A3}\rangle + |\downarrow_{A1}\uparrow_{A3}\rangle)(|\uparrow_{A2}\downarrow_{A4}\rangle + |\downarrow_{A2}\uparrow_{A4}\rangle)/2$$

First consider the case where no coupling is applied between the qubits. If we can engineer our system so that $\delta_{1,3}^z = E_{A1}^z - E_{A3}^z \ll 1/T_1$, and similarly $\delta_{2,4}^z \ll 1/T_1$ (with T_1 the timescale of bit flip errors), then the energy difference between the two branches of each qubit singlet is sufficiently small that rate at which the triplets rotate into triplets, and so register a tunnelling event on the reflectometry apparatus, is long compared to the error time and so not limiting. We can see this by considering a single ancilla branch:

$$|\psi(t)\rangle = \frac{1}{\sqrt{2}}(|\uparrow\downarrow\rangle + \exp(-2i\delta t)|\downarrow\uparrow\rangle) = \frac{1}{2}|T_0\rangle(1 + \exp(-2i\delta t)) + \frac{1}{2}|S_0\rangle(1 - \exp(-2i\delta t)) \quad (2.1)$$

, so that the amplitude of the singlet state goes as

$$|\langle S_0|\psi(t)\rangle|^2 = \sin(\delta t)^2$$

Now we apply a coupling between the following four pairs of qubits: Q1 and A1, Q2 and A1, Q2 and A2, Q3 and A2. We consider the four branches of the state in the computational basis of the ancilla qubits separately, and apply second order perturbation theory to approximate the energy shift of each state due to the adiabatic application of a weak coupling, of strength t . The limits of this approximate treatment will be tested using a numerical simulation developed in chapter 3.

The standard formula for second order perturbation theory is (there is no first-order shift in energy as the perturbing terms are off-diagonal):

$$E_i^1 \approx E_i^0 + \sum_j \frac{|\langle\psi_i|V|\psi_j\rangle|^2}{E_i^0 - E_j^0}$$

where E_i^0 and E_i^1 are the unperturbed and perturbed energies of state i , and H_0 and V are the basic and perturbative hamiltonians. If we set V to the hopping term in the hubbard model, then we obtain, for example,

$$\begin{aligned} E_{\uparrow\downarrow\uparrow,\uparrow\downarrow,\uparrow\downarrow}^1 - E_{\uparrow\downarrow\uparrow,\uparrow\downarrow,\uparrow\downarrow}^0 &= -\frac{t^2}{E_{\uparrow\uparrow,\emptyset\downarrow,\uparrow\downarrow}^0 - E_{\uparrow\downarrow,\uparrow\downarrow,\uparrow\downarrow}^0} - \frac{t^2}{E_{\uparrow\emptyset\uparrow,\uparrow\downarrow,\uparrow\downarrow}^0 - E_{\uparrow\downarrow,\uparrow\downarrow,\uparrow\downarrow}^0} - \frac{t^2}{E_{\uparrow\uparrow\uparrow,\uparrow\downarrow,\emptyset\downarrow}^0 - E_{\uparrow\downarrow,\uparrow\downarrow,\uparrow\downarrow}^0} - \frac{t^2}{E_{\uparrow\emptyset\uparrow,\uparrow\downarrow,\uparrow\downarrow}^0 - E_{\uparrow\downarrow,\uparrow\downarrow,\uparrow\downarrow}^0} \\ &= -t^2 \left(\frac{1}{\Delta_1} + \frac{1}{\Delta_2} + \frac{1}{\Delta_3} + \frac{1}{\Delta_4} \right) \end{aligned}$$

Computational branch		Ancilla branch			
Subspace	branch	$\uparrow\downarrow\uparrow\downarrow$	$\uparrow\downarrow\downarrow\uparrow$	$\downarrow\uparrow\uparrow\downarrow$	$\downarrow\uparrow\downarrow\uparrow$
Codespace	$\uparrow\downarrow\uparrow\downarrow$ $\uparrow\downarrow\downarrow\uparrow$	$2\Lambda_t$	$2\Lambda_t$	$2\Lambda_t$	$2\Lambda_t$
E1	$\uparrow\downarrow\uparrow\downarrow$ $\downarrow\uparrow\uparrow\downarrow$	$1\Lambda_t$ $3\Lambda_t$	$1\Lambda_t$ $3\Lambda_t$	$3\Lambda_t$ $1\Lambda_t$	$3\Lambda_t$ $1\Lambda_t$
E2	$\uparrow\downarrow\uparrow\downarrow$ $\uparrow\uparrow\uparrow$	$4\Lambda_t$ 0	$2\Lambda_t$	$2\Lambda_t$	0 $4\Lambda_t$
E3	$\uparrow\downarrow\uparrow\downarrow$ $\uparrow\downarrow\downarrow\uparrow$	$1\Lambda_t$ $3\Lambda_t$	$3\Lambda_t$ $1\Lambda_t$	$1\Lambda_t$ $3\Lambda_t$	$3\Lambda_t$ $1\Lambda_t$

Table 2.1: Table of the (negative) energy shifts due to coupling of each ancilla branch, of each computational branch, of each subspace.

with

$$\begin{aligned}
\Delta_1 &= U_{Q2} + \mu_{Q2} - \mu_{A1} + E_{Q2}^z - E_{A1}^z \\
\Delta_2 &= U_{A1} + \mu_{A1} - \mu_{Q2} + E_{Q2}^z - E_{A1}^z \\
\Delta_3 &= U_{Q2} + \mu_{Q2} - \mu_{A2} + E_{Q2}^z - E_{A2}^z \\
\Delta_4 &= U_{A2} + \mu_{A2} - \mu_{Q2} + E_{Q2}^z - E_{A2}^z
\end{aligned}$$

, where the terms describe parameters of the Hubbard model described in section 1.2.1.

We note that $U_i \gg \Delta\mu_i, E_i^z$, and that the magnitude of U_i (order of a few meV) varies little from site to site, so that we may define $\Lambda_t = 2t^2/\langle U_i \rangle$. Then we have

$$E_{C\uparrow\downarrow,\uparrow\downarrow}^1 - E_{C\uparrow\downarrow,\uparrow\downarrow}^0 = \Delta E_{C\uparrow\downarrow,\uparrow\downarrow} \approx \Delta E_{C\uparrow\downarrow,\uparrow\downarrow} \approx \Delta E_{C\downarrow\uparrow,\uparrow\downarrow} \approx \Delta E_{C\downarrow\uparrow,\uparrow\downarrow} \approx -2\Lambda_t$$

, where we note the shift in energy is the same, within the above approximations, for both of the codespace branches, $|\uparrow\downarrow\uparrow\rangle$ and $|\downarrow\uparrow\downarrow\rangle$, and denote the energies of both branches by the subscript C. All branches of the ancilla state will therefore accrue phase at the same rate as long as the state code qubits remains within the codespace.

When a bit-flip error occurs, the state will acquire a component in one of the error subspaces, denoted E1, E2 and E3 depending on which qubit has flipped. Applying the above analysis, we calculate the energy shifts of each of the ancilla branches of each computational branch in each codespace. These are given in table 2.1. These shifts result in the A1, A3, or A2, A4 ancilla pairs rotating from triplets to singlets, or both, depending on the error subspace. To see this, consider the E1 error subspace, into which the state moves when the Q1 qubit experiences a bit-flip error. If both ancilla pairs are in triplet states at the point when the system transitions into the E1 subspace from the codespace, the state undergoes the following evolution ¹:

$$\begin{aligned}
|\psi(t)\rangle &= \alpha |\downarrow\downarrow\uparrow\rangle (e^{-3i\Lambda_t t} |\uparrow\downarrow\uparrow\rangle + e^{-3i\Lambda_t t} |\uparrow\downarrow\downarrow\rangle + e^{-i\Lambda_t t} |\downarrow\uparrow\uparrow\rangle + e^{-i\Lambda_t t} |\downarrow\uparrow\downarrow\rangle)/2 \\
&+ e^{-i\delta_{Q1,Q3}t} \beta |\uparrow\uparrow\downarrow\rangle (e^{-i\Lambda_t t} |\uparrow\downarrow\uparrow\rangle + e^{-i\Lambda_t t} |\uparrow\downarrow\downarrow\rangle + e^{-3i\Lambda_t t} |\downarrow\uparrow\uparrow\rangle + e^{-3i\Lambda_t t} |\downarrow\uparrow\downarrow\rangle)/2 \\
&\approx e^{-3i\Lambda_t t} \alpha |\downarrow\downarrow\uparrow\rangle (|\uparrow\downarrow\rangle + e^{2i\Lambda_t t} |\downarrow\uparrow\rangle)(|\uparrow\downarrow\rangle + |\downarrow\uparrow\rangle)/2 \\
&+ e^{-i\Lambda_t t} \beta |\uparrow\uparrow\downarrow\rangle (|\uparrow\downarrow\rangle + e^{-2i\Lambda_t t} |\downarrow\uparrow\rangle)(|\uparrow\downarrow\rangle + |\downarrow\uparrow\rangle)/2
\end{aligned}$$

, where the $\delta_{Q1,Q3}$ phase term accounts for the difference in energy between the two branches, which is small compared to Λ_t , and irrelevant to our conclusion, so we ignore it. By comparison with equation 2.2.2, we see that the A1,A3 ancilla pair in both branches will rotate into a triplet state with a characteristic timescale $1/\Lambda_t$. The same analysis yields for E2 error subspace:

$$\begin{aligned}
|\psi(t)\rangle &\approx \alpha |\downarrow\downarrow\uparrow\rangle (|\uparrow\downarrow\rangle + e^{-2i\Lambda_t t} |\downarrow\uparrow\rangle)(|\uparrow\downarrow\rangle + e^{-2i\Lambda_t t} |\downarrow\uparrow\rangle)/2 \\
&+ e^{-4i\Lambda_t t} \beta |\downarrow\downarrow\uparrow\rangle (|\uparrow\downarrow\rangle + e^{2i\Lambda_t t} |\downarrow\uparrow\rangle)(|\uparrow\downarrow\rangle + e^{2i\Lambda_t t} |\downarrow\uparrow\rangle)/2
\end{aligned}$$

, so that both pairs of ancillas rotate into singlets. The E3 case is identical to the E1 case with the appropriate qubits exchanged, and leads to only the A2, A4 qubit pair rotating into a singlet.

¹We assume that the resultant state is approximately a stationary state of the perturbed hamiltonian.

2.2.3 Further questions related to tunnelling physics

Much work has been done optimising the implementation and examining the classical physics of reflectometry as a tool for quantum measurement. For a review see, e.g. [6], especially section VIII. During the project the author has not had time to study the physics behind this technique in detail. It appears, however, that present analyses treat the system semi-classically in assuming that an electron tunnels, or does not, within the measurement period (i.e. the period of time that the tunneling barrier between the pair of qubits is lowered). They then seek to maximise the fidelity with which this charge-shifting event, or lack thereof, is detected. This tunneling event is interpreted as a projective measurement of the singlet/triplet system at the point of measurement, with probabilities of either state given by Born's rule. In the present case this is an insufficient analysis: at any given time, there is necessarily some finite component of the electron pair state that is a singlet, and we need to establish a rate at which tunnelling occurs as a function of the amplitude of this component. A full treatment of this problem probably needs to be formulated in terms of quantum master equations and the theory of quantum decoherence [7].

Time has not permitted me to look into this issue, so at the suggestion of my supervisor I will make the simplifying assumption that any ancilla pair that acquires more than a 10% square-amplitude of singlet immediately tunnels and is hence registered by the dispersive readout device with empirically determined fidelity.

Chapter 3

Methods

3.1 Simulating the hubbard model

This section will discuss the methodology used to perform simulations of scheme described in section 2.2 on a hubbard model of a multiple quantum dot system. The treatment of the error correction scheme described in section 2.2 is necessarily highly approximate due to the complexity of the high-dimensional hilbert space involved (for 7 electrons on 7 sites, the occupation basis is of dimension 3432) and the large number of parameters of the Hubbard model, so makes multiple simplifying assumptions, chiefly the application of second-order perturbation theory and ignoring the site-to-site variance in certain model parameters. It is therefore valuable to simulate the Hubbard model numerically to test the validity of these assumptions in various parameter regimes.

My supervisor Dr. Arvidsson-Shukur have developed a code (yet unpublished), **DotHamiltoniser**, designed to generate Hubbard hamiltonians in an occupation basis given a set of physical parameters (E_i^z, U_i , etc.), as well as to perform low-dimensional time-evolution simulations. This was very helpful at the start of my project to visualise the energy levels of the system and their dependence on coupling, but was not designed for higher dimensional time evolution, and could not be used for this purpose due to performance limitations.

I therefore designed a Hamiltonian simulation code, inspired by the work of Dr Arvidsson-Shuker et al., with the ability to handle higher dimensional systems. The system has three main components:

1. A python extension, **FastHamiltoniser**, written in C, C++ and CUDA, which provides highly performant specialised sparse matrix operations for use in Hamiltonian time evolution;
2. **Hamiltonian**, a python class that builds a Hubbard model Hamiltonian from physical parameters; and
3. **Evolution**, a python class which takes as input a Hamiltonian and a set of continuous time instructions to produce a time evolution of a given starting statevector.

3.1.1 Python extension: FastHamiltoniser

Structure of extension

The purpose of the simulator is to solve the Schrödinger equation

$$\dot{\psi} = -iH\psi \quad (3.1)$$

(here expressed in natural units $\hbar = 1$) where ψ is a vector in the occupation basis of a Hubbard Hamiltonian. In order to use Runge-Kutta [1] methods of integration, we need to be able to calculate this derivative efficiently at an arbitrary time and with arbitrary input. H is typically very sparse, with the only off-diagonal terms given by magnetic drives and hopping couplings, of which typically only a subset of size linear in the number of qubits are non-zero at a given time.

To perform this calculation efficiently, I implemented the following data structure in C. A **HamiltonianMatrix** object, illustrated in figure 3.1, contains basic data about the dimension of the array, as well as an array of pointers to objects of type **Permuter**. Each of these **Permuter** objects stores three arrays of equal length, two of integers representing source and target vector indices, and one of floats representing matrix coefficients, where each entry represents a matrix element. **Permuters** also store a complex coefficient, which can be changed over time by writing to a single variable, and a boolean representing whether it is ‘on’ or ‘off’. The internal arrays are determined in advance of the simulation by python code, and thereafter only the strength and activations of the **Permuter** objects need be

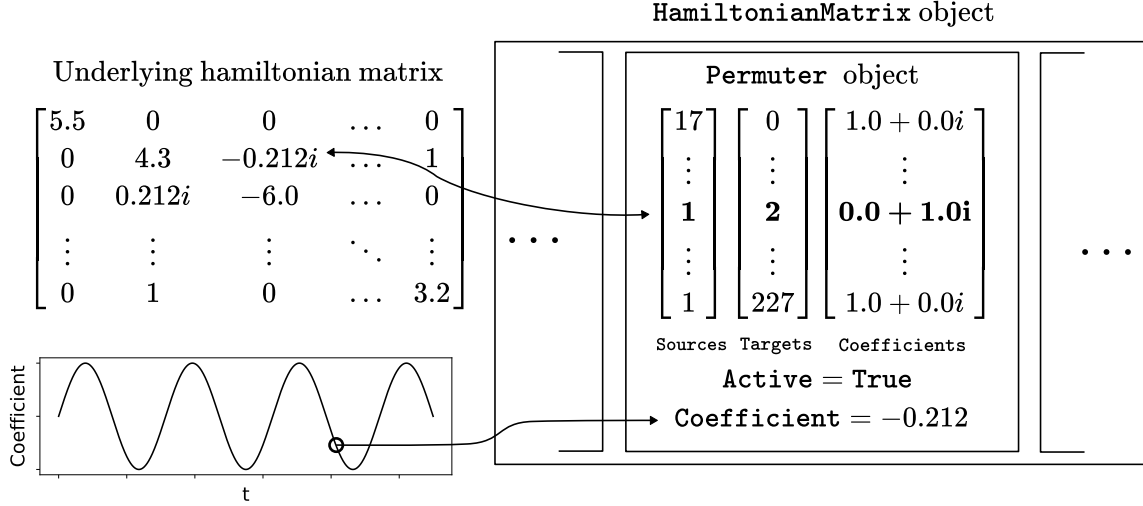


Figure 3.1: How sparse Hamiltonian matrices are represented in the custom python extension `FastHamiltoniser`. The non-zero matrix elements are grouped into objects of type `Permuter`, whose structure is calculated in advance but whose activation and multiplicative coefficients are modified efficiently as runtime.

modified during the simulation. For example, one such object provides all the on-diagonal elements of the matrix, and another might provide all the off-diagonal elements associated with a particular J-coupling between two qubits.

The calculation 3.1 can then be performed by iterating over the active permuters and building up the derivative vector according to the permutation and coefficient arrays.

Interaction picture calculation

A further requirement that favoured the development of a custom C backend was the ability to perform simulations in the interaction picture. Performing simulations in the interaction picture is necessary due to the large differences in energy between eigenstates, for example due to the U_i on-site repulsion in some states. With any choice of reference energy in a static basis, there are always states whose phase rotate much faster than the timescale of the evolutions that I wish to simulate, meaning there is no suitable step-size that captures the desired phenomena within reasonable computational time with the simulation remaining stable. Instead, we perform the following transformation to the interaction picture:

$$\begin{aligned}\dot{\psi}_i &= -i \sum_j H_{ij} \psi_j \\ \psi_i &\rightarrow \phi_i = e^{-iH_{ii}t} \psi_i \\ \Rightarrow \dot{\phi} &= -i \sum_{j \neq i} e^{-i(H_{jj} - H_{ii})t} H_{ij} \phi_j\end{aligned}$$

Since most off-diagonal terms H_{ij} are zero at most points in time, this results in most states being stationary and lowering the rate of rotation of affected states to a longer timescale that can be simulated. It introduces the additional complexity, however, of introducing a time-dependence into the hamiltonian. I wrote a C function to perform the modified matrix multiplication, calculating the phase factors as required.

GPU acceleration

The calculation to be performed is parallel in nature, and thus a candidate for GPU acceleration. I wrote the requisite kernels¹ to perform the matrix multiplication in CUDA, a variant of C++ developed by Nvidia to control their graphics processing units ('GPU's') [4], one of which (a P620 Quadro) is installed in my laptop. On testing, the CUDA variant

¹This presented a lot of technical challenges as the process for incorporating CUDA C++ into python C extensions and compiling them is not documented and very technical. I made a template project with the setup to use in the future: <https://github.com/Stasiu51/TemplateCudaEx>

outperformed the CPU calculation only for systems with more than around 15 qubits, more than required and where the speed of computation is too slow for practical use anyway (< 10 iterations per second, several orders of magnitude too slow for the simulations of the present project).

3.1.2 Python class for Hubbard model construction: `Hamiltonian`

This is a python class (and supporting code) that implements a Hubbard Hamiltonian, using `FastHamiltoniser` as a backend to store the underlying matrix. In particular it

1. creates a list of basis vectors, represented as bit strings (with convenience functions for printing them in canonical notation), that span the fock basis of a system with a specified number of sites and number of electrons;
2. defines creation and annihilation operators on these states, which it uses to
3. construct the diagonal `Permuter` object that represents the self-energy terms due to chemical potential, coulomb repulsion, and static magnetic z-fields; and to
4. construct the off diagonal `Permuter` objects generated by hopping terms and x and y magnetic fields. It also
5. provides transformations to and from the interaction picture; and
6. provides functions for creating projector and X matrices in the fock basis, which are more involved to calculate than in the computational (single electron per site) basis.

3.1.3 Python class for time evolution of statevector: `Evolution`

This class takes a reference to a `Hamiltonian` object, and a set² of objects that subclass the abstract base class `TimeStep` whose role is to provide instructions as to how to modify the parameters of the Hubbard model at given time. For example, one such object instructs the simulation to ramp a set of hopping couplings to a specified value over a given period of time, and another might oscillate the strength of the x component of the applied magnetic field at a given frequency and phase.

The `scipy` routine `solve_ivp` is used along with the calculations of the derivative described above in a Runge-Kutta 5th order [1] integration routine to calculate the evolution of a given initial statevector. The routine is designed to estimate the errors and dynamically choose a step size for the integration. This works poorly in the present use case, as the calculated (vector-norm) error scales with the number of non-zero elements of the vector. Instead, the code allows the user to manually specify a step size, typically around 1/10 of the smallest relevant timescale of the system, and checks the continued normalisation of the statevector as an imperfect check on the stability of the simulation.

Some demonstrations of the simulator and the results of a benchmarking exercise are shown in figure 3.2.

3.2 Truly continuous error correction prototype

This section explains how the software described above was used to simulate the ‘truly continuous’ error correction scheme developed in section 2.2. In particular, we want to explore suitable physical parameters, such that when we initialise the system in an initial codespace state, and adiabatically turn on a weak coupling between the relevant qubits

- the ancilla pairs remain in triplet states as long as the system remains in the codespace, for at least a time scale longer longer than ; and
- when a bit-flip error is artificially introduced, one or both of the ancilla pairs rotates into a singlet, as appropriate to the error syndrome.

Physical parameters

It is useful to survey the orders of magnitude of the various physical paramters of the experiment based on theoretical and experimental work to date. These can be found in table 3.1. As this is an exploratory piece of thoeretical work, I will take these as fairly liberal bounds.

²There is in fact only one such object, as they are defined in a recursive tree structure.

⁴Applying a Z unitary rotation is not a capability typically built into such processors as it is unnecessary due to a technique called *Rz* delaying (see, e.g. [Nielsen2010] chapter 4.), but I implement it here to make the operation of the CPHASE clearer.

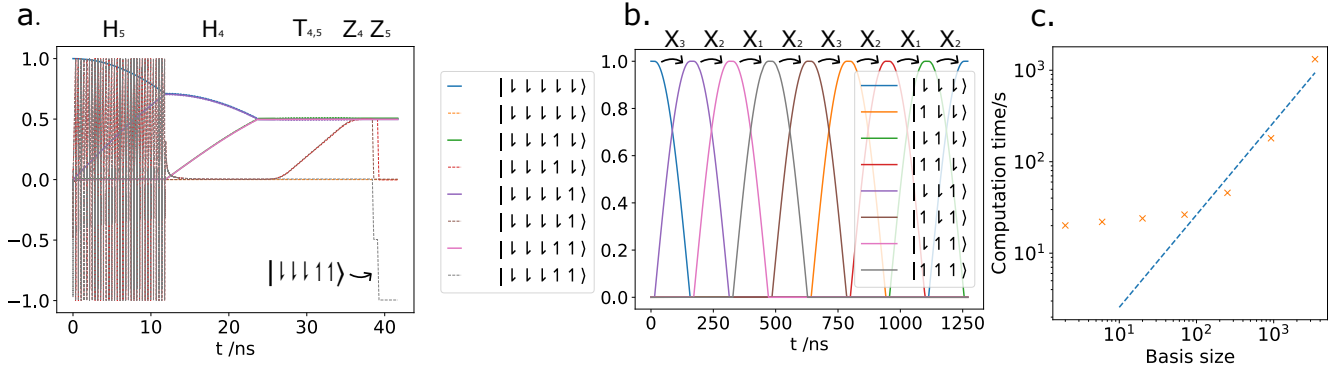


Figure 3.2: Testing the capabilities of the simulation software. **a.** A plot of 5-qubit system showing the amplitudes (solid lines) and phases (dashed lines) of significant basis states. The plot demonstrates of a CPHASE gate acting on the last two qubits, so that the $|\downarrow\downarrow\downarrow\uparrow\uparrow\rangle$ state acquires a phase shift of π . First an X gate is applied to each of the two qubits, then a coupling between them is applied, and finally a z-rotation is applied by increasing the static field⁴. **b.** A sequence of X flips on a 3-qubit system. This circuit is generalised to N qubits to benchmark the simulator, with results given in **c.** The results indicate a consistent speed of around 7.5 million statevector derivative elements calculated per second.

Parameter	Symbol	Range	Example Reference
On-site coulomb repulsion	U_i	$\sim 1\text{Ry} \sim 1\text{-}10 \text{ meV}$	[Shim2022]
Static magnetic (z) field	B_i^z	1 - 10 T	[Jock2018]
Variation in electron g factor	δB_i^z	0.01 - 0.1 MHz	[2]
Hopping coupling strength	t_{ij}	0.001 – 0.01 meV	[Veldhorst2015]
Phase flip coherence time	T_2^*	up to 10^3ns in Si, 10^5ns in ^{28}Si	[Loss2022]
Bit flip coherence/relaxation time	T_1	up to 10^6ns in normal Si	[Loss2022]

Table 3.1: Table of the (negative) energy shifts due to coupling of each ancilla branch, of each computational branch, of each subspace.

3.2.1 Hubbard Hamiltonian setup

The software enables us to set up a qubit basis that explicitly models the qubit layout shown in 2.1, with three code qubits and two pairs of ancilla qubits. This seven-qubit system has a fock basis of dimension 3432, which is just about feasible to simulate using the software I developed (simulations of sufficiently small time step take several minutes each, and there are memory limitations in storing the results). However, this can be sped up by noting that in the circuit to be simulated, the second of each pair of ancillas, A3 and A4, do not experience any coupling, so the hamiltonian is always diagonal in these dimensions. Furthermore, the ancilla part of all states that are non-zero in the simulation are constructed out of the $|\uparrow\downarrow\rangle$ and $|\downarrow\uparrow\rangle$ states with respect to these pairs of qubits, as well as non-computational states. Therefore, we may reduce our basis if we make an appropriate modification to the Hamiltonian:

$$\begin{aligned}
|\dots\uparrow\downarrow\uparrow\downarrow\rangle &\rightarrow |\dots\uparrow\uparrow\rangle & |\dots\uparrow\downarrow\downarrow\uparrow\rangle &\rightarrow |\dots\uparrow\downarrow\rangle \\
|\dots\downarrow\uparrow\uparrow\downarrow\rangle &\rightarrow |\dots\downarrow\downarrow\rangle & |\dots\downarrow\uparrow\downarrow\uparrow\rangle &\rightarrow |\dots\downarrow\downarrow\rangle \\
E_{A1}^z &\rightarrow E_{A1}^z - E_{A3}^z & E_{A2}^z &\rightarrow E_{A2}^z - E_{A4}^z
\end{aligned}$$

this does not affect the energy levels of any of the computational states, and only affects the non-computational basis states by a quantity of order $E_i^z \ll U_i$. If we accept this approximation, we can reduce our basis size to 252, which increases the simulation speed by a factor of around $14 - 20\times$ (this is a non-linear speedup as all the simulation data now fits into memory).

Also note that most variation in spin energy from site to site come from variation in the electron G factor at different sites rather than inhomogenaities in the magnetic field [2]. It is convenient and equivalent, however, to simply include these variations into the magnetic field⁵

⁵Although my software does have the capability to vary the electron g-factor from site to site, it is simpler not to use it.

3.2.2 Energy shifts due to hopping coupling

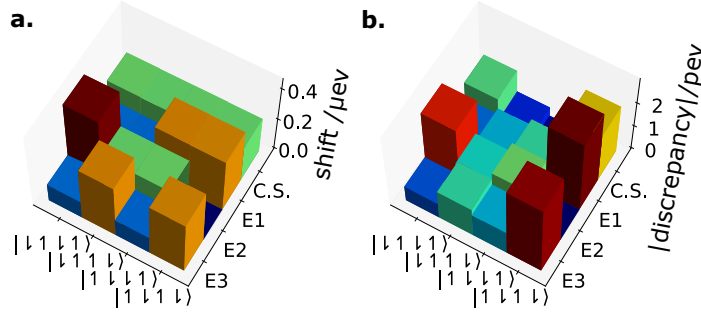


Figure 3.3: **a.** shows the calculation of energy shifts of the various relevant computational basis states calculated with a coupling of 0.015meV applied. Only one branch $|\uparrow\downarrow\uparrow\rangle$ of the computational state in each subspace is shown. This can be compared with the second order perturbation theory result calculated in table 2.1. The (absolute value of) the discrepancy between the simulation and the perturbation theory result is given in **b.** Note the scale in the second plot.

It is important to know the energies of the perturbed computational states for two reasons: firstly, in order to validate the second-order perturbation theory results in section 2.2; and secondly knowing the exact energies of these states is important if one is to use oscillating microwave signals to drive transitions between the states, to implement unitary rotations on the qubits while the coupling is active.

We find these by the following procedure: we initialise the system in the computational basis state of interest, and the hamiltonian with all couplings zero. we perform an interaction picture simulation where the couplings are ramped up to the steady state value over a time period that is sufficiently long so as to ensure adiabatic evolution of the state ($100 \times \langle U_i \rangle^{-1}$ as this is the shortest timescale in the system). We compare this state to a list of the eigenvectors of the matrix generated by the Hamiltonian with the coupling at full strength. The eigenvector which has the maximal inner product with this state (the inner product is close to unity for the correct state, and close to zero otherwise) is the perturbed eigenvector associated with the original state in the computational basis state, and we take the associated eigenvalue as the perturbed energy.

The results for a coupling of 0.015meV , which is slightly higher than a typical coupling, and other typical experimental parameters, are given in figure 3.3. The simulation agrees with the perturbation theory calculation within 1%. The agreement is closer for smaller coupling values.

3.2.3 Demonstrating the error propagation

We now want to demonstrate the process of propagation by which bit-flip errors in the code qubits Q1, Q2, Q3 result in the relevant ancilla pairs rotating into singlets, or in the reduced basis approximation described in 3.2.1, the qubits A1 and A2 rotating from $(|\uparrow\rangle + |\downarrow\rangle)/\sqrt{2}$ to $(|\uparrow\rangle - |\downarrow\rangle)/\sqrt{2}$, which is approximately equivalent.

We can achieve this by initialising the code qubits in the codespace, and then adiabatically turning on the coupling. We let this system run in steady state for enough time to establish the rate at which the singlets are introduced in the codespace. We then instantaneously apply an X unitary on one of the qubits by, and continue to let the simulation run. Afterwards, we can project the resulting time-evolved statevector data into the singlet and triplet spaces. The system prior to the bit flip is shown in figure 3.4, and the behaviour when a bitflip is introduced mid-way through the simulation is shown in figure 3.5.

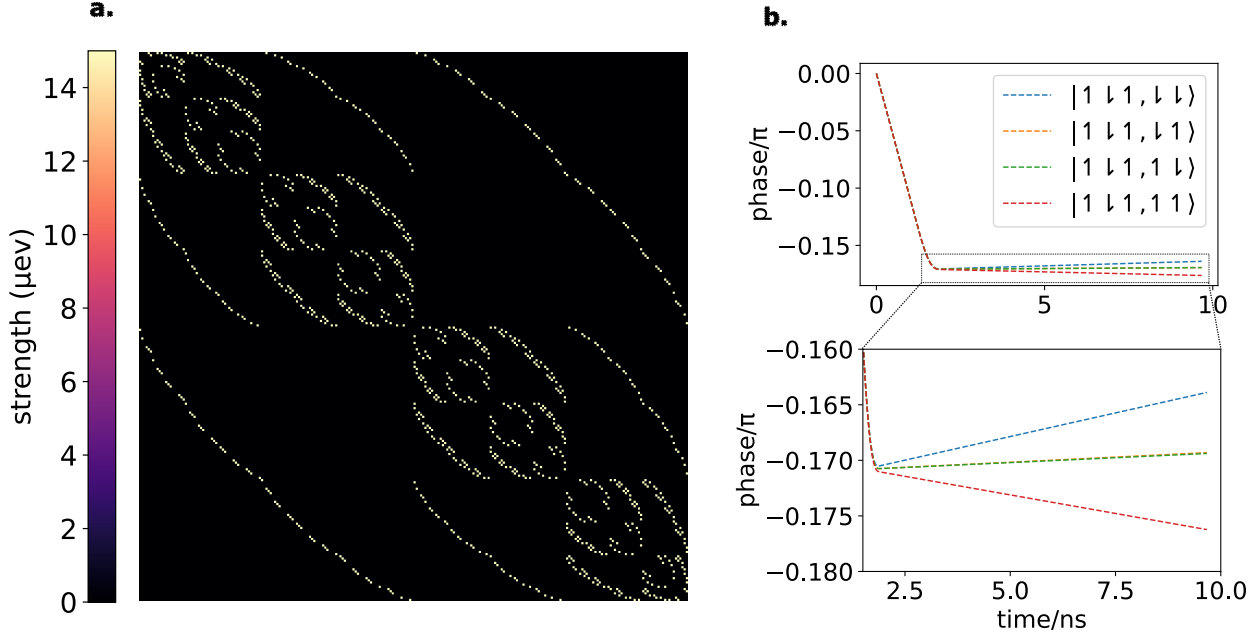


Figure 3.4: The effect of applying the hopping coupling to the system still inhabiting the codespace. **a.** shows the (elementwise absolute value of the) hamiltonian matrix in the interaction picture at full strength. **b.** shows the rate of acquisition of phase of the relevant basis states. The phase initially drops when the coupling is applied, and then the states slowly drift apart from one another in phase due to the difference in electron g-factor of around 30 MHz. At this rate, the triplet will ‘flip’ into a singlet after around 200 ns, so this combination of parameters (coupling strength = 0.015 meV) is not suitable for error correction. The inhomogeneity between the sites needs to be reduced.

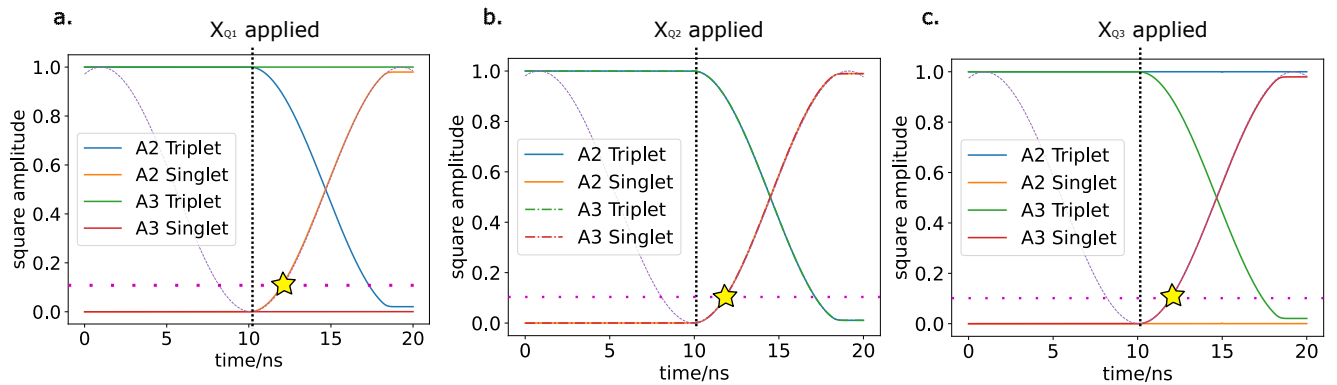


Figure 3.5: Demonstrating the expected behaviour of the ancillas when a bit flip error is artificially introduced on one of the code qubits: Q1 is flipped in **a.**, resulting in a rotation of the A2 ancilla; Q2 is flipped in **b.**, resulting in *both* A2 and A3 rotating, and Q3 is flipped in **c.**, so that A3 rotates only. In each case, the purple dashed line shows the expected frequency of rotation derived in section 2.2, Λ_t . The dotted purple line shows the arbitrarily set threshold for detection. Note that the period of rotation is much faster than the other typical timescales of quantum dot qubit systems.

Chapter 4

Feasibility of the scheme

There are features of this scheme which are potentially attractive, such as the fact that it does not require error correction steps to be programmed into a circuit, and the essentially instantaneous error detection that should lead to reduced ‘dead time’. However, the above proposal is unlikely to be practical as a means of increasing electron fidelity with current experimental parameters.

The main problem with this scheme is that it only provides a way to protect against bit flip errors. The typical coherence time for such ‘relaxation’ processes, T_1 in silicon systems is typically in the hundreds of microseconds or a few milliseconds [Loss2021][Zwerver2022]; the best performance in this respect is isotopically purified ^{28}Si , in which all atomic nuclei have zero spin. The more pressing limitation for such systems are phase-flip errors, whose coherence times T_2^{*1} are typically 1-3 orders of magnitude shorter [Loss2021]. I can see no obvious way to extend the scheme as described here to phase-flip errors. The issue is that most discrete error corrections can make use of the fact that phase- and bit- flip errors are equivalent up to a change of basis $|0\rangle \leftrightarrow |+\rangle$, $|1\rangle \leftrightarrow |-\rangle$, and just transform into the appropriate basis before the error correcting step using Hadamard gates. This is not possible in a continuous scheme like the one described here.

Even if we restrict ourselves to trying to reduce errors due to relaxation and leave phase-flip correction to other techniques, there are still limitations due to the inhomogeneities of silicon systems. In order for the correction to provide an improvement to the current T_1 times of order milliseconds, the time it takes the ancillas rotate into singlets whilst the system is in the codespace longer than this. This requires the overall spin difference $\delta_{A1,A3} = E_{A1}^z - E_{A3}^z$ and $\delta_{A2,A4}$ to be significantly less than $1/T_1 \approx 1\text{kHz}$, which is smaller than the expected range [2].

During the course of the project I also developed a highly theoretical error-correction scheme that in principle could correct against both phase flip and bit flip errors. This I describe in appendix ??.

4.1 Conclusions

¹ T_2^* is the ‘effective’ decoherence time taking into account unknown but constant inhomogeneities in the system and applied field, whereas $T_2 > T_2^*$ is for decoherence cause by purely random environmental noise. T_2^* is normally the more relevant quantity.

Bibliography

- [1] J.C. Butcher. “A history of Runge-Kutta methods”. In: *Applied Numerical Mathematics* 20.3 (1996), pp. 247–260. ISSN: 0168-9274. DOI: [https://doi.org/10.1016/0168-9274\(95\)00108-5](https://doi.org/10.1016/0168-9274(95)00108-5). URL: <https://www.sciencedirect.com/science/article/pii/0168927495001085>.
- [2] J. C. C. Hwang et al. “Impact of g -factors and valleys on spin qubits in a silicon double quantum dot”. In: *Phys. Rev. B* 96 (4 July 2017), p. 045302. DOI: 10.1103/PhysRevB.96.045302. URL: <https://link.aps.org/doi/10.1103/PhysRevB.96.045302>.
- [3] Takashi Nakajima et al. “Quantum non-demolition measurement of an electron spin qubit”. In: *Nature Nanotechnology* 14 (2019). DOI: 10.1038/s41565-019-0426-x. URL: <https://doi.org/10.1038/s41565-019-0426-x>.
- [4] NVIDIA, Péter Vingelmann, and Frank H.P. Fitzek. *CUDA, release: 10.2.89*. 2020. URL: <https://developer.nvidia.com/cuda-toolkit>.
- [5] G A Oakes et al. “Fast High-Fidelity Single-Shot Readout of Spins in Silicon Using a Single-Electron Box”. In: (2023). DOI: 10.1103/PhysRevX.13.011023.
- [6] Florian Vigneau et al. “Probing quantum devices with radio-frequency reflectometry”. In: (2023).
- [7] Howard M Wiseman. “Quantum trajectories and quantum measurement theory”. In: *Quantum and Semiclassical Optics: Journal of the European Optical Society Part B* 8.1 (1996), p. 205.

Appendices

Appendix A

Decoherence theory

Decoherence in qubits is commonly decomposed into two main channels: phase damping and amplitude damping. Amplitude damping is concerned with processes of relaxation that cause a qubit in the higher energy $|1\rangle$ state (in my project this is represented with a spin-up electron on a qubit) to the $|0\rangle$ state. Since these states have different energies, this process is asymmetrical: relaxation requires emission of a photon and occurs at a higher rate than the reverse process, electron promotion via absorption.

The rate of amplitude damping, T_1 in silicon ranges from the order of milliseconds (e.g. 5ms in Giovanni's 2022 Paper) to seconds (20s in D. Keith et al 2019), which is between 4 and 7 orders of magnitude slower than typical gate operation times of hundreds of nanoseconds and 3-6 orders of magnitude than the best (RF reflectometry) readout times, which are 1.5 μ s in D Keith and 6 μ s in Giovanni's paper - note these give different fidelities and are just to give an idea of order of magnitude. These techniques, like the original Elzerman readout readout, rely on a spin-selective tunneling process with some known rate. Dephasing rates, T_2 and T_2^* , are much faster in silicon than T_1 : the best recorded T_2^* is (I think) 270 μ s in isotopically purified silicon (Tyryshkin et al. 2012). Since this is more comparable to the gate operation and crucially readout times, and it is a much greater problem, I will focus on trying to mitigate dephasing through a $|++\rangle, |--\rangle$ three-bit code.

Amplitude damping has krauss operators (see, e.g. Nielsen and Chuang):

$$E_0 = \begin{bmatrix} 1 & 0 \\ 0 & \sqrt{1-\gamma} \end{bmatrix} \quad E_1 = \begin{bmatrix} 0 & \sqrt{\gamma} \\ 0 & 0 \end{bmatrix}$$

where gamma can be thought of as the probability that a photon is lost in the time interval the krauss operators describe. These have the following effect on a density matrix:

$$\begin{bmatrix} \rho_{00} & \rho_{01} \\ \rho_{10} & \rho_{11} \end{bmatrix} \longrightarrow \begin{bmatrix} \rho_{00} + \gamma\rho_{11} & \sqrt{1-\gamma}\rho_{01} \\ \sqrt{1-\gamma}\rho_{10} & (1-\gamma)\rho_{11} \end{bmatrix}$$

The other decoherence channel of interest is phase damping. This is the channel we must worry about if we wish to preserve the fidelity of electrons in a $|+\rangle = (|0\rangle + |1\rangle)/\sqrt{2}$ state, as its rate is generally much greater than that of amplitude damping in silicon. Intuitively, one imagines where stray and unknown EM fields shift the relative energies of the $|1\rangle$ and $|0\rangle$ states, resulting in a phase drift of the qubits over time, which, for our ignorance of the specifics, we might model as Brownian motion:

$$\alpha|0\rangle + \beta|1\rangle \longrightarrow \alpha|0\rangle + \exp(-iW(\Delta t)\sqrt{2\lambda})\beta|1\rangle$$

where $W(\Delta t)$ is a Wiener process, that is, a Gaussian random variable with units of $t^{1/2}$ with $\mathbf{E}[W(\Delta t)] = 0$ and $\mathbf{Var}[W(\Delta t)] = \Delta t$. λ is the rate of wander with units of inverse time. This process results in a transformation on a general density matrix:

$$\begin{bmatrix} \rho_{00} & \rho_{01} \\ \rho_{10} & \rho_{11} \end{bmatrix} \longrightarrow \begin{bmatrix} \rho_{00} & e^{-\Delta t\lambda}\rho_{01} \\ e^{-\Delta t\lambda}\rho_{10} & \rho_{11} \end{bmatrix} \quad (\text{A.1})$$

or using Krauss operators:

$$E_0 = \begin{bmatrix} 1 & 0 \\ 0 & e^{-\Delta t\lambda} \end{bmatrix} \quad E_1 = \begin{bmatrix} 0 & 0 \\ 0 & \sqrt{1-e^{-2\Delta t\lambda}} \end{bmatrix}$$

This process, although apparently qualitatively different, is mathematically equivalent to a Poisson ‘phase-flip’ process with rate λ , in which case we have

$$\alpha |0\rangle + \beta |1\rangle \longrightarrow \begin{cases} \alpha |0\rangle - \beta |1\rangle & p \\ \alpha |0\rangle + \beta |1\rangle & 1 - p \end{cases}$$

where the probability of ending up flipped at the end of the time interval is

$$p = e^{-\Delta t \lambda} \left(\frac{\lambda}{1!} + \frac{\lambda^3}{3!} \dots \right) = \frac{1}{2}(1 - e^{-\Delta t \lambda}) \quad (\text{A.2})$$

it is then straightforward to verify this gives the same density matrix transformation as in (A.1). This equivalence gives an intuitive way of understanding the futility of attempting to preserve the state of the state undergoing Brownian phase motion through the quantum Zeno effect: measurement has no effect on states that are already an eigenstate of the measurement. Mathematically this is easiest to see by transforming to the $|+\rangle$, $|-\rangle$ basis, where a mixed state whose eigenstates are basis states transforms as follows:

$$\begin{aligned} \begin{bmatrix} \rho_{++} & 0 \\ 0 & \rho_{--} \end{bmatrix} &\longrightarrow \begin{bmatrix} \frac{1}{2}(1 + e^{-\Delta t \lambda})\rho_{++} + \frac{1}{2}(1 - e^{-\Delta t \lambda})\rho_{--} & 0 \\ 0 & \frac{1}{2}(1 - e^{-\Delta t \lambda})\rho_{++} + \frac{1}{2}(1 + e^{-\Delta t \lambda})\rho_{--} \end{bmatrix} \\ &= \begin{bmatrix} (1 - p)\rho_{++} + p\rho_{--} & 0 \\ 0 & p\rho_{++} + (1 - p)\rho_{--} \end{bmatrix} \end{aligned}$$

In this basis we may write:

$$\rho \longrightarrow (1 - p)\rho + pX\rho X$$

So after the decoherence has been applied, whatever the underlying process, we may consider the system to still be a purely probabilistic mixture of $|+\rangle$ and $|-\rangle$, decaying to the maximally mixed state as $\Delta t \rightarrow \infty$.

This result, which extends naturally to the N qubit case, opens the door to a purely Markovian analysis of complex error correction schemes.

Explanation in terms of Coherence Theory

The theory of coherence quantification, set in motion by Baumgratz et al. in 2014 with their paper ‘Quantifying Coherence’, seeks to treat quantify quantum coherence as a resource to be used up by decohering operations. The development of this theory has led to the definitions of *incoherent* (from the original paper) and *strictly incoherent* (Yadin et al 2016) quantum operations. The theory requires the definition of a fixed basis with respect to which coherence is measured (the $|+\rangle$ is only incoherent in the $|+\rangle$, $|-\rangle$ basis, for instance).

An incoherent operation is defined as one that maps incoherent states, states which are diagonal in the chosen basis, to other diagonal states. The definition given is that for an operation defined by Krauss operators $\{E_\mu\}$, for each of the E_μ :

$$E_\mu \sum_i p_i |i\rangle\langle i| E_\mu^\dagger = \sum_i q_i |i\rangle\langle i|$$

While not mentioned by the authors, it is straightforward to show that the property of incoherence of an operation, in contrast to the property of incoherence of a *state*, is unitarily invariant, that is, an operation that is incoherent in one basis is necessarily incoherent in all others. In addition, it is obvious that an operation whose Krauss operators are chosen from the (possibly scaled) Krauss operators of other incoherent operators is itself incoherent. Within this framework it is easy to see why amplitude damping, bit flip (both manifestly incoherent in the computational basis), phase flip (manifestly incoherent in the $|+\rangle$, $|-\rangle$ basis), and phase-bit flip (manifestly incoherent in the $(|0\rangle \pm i|1\rangle)/\sqrt{2}$ basis) channels as well as combinations of the above such as the depolarising channel are all incoherent operations. Of these, the amplitude damping and bit-flip channels are further *strictly incoherent*, meaning the measurement statistics of the state after their application are independent of the off-diagonal elements of the state prior to the application. The theory of quantifying coherence then requires that any useful measure of coherence monotonically decrease under the application of these operations. In our case, we start in a incoherent state, so for all such measures \mathcal{C} , $\mathcal{C}(\rho_0) = 0$, which remains true for all subsequent time.

Appendix B

Detecting errors in ancilla qubits.

One problem the error correction scheme described in section 2.2 is that it is entirely vulnerable to errors occurring in the ancilla qubits. If one such error did occur, the system would assume the resulting tunneling event was a result of a code qubit error and apply an erroneous correction gate, which would then push the system irreversibly out of the codespace. Figure B.1 shows a layout that incorporates redundancy into the ancilla measurements. With this setup,

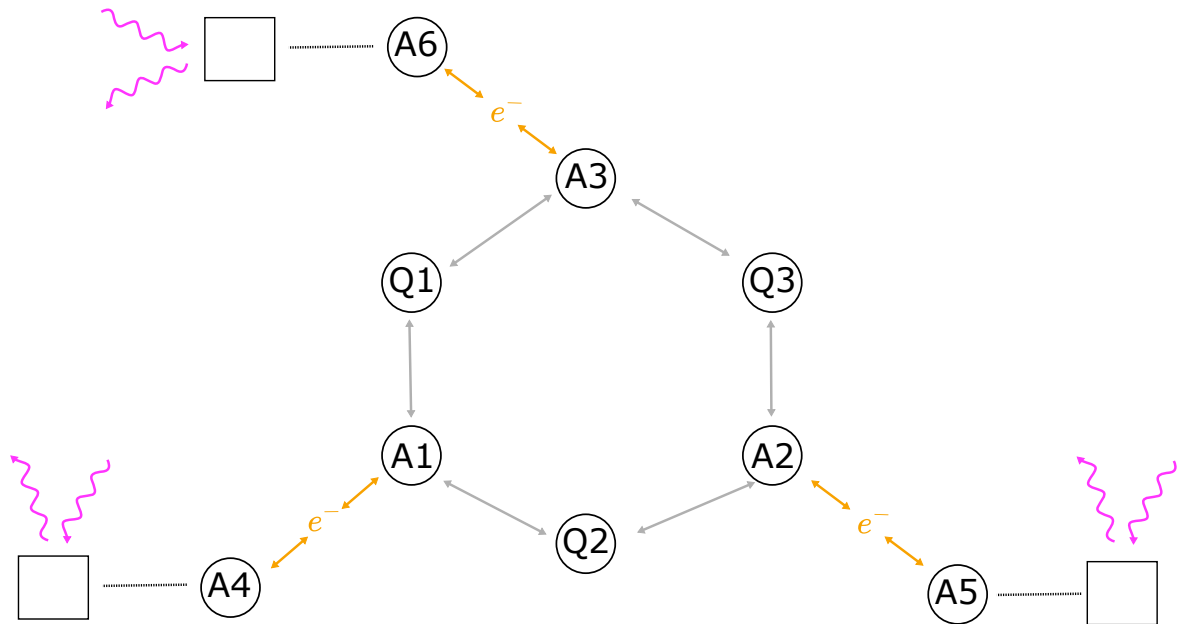


Figure B.1: The modified qubit layout that would allow for detection of ancilla errors.

the system would always expect to see two ancilla tunneling events in case of an error in the code qubits. If only one tunneling event was detected, the system could assume that the error was only in the that ancilla system, and reset it without taking the code qubits out of the codespace.

The limitation in implementing a scheme such as this one is at present topological; there has not yet been any experimental demonstration of 2D qubits arrays large enough for such a system.

Appendix C

Locked states

The purpose of this section is a system that is continuously sensitive to both phase- and bit-flip errors. I will describe a set of what I call ‘locked states’ which have the property that either a spin or phase flip results in an immediately visible charge movement, and suggest a highly theoretical outline of an error correction scheme for qubit storage. I suspect this scheme is highly impractical, but it is mathematically interesting.

Bit-flip only scheme

The idea presented in this subsection is due to Chris Long, a PhD candidate at the Cavendish and Hitachi Laboratory. Consider a line of three qubits, with charge sensors coupled to each end. We encode the qubit state $\alpha|0\rangle + \beta|1\rangle$ as $\alpha|000\rangle + \beta|111\rangle$. If we apply a strong enough permanent potential difference so that the middle quantum dot has a lower chemical potential than the outer ones, then the system would be able to lower its energy by significantly shifting the system wavefunction to have a substantial component with double occupation on the middle qubit, that is, the states $|\dots\uparrow\dots\rangle$. Colloquially, the electrons will ‘want’ to tunnel onto the middle qubit. However, the Pauli exclusion principle prevents them from doing so, as in each branch of the wavefunction all three electrons have the same spin. When one of the qubits changes state from up to down, however, one or both of the outer qubits (depending on which electron flipped) will be free to tunnel. This should be visible through the charge detector. It is not clear how possible it is to then reset the qubit to the codespace whilst preserving the encoded qubit. In any event, this scheme does not protect against phase-flip errors.

Locked states

The following are my own invention (as far as I know). Consider the n -qubit states ($n \geq 3$) defined by

$$|L0\rangle = \sqrt{\frac{2}{2^N}} \left[\sum_{n_1(s)=0 \bmod 4} |s\rangle - \sum_{n_1(s)=2 \bmod 4} |s\rangle \right]$$

$$|L1\rangle = \sqrt{\frac{2}{2^N}} \left[\sum_{n_1(s)=0 \bmod 1} |s\rangle - \sum_{n_1(s)=2 \bmod 3} |s\rangle \right]$$

where $n_1(s)$ is the number of 1s that appear in the bit-string representation of state $|s\rangle$.

For example, for $n = 3$ we have

$$|L0\rangle = (|000\rangle - |011\rangle - |101\rangle - |110\rangle)/2 \equiv (|\uparrow\uparrow\uparrow\rangle - |\uparrow\downarrow\downarrow\rangle - |\downarrow\uparrow\downarrow\rangle - |\downarrow\downarrow\uparrow\rangle)/2$$

$$|L1\rangle = (-|111\rangle + |100\rangle + |010\rangle + |001\rangle)/2 \equiv (-|\downarrow\downarrow\downarrow\rangle + |\downarrow\uparrow\uparrow\rangle + |\uparrow\downarrow\uparrow\rangle + |\uparrow\uparrow\downarrow\rangle)/2$$

. The $n = 3$ case is most relevant for physical systems, but the mathematical properties given here are general. Note

- These two states are orthogonal $\langle L0|L1\rangle = 0$, so linear combinations of them can encode a qubit.
- All electron hopping in this system is forbidden by either Pauli or triplet blockade. The state has no coupling to any doubly occupied state in the fock basis, regardless of which hopping terms in a Hubbard Hamiltonian are switched on.

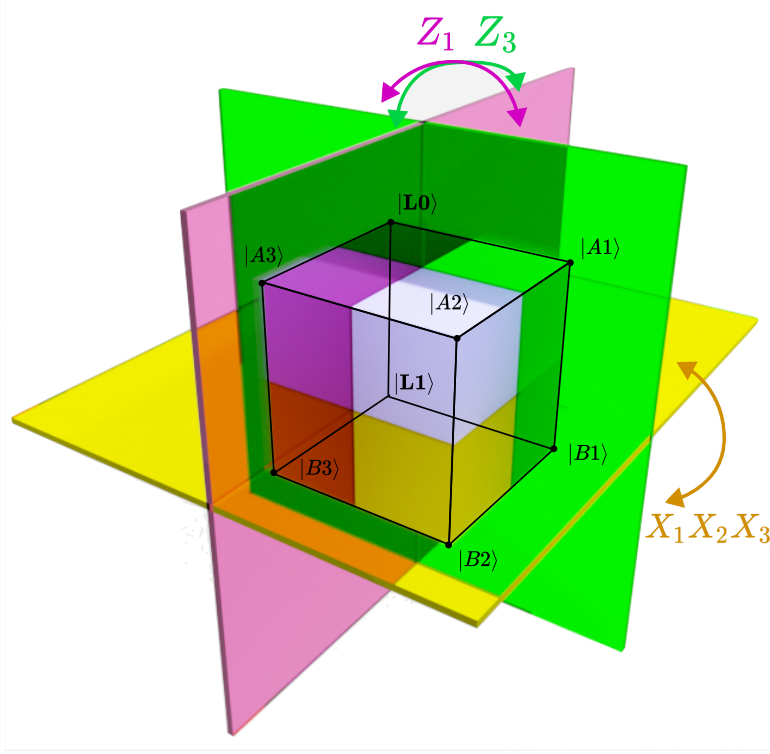


Figure C.1: Group structure of the $n = 3$ locked states and error states. Starting from the locked states, phase and bit flip errors with a combination of reflections in the planes shown.

Now suppose a bit-flip or error flip occurs on any of the qubits. Then

$$\begin{aligned} Z_1 |L0\rangle &= -X_1 |L1\rangle = (|000\rangle - |011\rangle + |101\rangle + |110\rangle)/2 = |A1\rangle \\ X_1 |L0\rangle &= Z_1 |L1\rangle = (|100\rangle - |111\rangle + |001\rangle + |010\rangle)/2 = |B1\rangle \end{aligned}$$

with $|A2\rangle, |A3\rangle, |B2\rangle, |B3\rangle$ defined similarly. The three generators of the group $\mathbb{Z}_2 \times \mathbb{Z}_2 \times \mathbb{Z}_2$ can be mapped to Z_1, Z_3 and $X_1 X_2 X_3$ which defines a representation of this group acting on the 3 qubit Hilbert space. The 8 group elements can be mapped to the identity, the three phase-flips, the three bit-flips, and a phase-and-bit flip that takes $|L0\rangle$ to $|L1\rangle$, an instantaneous process for which there would be no clear mechanism. It is clear, therefore, that any combination of these errors will keep the state within this set of states.

The states $|Ai\rangle$ and $|Bi\rangle$ have the property that turning on the hopping term in the Hubbard Hamiltonian between qubit i and any other qubit j will now result in a non-zero coupling to the the relevant doubly occupied states. If a potential difference is applied to encourage tunneling, this double occupation will be registered on a charge sensor. The nature of the wavefunction collapse in this case is somewhat unclear.

Creating the locked states

Despite their apparent complexity, these states can be created using a quantum circuit with depth that is linear in the number of qubits. I will give the n -qubit circuit, and illustrate the effects for clarity on the $n = 3$ case.

Start with a single ancilla qubit and the n qubits which we want to turn into a locked state and apply a Hadamard gate to each of this latter group:

$$|0\rangle |000\rangle \xrightarrow{H^{\otimes n}} |0\rangle (|000\rangle + |001\rangle + |010\rangle + |011\rangle + |100\rangle + |101\rangle + |110\rangle + |111\rangle)/\sqrt{8}$$

. Then apply a $Rz(\pi/2)$ rotation to every qubit:

$$\xrightarrow{Rz(\pi/2)^{\otimes n}} |0\rangle (|000\rangle + i|001\rangle + i|010\rangle - |011\rangle + i|100\rangle - |101\rangle - |110\rangle - i|111\rangle)/\sqrt{8}$$

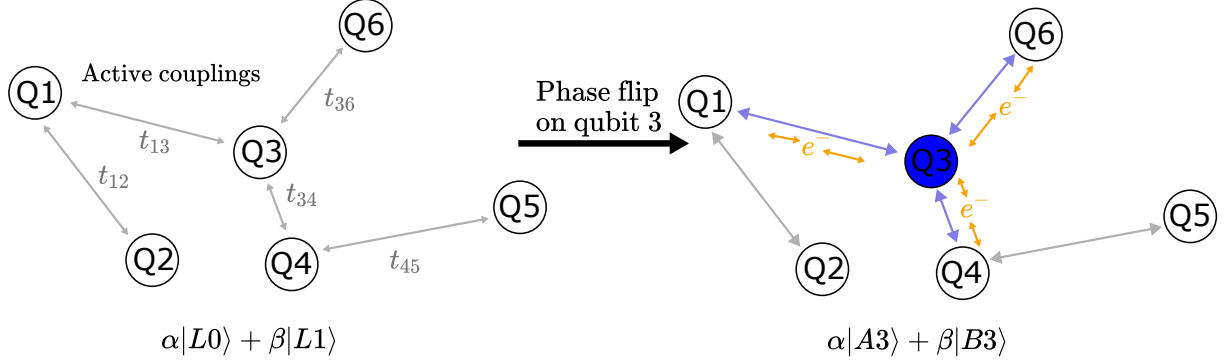


Figure C.2: The locked property of a locked state. In the codespace, spanned by $|L0\rangle$ and $|L1\rangle$, no electrons can ‘move’ (i.e. the state is uncoupled to the doubly occupied states). If, for example, a phase flip occurs on qubit 3, then the state moves into the $|A3\rangle, |B3\rangle$ error subspace. In this subspace, electrons are free to hop between qubit 3 and the neighbouring qubits where there is a coupling applied. This can perhaps be detected by a charge sensor.

. Now apply controlled X gates, conditioned on each qubit and targeting the ancilla:

$$\begin{aligned} \frac{\prod_i^{\otimes} CX(i,0)}{4} & \frac{|0\rangle}{4} (|000\rangle - |011\rangle - |101\rangle - |110\rangle) + \frac{i|1\rangle}{4} (|001\rangle + |010\rangle + |100\rangle - |111\rangle) \\ & = \frac{|0\rangle}{\sqrt{(2)}} |L0\rangle + \frac{i|1\rangle}{\sqrt{(2)}} |L1\rangle \end{aligned}$$

. We now need to configure this into a general linear combination of $|L0\rangle$ and $|L1\rangle$.

$$\begin{aligned} & \xrightarrow{Rz_0(-\pi/2)} \frac{|0\rangle}{\sqrt{(2)}} |L0\rangle + \frac{|1\rangle}{\sqrt{(2)}} |L1\rangle \\ & \xrightarrow{\prod_i^{\otimes} CX(0,i)} \frac{|0\rangle + |1\rangle}{\sqrt{2}} |L0\rangle \\ & \xrightarrow{U} (\alpha|0\rangle + \beta|1\rangle) |L0\rangle \\ & \xrightarrow{\prod_i^{\otimes} CX(i,0)} \alpha|0\rangle |L0\rangle + \beta|1\rangle |L1\rangle \end{aligned}$$

Error detection and correction

It is clear from the above how these locked states could be used for an error detection scheme. It is less clear to what extent these states could be used for error correction. An outline scheme could be: Start with two groups of qubits, each in a locked state, with no t coupling between them, in the codespace:

$$|\psi(0)\rangle = \alpha |L0\rangle |L0\rangle + \beta |L1\rangle |L1\rangle$$

. Now suppose a phase flip occurs in the second qubit of the second group of qubits:

$$\begin{aligned} & \xrightarrow{Z_{2,2}} \alpha |L0\rangle |A2\rangle + \beta |L1\rangle |B2\rangle \\ & = \alpha |L0\rangle (-|101\rangle + |111\rangle + |110\rangle + |011\rangle)/2 + \beta |L1\rangle (-|010\rangle + |000\rangle + |001\rangle + |100\rangle)/2. \end{aligned}$$

This allows electrons to hop between qubits 1 and 2, and between 2 and 3 of the second group. The component of double occupation should cause the system to collapse into a charge eigenstate with double occupation on one of the three qubits. If the system then relaxes back into the computational basis, perhaps due to the potential difference being lowered, the only trace of which type of error occurred will remain on the unaffected qubit in the second group.

In the example above, the charge hops between qubits 2 and 3 of the second group (in bold). The single occupation states must collapse away, and we are left with, after relaxation:

$$\rightarrow \alpha |L0\rangle |110\rangle + \beta |L1\rangle |010\rangle.$$

. We can use an ancilla qubit and a series of CX gates with the qubits of the first locked state to find out which error occurred:

$$\xrightarrow{\text{ancilla}} |0\rangle (\alpha |L0\rangle |110\rangle + \beta |L1\rangle |010\rangle) \xrightarrow{\prod_i^{\otimes} CX(i,0)} \alpha |0\rangle |L0\rangle |110\rangle + \beta |0\rangle |L1\rangle |010\rangle$$

. We can then apply the following projective measurement:

$$P_1 = |00\rangle\langle 00| + |11\rangle\langle 11| \qquad P_2 = |01\rangle\langle 01| + |10\rangle\langle 10|$$

where the first and second qubit represent the ancilla qubit and the first qubit of the second group respectively. This tells us the error syndrome, and it is just a matter of applying more CX gates following the construction procedure detailed above to return the system to the codespace.